

AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

Əlyazma hüququnda

Məmmədov Rasim Fəmil oğlu
Hümbətov Qara Ələkbər oğlu
Məmmədli Ülvi Aftandil oğlu
İbişov Natiq Vahid oğlu
Nəbiyev Şahin Ramin oğlu

**Ağıllı şəhər ekosistemi çərçivəsində turistlərə dəstək Web layihəsinin
işlənməsi**

mövzusunda

MAGİSTRİK DİSSERTASIYASI

İxtisas: 060632 – “İnformasiya texnologiyaları və sistemləri mühəndisliyi”

İxtisaslaşma: “Tətbiqi program təminatı”

Elmi rəhbər:

t.e.d., professor Əliyev Ələkbər Əliqə oğlu

BAKİ – 2023

MÜNDƏRİCAT

İXTİSARLARIN SİYAHISI.....	4
GİRİŞ.....	5
I Titul vərəqi (Məmmədli Ülvi Aftandil oğlu, İbişov Natiq Vahid oğlu).....	7
I FƏSİL. AĞILLI ŞƏHƏRLƏR VƏ TURİZMƏ TƏSİRİ	8
1.1. Texnologiya turizm sahəsini necə dəyişdirir	8
1.2. Ağıllı şəhər nədir və niyə önəmlidir?	10
1.3. Ağıllı şəhərlərin tarixi və inkişafı haqqında qısa məlumat	11
1.4. Ağıllı şəhərlərin komponentləri.....	11
1.5. Əlaqə və kommunikasiya infrastrukturu	12
1.6. Ağıllı şəhərlərin faydaları.....	13
1.7. Ağıllı şəhərlərin çətinlikləri.....	15
1.8. Ağıllı şəhərlərin nümunələri və ağıllı şəhərlərin gələcəyi	16
1.9. Ağıllı şəhərin inkişafı üçün trendlər və proqnozlar.....	18
1.10. Gələcəkdəki inkişaf üçün potensial maneələr və problemlər	18
II Titul vərəqi (Məmmədov Rasim Famil oğlu, Məmmədli Ülvi Aftandil oğlu, Hümbətov Qara Ələkbər oğlu, Nəbiyev Şahin Ramin oğlu).....	20
II FƏSİL. AIRBNB VƏ BƏNZƏRİ TURİSTİK XİDMƏTLƏR TƏKLİF EDƏN LAYİHƏLƏR VƏ ONLARIN İŞLƏMƏ MEXANİZMİ.....	21
2.1. Bizim layihə necə özünü maliyyələşdirəcək və pul qazanacaq?	22
2.2. Layihənin qarşısındakı əsas məqsədlər və həll edəcəyi əsas çətinliklər.....	22
2.3. Layihənin istifadəçilərinə və ağıllı şəhər ekosisteminə təmin etdiyi üstünlüklər	25
2.4. Layihənin rəqiblərindən üstün cəhətləri.....	26
2.5. Metodologiya	27
2.6. İstifadə olunan texnologiyalar.....	28
III Titul vərəqi (Məmmədov Rasim Famil oğlu, Hümbətov Qara Ələkbər oğlu, Nəbiyev Şahin Ramin oğlu).....	37
III FƏSİL. TƏTBİQİN HAZIRLANMASI	38
3.1. Backend (server tərəfi).....	38
3.2. Repository service pattern (repository və servis ilə dizayn).....	40

3.3. Swagger.....	42
3.4. Dependency Injection	42
3.5. Json veb token	43
3.6. ORM.....	44
3.7. Database (verilənlər bazası).....	46
3.8. Frontend (istifadəçi tərəfi)	49
NƏTİCƏ.....	59
İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT.....	63

İXTİSARLARIN SİYAHISI

API - Application programming interface

JSON - Javascript object notation

ORM - Object relational mapper

DI - Dependency injection

JSX - Javascript extension

XML - Extensible Markup Language

CRUD - Create, Read, Update, Delete

ACID - Atomicity, Consistency, Isolation, Durability

SQL - Structured Query Language

OOP - Object oriented programming

NPM - node package management

GİRİŞ

Mövzunun aktuallığı. Ağıllı şəhər texnologiyasının istifadəsi ilə xidmət edən, yaşayış qabiliyyətini artırmağı və vətəndaşların ehtiyaclarını ödəməyi hədəfləyən davamlı, səmərəli və yenilikçi bir şəhər modelidir. Bu şəhərlər ekoloji və sosial davamlılığı yaxşılaşdırmaq, kommunal xidmətləri yaxşılaşdırmaq və internetə qoşulmuş sensorlar, cihazlar və məlumatların təhlili vasitəsilə vətəndaşların həyatını asanlaşdırmaq üçün nəzərdə tutulub. Ağıllı şəhərlər bir çox sahələrdə innovativ həllər təklif edir. Məsələn, ağıllı binalar ekoloji dayanıqlığı artırır, eyni zamanda enerji istehlakını optimallaşdıraraq xərcləri azaldır. İntellektual nəqliyyat sistemləri vətəndaşların ictimai nəqliyyatdan daha rahat və təhlükəsiz istifadə etməsinə şərait yaradaraq, nəqliyyat axınını optimallaşdırır, trafiği azaldır və səyahət müddətlərini qısaldır. Ağıllı su idarəetmə sistemləri su ehtiyatlarını idarə etmək və suya qənaəti artırmaq üçün istifadə olunur. Vətəndaşlar ağıllı şəhər proqramları vasitəsilə şəhərdə havanın keyfiyyəti və səs-küy səviyyəsi kimi mühüm məlumatları əldə edə, ekoloji problemləri izləyə və bələdiyyələrlə qarşılıqlı əlaqə qura bilərlər. Bu şəhərlər internetə qoşulmuş sensorlar, cihazlar və məlumat analitikasından istifadə edərək şəhərləri daha ağıllı edir. Ağıllı şəhərlərin inkişafı özü ilə bir çox üstünlüklərlə yanaşı, bir çox çətinliklər də gətirir. Onlar maliyyə, texniki, hüquqi və idarəetmə baxımından bir çox problemlərlə üzləşə bilərlər. Xüsusilə maliyyə baxımından ciddi problemlər ola bilər. Ağıllı şəhərlərin yüksək qiymətli texnoloji infrastruktura və ekspert kadrlara ehtiyacı var. Bu xərclər adətən bələdiyyələr və dövlət qurumları tərəfindən ödənilir. Lakin bu qurumlar çox vaxt məhdud büdcə ilə işləməli olurlar. Texniki nöqtəyi-nəzərdən ağıllı şəhərlərin dizaynı və tətbiqi bir çox fənləri və texnologiyaları birləşdirən mürəkkəb bir prosesdir.

İşin məqsədi. Bu dissertasiya işi turistlər üçün veb-sayt layihəsinin hazırlanmasına yönəlmişdir. Bu layihənin məqsədi turistlərə təfərrüatlı təyinatlar, yerli hadisələr, yerləşdirmə variantları, nəqliyyat və fəaliyyətlər kimi əsas səyahət məlumatları ilə təmin edən istifadəçi dostu və məlumatlandırıcı platforma yaratmaqdır.

Tədqiqat metodları. Layihə turistlərin üzləşdiyi ən son tendensiyalar və problemlər də daxil olmaqla, cari turizm bazarında araşdırma aparmaqla başlayır. Tədqiqatın nəticələrinə əsasən, interaktiv xəritələr və fərdi tövsiyələr kimi yenilikçi xüsusiyyətləri özündə birləşdirən veb-sayt dizaynını təklif edir. Həmçinin veb-saytın funksiyalarını həyata keçirmək üçün tələb olunan müxtəlif inkişaf alətləri və üsullarını, o cümlədən cavab verən dizayn, axtarış sisteminin optimallaşdırılması və təhlükəsiz ödəniş inteqrasiyasını təhlil edir. Nəhayət, layihənin uğurunu istifadəçi cəlbi, sayt trafiki və gəlir əldə etmək kimi göstəriciləri ölçməklə qiymətləndirir. Ümumiyyətlə, bu layihə turistlərə hərtərəfli və fərdi səyahət təcrübəsi təqdim etməklə turizmin inkişafına töhfə vermək məqsədi daşıyır.

İşin strukturu və həcmi. Magistr dissertasiya işimizin strukturu giriş, ixtisarlara siyahısı, əsas hissə, nəticə və istifadə edilmiş ədəbiyyat siyahısından ibarətdir. Əsas hissə 3 fəsildən təşkil olunub. Birinci fəsil, ağıllı şəhərlər və onların turizmə təsirlərindən bəhs edir. İkinci fəsildə, layihəmizin bənzəri olan digər platformaların göstərdiyi xidmətlərdən və işləmə mexanizmlərindən danışılır. Sonuncu fəsildə isə bizim layihənin hazırlanma metodologiyası və tətbiq haqqında məlumatlar qeyd edilib.

AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

Əlyazma hüququnda

Məmmədli Ülvi Aftandil oğlu
İbişov Natiq Vahid oğlu

Ağıllı şəhərlər və turizmə təsiri

mövzusunda

MAGİSTRİK DİSSERTASİYASI

İxtisas: 060632 – “İnformasiya texnologiyaları və sistemləri mühəndisliyi”

İxtisaslaşma: “Tətbiqi program təminatı”

Elmi rəhbər:

t.e.d., professor Əliyev Ələkbər Əliqə oğlu

BAKİ – 2023

I FƏSİL. AĞILLI ŞƏHƏRLƏR VƏ TURİZMƏ TƏSİRİ

1.1. Texnologiya turizm sahəsini necə dəyişdirir

Son illərdə texnologiya insanların, xüsusən də şəhərlərdə səyahət tərzində inqilab edir. Ağıllı şəhərlər hazırda turistlərə nəqliyyatın idarə olunmasından tutmuş mədəni və əyləncə məkanlarına daha yaxşı çıxışa qədər təkmil təcrübə təqdim etmək üçün texnologiyadan istifadə edir. Ağıllı şəhərlər konsepsiyası yeni deyil, lakin turizm sənayesində getdikcə daha çox əhəmiyyət kəsb edir. Ağıllı şəhərlər şəhər infrastrukturunu optimallaşdırmaq və həm vətəndaşlar, həm də ziyarətçilər üçün həyat keyfiyyətini yaxşılaşdırmaq üçün texnologiyadan istifadə edir. Bura nəqliyyatın daha səmərəli idarə edilməsi, təkmil ictimai nəqliyyat, mədəni və əyləncə məkanlarına daha asan giriş daxildir. Əşyaların İnternetindən (IoT) istifadə etməklə, ağıllı şəhərlər həm də təhlükəsizliyi, təhlükəsizliyi və davamlılığını yaxşılaşdırmağa bilər. Turistlər üçün bu, daha yaxşı ümumi təcrübə deməkdir. Ağıllı şəhərlər turistlərə real vaxt rejimində şəhər və onun görməli yerləri haqqında məlumat, həmçinin ziyarət ediləcək yerlər və görülməli işlər üçün fərdi tövsiyələr verə bilər. Turistlər həmçinin ən yaxşı marşrutları tapmaq və insanların sıx olduğu yerlərdən qaçmaq üçün mobil proqramlardan istifadə edə bilərlər. Bundan əlavə, sensorlar və kameralar ictimai yerləri izləmək və turistlərə daha təhlükəsiz təcrübə təmin etmək üçün istifadə edilə bilər. Ağıllı şəhərlərin faydaları təkcə turistlərlə məhdudlaşmır. Əyləncə və mədəniyyətə təkmil çıxışı təmin etməklə, ağıllı şəhərlər yerli iqtisadiyyatları da stimullaşdırmağa bilər. Bundan əlavə, ictimai nəqliyyatı və nəqliyyat axını yaxşılaşdırmaqla, ağıllı şəhərlər çirklənməni azalda və havanın keyfiyyətini yaxşılaşdırmağa bilər. Ağıllı şəhərlər turizm sənayesində getdikcə daha çox əhəmiyyət kəsb edir və onların təsirinin yalnız gələcək illərdə artacağı gözlənilir. Düzgün texnologiya ilə ağıllı şəhərlər yerli iqtisadiyyatları stimullaşdırmaq və davamlılığını təşviq etməklə yanaşı, turistlərə təkmil təcrübə təqdim edə bilər [1]. Dünyanın şəhərləri öz infrastruktur və xidmətlərini təkmilləşdirmək üçün texnologiyaya sərmayə qoyduqca “ağıllı şəhərlər” anlayışı getdikcə populyarlaşır. Ağıllı şəhərlər şəhər ərazilərini öz vətəndaşları üçün daha səmərəli, dayanıqlı və yaşana bilən etmək üçün texnologiyadan istifadə edirlər. Səyahətçilər üçün bu texnologiya gücləndirilmiş təhlükəsizlik və rahatlıqdan tutmuş xərclərin azaldılmasına qədər bir

sıra üstünlüklər təmin edə bilər. Səyahətçilər üçün ağıllı şəhər texnologiyasının ən böyük üstünlüklərindən biri artan təhlükəsizlikdir. Ağıllı şəhərlər ictimai təhlükəsizliyi artırmaq üçün müşahidə kameraları, sifətin tanınması proqramı, süni intellektlə işləyən analitika və s. kimi müxtəlif texnologiyalardan istifadə edir. Bu texnologiyalar şübhəli fəaliyyətləri tez aşkarlaya və səyahətçilərin təhlükəsizliyini təmin etmək üçün səlahiyyətliyə xəbər verə bilər. Bundan əlavə, ağıllı şəhər texnologiyası səyahətçilərə səyahətin mümkün qədər səmərəli və təhlükəsiz olmasını təmin etmək üçün nəqliyyat, yol şəraiti və hava haqqında real vaxt məlumatı təqdim edə bilər. Ağıllı şəhər texnologiyası da səyahəti daha rahat edə bilər. Ağıllı şəhərlər havanın keyfiyyətindən tutmuş parkinqin mövcudluğuna qədər hər şey haqqında məlumat toplayan sensorlarla təchiz olunub. Bu məlumat səyahətçilər tərəfindən təyinat yerinə ən sürətli marşrutu tapmaq, mövcud parkinq yerlərini tapmaq və öz marşrutlarını planlaşdırmaq üçün istifadə edilə bilər. Bundan əlavə, bəzi şəhərlərdə səyahətçilərin tranzit, parkinq və digər xidmətlər üçün ödəniş etmələrini asanlaşdırmaq üçün təmassız ödəniş sistemləri tətbiq edilib. Nəhayət, ağıllı şəhərlər səyahətçilərə pula qənaət etməyə kömək edə bilər. Ağıllı şəhərlər enerji səmərəliliyini artırmaq, tullantıları azaltmaq və dövlət xidmətlərini sadələşdirmək üçün texnologiyadan istifadə edir, nəticədə həm vətəndaşlar, həm də səyahətçilər üçün daha az xərclər olur [2]. Ağıllı şəhərlər ictimai nəqliyyat sistemlərini optimallaşdırmaq üçün məlumatlardan da istifadə edir ki, bu da gediş haqqının aşağı salınması və daha səmərəli marşrutların olması ilə nəticələnir. Nəticə olaraq, ağıllı şəhər texnologiyası səyahətçilər üçün artan təhlükəsizlikdən tutmuş xərclərin azaldılmasına qədər bir sıra üstünlüklər təklif edir. Daha çox şəhər ağıllı şəhər texnologiyasına sərmayə qoyduqca, səyahətçilər səyahətlərini daha əyləncəli, səmərəli və təhlükəsiz edəcək bir sıra üstünlüklər görəceklerini gözləyə bilərlər.

Ağıllı şəhərlər və onlar xidmət təklif edən saytlar istifadəyə verilmədən öncə insanların qarşılaşdığı problemlər və layihənin bu problemlərə gətirəcəyi həllər Airbnb və digər onlayn veb-saytlardan əvvəl insanlar yaşayış yerlərini idarə etmək üçün başqa üsullardan istifadə edirdilər, məsələn:

1. Otel rezervasiyalari: İnsanlar otelə birbaşa zəng edərək və ya səyahət agentlərindən istifadə etməklə oteldə otaq sifariş edə bilərdilər.
2. Hostellər: Hostellər büdcə ilə səyahət edən insanlar, xüsusən də gənc bel çantaları ilə səyahət edənlər üçün sərfəli seçim olub və hələ də belədir.
3. Tətil kirayələri: Tətil kirayələri yerli əmlak idarəçiliyi şirkətləri tərəfindən təklif edilirdi və adətən ağızdan-ağıza və ya yerli turizm ofisinə müraciət etməklə tapılırdı.
4. Evin dəyişdirilməsi: İnsanlar evlərini başqa yerdə yaşayan başqaları ilə dəyişdirərdilər. Bu üsul adətən şəxsi əlaqələr vasitəsilə və ya ev dəyişdirmə təşkilatlarına qoşulmaqla təşkil edilirdi.
5. Couchsurfing: İnsanlar adətən mədəni mübadilə və qarşılıqlı mehribanlıq müqabilində yerli sakinlərlə pulsuz qala bilərdilər. Couchsurfing adətən onlayn forumlar və sosial media qrupları vasitəsilə təşkil edilirdi.

Bu üsullar indi mövcud olan sürətli və asan onlayn sifarişdən daha çox səy və planlaşdırma tələb edirdi; bununla belə, onlar Airbnb-nin təklif edə bilməyəcəyi daha çox şəxsi əlaqələr və fərdi tənzimləmələr üstünlüyünə sahib idilər [1].

1.2. Ağıllı şəhər nədir və niyə önəmlidir?

Ağıllı şəhərlər sürətlə artan şəhər əhalisi və davamlılıq problemləri kimi bir çox problemə innovativ həllər təklif edən şəhər çevrilmə layihələridir. Bu layihələrin inkişafı həm vətəndaşların həyat keyfiyyətini yaxşılaşdırmaqla, həm də ətraf mühiti mühafizə etməklə şəhərin davamlılığını təmin edir. Ağıllı şəhərlər vətəndaşların ehtiyac və istəklərini qarşılamağı hədəfləyir. Bu şəhərlərdə ağıllı nəqliyyat sistemləri sayəsində nəqliyyat axınının optimallaşdırılması ilə nəqliyyat sıxlığı azalır, yollardan daha səmərəli istifadə edilir, vaxta və yanacağa qənaət edilir. Eyni zamanda, ağıllı ictimai nəqliyyat sistemləri vətəndaşlara daha sürətli və təhlükəsiz səyahət etməyə imkan verir. Ağıllı şəhərlərin digər mühüm faydası ekoloji davamlılığı təmin etmələridir. Bu şəhərlərdə bərpa olunan enerji mənbələrindən istifadə edilir, enerji səmərəliliyi artır və karbon izi azalır. Ağıllı binalar enerji istehlakını azaltmaq və ətraf mühiti qorumaq üçün nəzərdə tutulub. Bundan əlavə, suya qənaət etmək və ətraf mühitin su ehtiyatlarını qorumaq üçün ağıllı su idarəetmə sistemləri hazırlanmışdır. Ağıllı şəhərlər həm də iqtisadi fayda verir. Bu şəhərlərdə innovativ texnologiyalardan

istifadə artır ki, bu da iş imkanlarına və iqtisadi artıma kömək edir. Həmçinin, ağıllı şəhərlər vətəndaşların həyat keyfiyyətini yaxşılaşdırdıqca daha cəlbedici olur və turizm kimi sektorlarda artımı təmin edir. Nəticə etibarlı ilə ağıllı şəhərlər innovativ texnologiyaların tətbiqi ilə bir çox problemin həllini təklif edən və vətəndaşların həyat keyfiyyətini artıran şəhər çevrilmə layihələridir [10]. Bu şəhərlərin inkişafı vətəndaşlar, ətraf mühit və iqtisadiyyat üçün əhəmiyyətli faydalar təmin edir.

1.3. Ağıllı şəhərlərin tarixi və inkişafı haqqında qısa məlumat

Ağıllı şəhər anlayışı texnoloji inkişafın və davamlılıq problemlərinin artması ilə ortaya çıxan bir şəhər çevrilmə hərəkatıdır. Bu hərəkat şəhərləri daha səmərəli, təhlükəsiz, davamlı və insan yönümlü etmək məqsədi daşıyır. Ağıllı şəhərlərin tarixi 1980-ci illərin sonlarında Yaponiyada ortaya çıxan "Tsukuba Elm Şəhəri" layihəsinə gedib çıxır. Bu layihə texnologiyanın istifadəsi ilə şəhər həyatını daha səmərəli və davamlı etmək məqsədi daşıyırdı. Daha sonra, 1990-cı illərin əvvəllərində İngiltərənin paytaxtı London "Broadgate Estate" layihəsi ilə ağıllı binalar və təhlükəsizlik sistemlərindən istifadə edən ağıllı şəhər nümunəsini təqdim etdi. Ağıllı şəhər anlayışı 21-ci əsrin əvvəllərində ABŞ və Avropada sürətlə yayılmağa başladı. Bu dövrdə internetin yayılması, simsiz texnologiyaların inkişafı, sensorların istifadəsi, ağıllı cihazların artması və böyük verilənlərin təhlilinin yayılması kimi texnoloji inkişaflarla ağıllı şəhərlərin inkişafı sürət qazandı. Bu gün bütün dünyada ağıllı şəhər anlayışından geniş istifadə olunur.

1.4. Ağıllı şəhərlərin komponentləri

IoT sensorları və cihazları. IoT (Əşyaların İnterneti), yəni Əşyaların İnterneti ağıllı şəhərlərin əsas texnoloji komponentlərindən biridir. IoT cihazları və sensorlar ağıllı şəhərlərdə müxtəlif vəzifələri yerinə yetirir və şəhərlərin daha səmərəli, təhlükəsiz və davamlı olmasına kömək edir. Ağıllı şəhərlərdə istifadə olunan bəzi IoT cihazları və sensorlar bunlardır [3]:

Ağıllı Nəqliyyat Sistemləri: Ağıllı şəhərlərdə istifadə edilən avtobuslar, metrolar və qatarlar kimi ictimai nəqliyyat vasitələri GPS texnologiyası ilə izlənilir. Beləliklə, şəhər idarəçiləri nəqliyyat axınını təhlil edə və tıxaclarla bağlı tədbirlər görə bilirlər.

Ağıllı binalar: Ağıllı binalar ətraf mühitin dəyişənlərinə və istifadəçi hərəkətlərinə həssas olan sensorlarla təchiz edilmişdir. Ağıllı binalar yanğın, sel və digər fəlakətlərə qarşı qabaqlayıcı tədbirlər görmək üçün bu sensorlardan istifadə edir.

Məlumatların toplanması və təhlili sistemləri. Ağıllı şəhərlər müxtəlif mənbələrdən böyük həcmdə məlumat yaradır. Bu məlumatlar şəhərlərin daha yaxşı idarə olunması üçün vacibdir. Məlumatların toplanması və təhlili sistemləri ağıllı şəhərlərdə məlumatların toplanması, saxlanması, təhlili və idarə edilməsi üçün istifadə olunur. Ağıllı şəhərlərdə istifadə olunan bəzi məlumat toplama və təhlil sistemləri bunlardır:

Xəbərdarlıq Sistemləri: Ağıllı şəhərlərdə xəbərdarlıq sistemləri, havanın çirklənməsi, sel, yanğın, tıxac və s. kimi fəvqəladə vəziyyətlərdə xalqı xəbərdar edir.

Monitorinq Sistemləri: Ağıllı şəhərlərdə monitorinq sistemləri su ehtiyatlarını, enerji sərfiyyatını, zibil toplama sürətini və s. kimi şəhər xidmətlərini izləmək üçün istifadə olunur.

1.5. Əlaqə və kommunikasiya infrastrukturu

Ağıllı şəhərlər çoxlu cihazlar və sensorlarla təchiz olunub. Bu qurğular və sensorlar bir-biri ilə və şəhər idarəetmə mərkəzləri ilə bağlıdır. Bu əlaqə və kommunikasiya infrastrukturu məlumatların düzgün toplanması, saxlanması və təhlili üçün vacibdir. Ağıllı şəhərlərdə istifadə olunan bəzi əlaqə və kommunikasiya infrastrukturları bunlardır [9]:

Simsiz Şəbəkələr: Simsiz şəbəkələr ağıllı şəhərlərdə ən çox istifadə edilən əlaqə infrastrukturudur. Bu şəbəkələr Wi-Fi, Bluetooth, Zigbee, LoRa və s. Texnologiyalardan istifadə edərək cihazlar və sensorların bir-biri ilə əlaqə saxlamasına imkan verir. Bu yolla şəhər idarəçiləri, vətəndaşlar və şirkətlər cihazlar vasitəsilə məlumatları toplaya, paylaşa və təhlil edə bilirlər.

Fiber Optik Şəbəkələr: Fiber optik şəbəkələr ağıllı şəhərlərdə yüksək sürətli internet bağlantısını təmin etmək üçün istifadə olunur. Bu şəbəkələr şəhər administratorlarına, vətəndaşlara və şirkətlərə böyük həcmdə məlumat toplamaq və təhlil etmək imkanı verir.

5G şəbəkələri: 5G şəbəkələri ağıllı şəhərlərdə istifadə olunan ən son nəsil simsiz rabitə texnologiyalarıdır. Bu şəbəkələr aşağı gecikmə və yüksək sürətli internet bağlantısı təmin edir. Bu yolla, ağıllı şəhərlərdəki cihazlar və sensorlar məlumatları daha tez və səmərəli şəkildə toplaya bilir.

Mobil rabitə: Ağıllı şəhərlərdə mobil rabitə vətəndaşlarla şəhər idarəçiləri arasında ünsiyyəti asanlaşdırmaq üçün istifadə olunur.

Yerli Şəbəkələr: Ağıllı şəhərlərdə yerli şəbəkələr, şəhər idarəetmə mərkəzləri, məktəblər, xəstəxanalar və s. kimi yerli təşkilatlar arasında ünsiyyəti asanlaşdırmaq üçün istifadə olunur.

1.6. Ağıllı şəhərlərin faydaları

Artan səmərəlilik və xərclərə qənaət. Ağıllı şəhərlərdə artan səmərəlilik və xərclərə qənaət, eləcə də davamlılıq və insan mərkəzli yanaşmalar təmin etmək məqsədi daşıyan bir yanaşma var. Bu yanaşma şəhər hökumətlərinə resurslardan daha səmərəli istifadə etməyə və xidmətlərin keyfiyyətini artırmağa imkan verir. Ağıllı şəhərlərdə məlumatların toplanması və təhlili sistemləri sayəsində şəhər hökumətləri xidmətlərini daha yaxşı planlaşdırır və daha yaxşı qərarlar qəbul edə bilirlər. Bu sistemlər şəhərdə nəqliyyat axınına nəzarət edə, enerji sərfiyyatını izləyə və zibil toplama xidmətlərini optimallaşdırır. Bu yolla, şəhər hökumətləri xidmətlərini daha səmərəli etməklə öz resurslarından daha səmərəli istifadə edə bilirlər. Bundan əlavə, ağıllı şəhərlərdə istifadə olunan texnologiyalar xərclərə qənaət edə bilər. Məsələn, ağıllı işıqlandırma sistemləri daha az enerji sərf edən LED işıqlardan istifadə etməklə enerjiyə qənaət edir və elektrik xərclərini azaldır. Bundan əlavə, ağıllı binaların enerji idarəetmə sistemləri enerji sərfiyyatına nəzarət edir və lazımsız xərclərin qarşısını alaraq pula qənaət edir. Ağıllı nəqliyyat sistemləri həm də xərclərə qənaət edir. Bu sistemlər tıxacların qarşısını almaq üçün nəqliyyat axınına nəzarət edə və işıqforları tənzimləyə bilər. Bundan əlavə, ağıllı ictimai nəqliyyat sistemləri səyahət vaxtlarını qısalda və yanacağa qənaət edə bilər. Nəticədə, ağıllı şəhərlər şəhər hökumətlərinə öz resurslarından daha səmərəli istifadə etməyə imkan verir, nəticədə səmərəliliyin artması və xərclərə qənaət edilir. Bu yanaşma şəhərlərin davamlılığını və xidmət keyfiyyətini, eləcə də davamlılığını və insan mərkəzli yanaşmalarını artırır.

Təkmilləşdirilmiş ictimai təhlükəsizlik - ağıllı şəhərlər ictimai təhlükəsizliyi və təhlükəsizliyi artırmaq üçün nəzərdə tutulub. Bu yanaşma şəhərlərdə cinayət nisbətlərinin azaldılması, fəvqəladə hallara reaksiyanın sürətləndirilməsi və fəlakətlərə hazırlıq kimi məqsədlər üçün istifadə olunur. Ağıllı şəhərlərdə təhlükəsizlik kameraları, ağıllı sensorlar və digər cihazlar vasitəsilə məlumatların toplanması və təhlili sistemindən istifadə edilir. Bu sistemlər oğurluq, vandalizm və digər cinayətlərə nəzarət etmək və qarşısını almaq üçün istifadə olunur. Bundan əlavə, ağıllı şəhərlərdə istifadə edilən bu sistemlər fəvqəladə hallara tez reaksiya verir. Məsələn, yanğın aşkarlayıcı sensorlar yanğın baş verdikdə tez xəbərdarlıq edir və yanğın söndürənlərin yanğına reaksiya verməsini təmin edir. Ağıllı şəhərlərdə fəlakətlərə hazır olmaq üçün də tədbirlər görülür. Bu tədbirlərə hava proqnozu sistemləri, daşqın və zəlzələ monitorinq sistemləri kimi bir sıra texnoloji həllər daxildir. Bu sistemlər fəlakətdən əvvəl xəbərdarlıq etməyə, insanların təhlükəsizliyini təmin etməyə və fəlakətdən sonra bərpa işlərinin tez başlamasına imkan verir.

Daha yaxşı ekoloji davamlılıq - ağıllı şəhərlər ekoloji dayanıqlığı artırmaq üçün bir sıra tədbirlər görür. Bu yanaşma ətraf mühitə ziyan vuran fəaliyyətləri azaltmaq, enerji səmərəliliyini artırmaq, tullantıları azaltmaq və təbii resurslara qənaət etmək üçün nəzərdə tutulub. Ağıllı şəhərlərdə enerji səmərəliliyini artırmaq üçün ağıllı enerji idarəetmə sistemlərindən istifadə olunur. Enerji sərfiyyatına nəzarət etməklə bu sistemlər enerjiyə qənaət edir və təbii resursların istehlakını azaldır. Bundan əlavə, bərpa olunan enerji mənbələrinə keçid üçün bir sıra tədbirlər həyata keçirilir. Məsələn, günəş panelləri, külək turbinləri və su elektrik stansiyaları kimi bərpa olunan enerji mənbələri şəhərlərin enerji ehtiyaclarını ödəmək üçün istifadə olunur. Ağıllı şəhərlər də tullantıların azaldılması və təkrar emalı üçün tədbirlər görür. Bu tədbirlərə təkrar emal proqramları, tullantıların ayrılması sistemləri və tullantıların idarə olunması həlləri daxildir. Tullantıların təkrar emalı və təkrar istifadəsini artıraraq, bu sistemlər təbii ehtiyatların istehlakını azaldır və ətraf mühitin qorunmasına kömək edir [11].

Vətəndaşların həyat keyfiyyətinin yaxşılaşdırılması - ağıllı şəhərlərin digər mühüm faydası vətəndaşların həyat keyfiyyətinin yaxşılaşdırılmasıdır. Bu sistemlər toplamaq, təhlil etmək və tez qərar qəbul etmək imkanı verməklə vətəndaşların həyatını

asan və təhlükəsiz edir. Məsələn, ağıllı şəhərlərdə sürətli və təhlükəsiz internetə çıxış vətəndaşlara rəqəmsal əlaqə saxlamağa və işlərini daha səmərəli yerinə yetirməyə imkan verir. Bundan əlavə, ağıllı nəqliyyat sistemləri vətəndaşlara daha sürətli və təhlükəsiz səyahət etmək imkanı verir. İntellektual bina sistemləri vətəndaşların öz evlərində daha rahat və təhlükəsiz həyat sürməsinə kömək edir. Ağıllı şəhərlərdə səhiyyə də daha sürətli və təsirli olur. Ağıllı sağlamlıq sistemləri vətəndaşlara öz sağlamlıq məlumatlarını təhlükəsiz saxlamağa, tibbi cihazlar və sensorlar vasitəsilə sağlamlıq vəziyyətlərinə nəzarət etməyə və teletibb vasitəsilə həkimlərlə əlaqə saxlamağa imkan verir. Nəticədə ağıllı şəhərlər vətəndaşların həyat keyfiyyətini yaxşılaşdıran bir çox üstünlüklər təklif edir. Sürətli internet, ağıllı nəqliyyat sistemləri, ağıllı bina sistemləri və ağıllı sağlamlıq sistemləri kimi bir sıra xidmətlər vətəndaşların həyatını asan və təhlükəsiz edir [9].

1.7. Ağıllı şəhərlərin çətinlikləri

Təhlükəsizlik və məxfilik problemləri. Ağıllı şəhərlərin inkişafı fərdi məxfilik və təhlükəsizlik problemləri ilə gəlir. Bu sistemlər bir çox müxtəlif mənbələrdən məlumat topladığı və bu məlumatları müxtəlif cihazlar və təşkilatlarla paylaşdığı üçün məxfiliklə bağlı narahatlıqlar yarana bilər. Məsələn, ağıllı nəqliyyat sistemləri vasitəsilə toplanan məlumatlar insanların səyahət vərzişlərini, yeri və səyahət vaxtını qeyd edə bilər. Bu məlumat müəyyən bir insanın harada yaşadığını və ya evdə olmamasını müəyyən etmək üçün istifadə edilə bilər. Buna görə də ağıllı şəhərlərin inkişafı zamanı məlumatların təhlükəsizliyini və məxfiliyini təmin etmək vacibdir. Bu o deməkdir ki, məlumat düzgün şəkildə qorunur və icazəsiz girişdən qorunur. Bundan əlavə, ağıllı şəhərlərin inkişafında istifadə edilən texnologiyaların əksəriyyəti internetə qoşulduğundan kibercümləmə riski artır. Ona görə də ağıllı şəhərlərin inkişafı zamanı kibertəhlükəsizliyin təmin edilməsi böyük əhəmiyyət kəsb edir [1].

Məlumatların idarə edilməsi və idarəetmə məsələləri - ağıllı şəhərlər bir çox müxtəlif məlumat mənbələrindən böyük həcmdə məlumat toplayan və bu məlumatlardan istifadə edərək müxtəlif qərarlar qəbul etməyə kömək edən mürəkkəb sistemdir. Buna görə də, məlumatların idarə edilməsi və idarəetmə ağıllı şəhərlərin uğurlu idarə edilməsi üçün vacibdir. Əsas məsələ çoxlu məlumat mənbələrindən alınan

məlumatları birləşdirmək və təhlil etməkdir. Bu məlumatların müxtəlif formatlarda, müxtəlif təsnifatlarda və müxtəlif vaxt intervallarında gəlməsi məlumatların bütövlüyünün qorunmasında və dəqiq təhlillərin aparılmasında problemlər yarada bilər. Digər bir məsələ isə məlumatların identifikasiyası və şəxsi məlumatların qorunmasıdır. Məlumatların düzgün idarə edilməsi və qorunması məlumatların idarə edilməsində və idarə olunmasında böyük əhəmiyyət kəsb edir.

Maliyyə və investisiya problemləri - ağıllı şəhərlər ənənəvi şəhərlərdən daha çox maliyyə və investisiya tələb edir. Bura ağıllı nəqliyyat sistemləri, ağıllı enerji idarəetmə sistemləri, IoT sensorları və cihazları, məlumatların toplanması və təhlili sistemləri, kommunikasiya infrastrukturunu, ağıllı bina və infrastruktur və s. daxildir. kimi müxtəlif texnologiyaların xərcləri hesabına baş verir. Bu texnologiyaların maliyyələşdirilməsi hökumətlər, özəl sektor şirkətləri və digər maraqlı tərəflər arasında bölüşdürülə bilər. Hökumətlər büdcə ayırmaqla və təşviqlər təklif etməklə ağıllı şəhər layihələrini dəstəkləyə bilərlər. Özəl sektor şirkətləri, xüsusilə enerji, tikinti və texnologiya sektorlarında fəaliyyət göstərən şirkətlər ağıllı şəhər layihələrinə sərmayə qoya bilərlər. Bununla belə, ağıllı şəhər layihələrinin maliyyələşdirilməsi və sərmayə qoyuluşu investisiyanın qaytarılması və geri qaytarılma müddəti baxımından da əhəmiyyətli problem yarada bilər. Ona görə də maliyyə modelləri və biznes planlarının hazırlanması və layihələrin iqtisadi dayanıqlığının təmin edilməsi üçün müxtəlif maliyyələşdirmə vasitələrindən istifadə etmək lazımdır. Nəticə etibarilə, ağıllı şəhərlərə maliyyələşdirmə və sərmayə qoyuluşu ilə bağlı çətinlikləri aradan qaldırmaq üçün hökumətlər, özəl sektor şirkətləri və digər maraqlı tərəflər arasında əməkdaşlıq etmək və müvafiq maliyyə alətlərindən istifadə etmək lazımdır. Uzunmüddətli davamlılığı və sosial faydaları nəzərə alan layihələrin hazırlanması da vacibdir.

1.8. Ağıllı şəhərlərin nümunələri və ağıllı şəhərlərin gələcəyi

Ağıllı şəhərlərin nümunələri. Dünyada bir çox ağıllı şəhər layihələri həyata keçirilib. Bəzi nümunələr bunlardır:

1. Sonqdo, Cənubi Koreya: Sonqdo Cənubi Koreyanın paytaxtı Seuldən 40 kilometr qərbdə yerləşən şəhərdir. Sonqdo təxminən 10 milyard dollar investisiya ilə tikilib.

Ağıllı şəhər texnologiyaları enerji səmərəliliyini, davamlılığı və iştirakçılığı təşviq edən bir çox xüsusiyyətlərlə təchiz edilmişdir.

2. Barselona, İspaniya: Barselona bir çox ağıllı şəhər texnologiyaları ilə təchiz olunmuş Aralıq dənizi şəhəridir. Şəhər ekoloji davamlılıq, ağıllı nəqliyyat, ağıllı binalar, ağıllı enerji idarəetməsi və ağıllı ictimai xidmətlər kimi bir çox sahədə innovativ layihələrə ev sahibliyi edir.

3. Amsterdam, Hollandiya: Amsterdam dünyanın ən inkişaf etmiş ağıllı şəhərlərindən biridir. Şəhər ağıllı nəqliyyat sistemləri, enerji idarəçiliyi, hava keyfiyyətinin monitorinqi və iştirakçı demokratiya kimi bir çox sahədə texnoloji yeniliklərə liderlik edir.

4. Sinqapur: Sinqapur ağıllı şəhər texnologiyalarından istifadə edərək şəhərin davamlılığını artırmaq üçün bir çox layihələrə ev sahibliyi edir. Şəhərdə hava keyfiyyətinin monitorinqi, ağıllı bina idarəetməsi, ağıllı trafik idarə edilməsi və tullantıların idarə edilməsi kimi bir çox innovativ texnologiyalardan istifadə olunur.

Ağıllı şəhərlər üçün potensial texnologiyalar və təsirləri - Ağıllı şəhərləri inkişaf etdirmək üçün istifadə edilə bilən bir çox texnologiya var və bu texnologiyalar şəhərlərin daha səmərəli, davamlı və insan yönümlü olmasına kömək edə bilər. Ağıllı şəhərlər üçün potensial texnologiyalar və təsirlər bunlardır:

1. Əşyaların İnterneti (IoT): IoT cihazlarından ağıllı şəhərlərdə bir çox sahədə istifadə oluna bilər. Məsələn, IoT cihazları ağıllı küçə işıqlarında, nəqliyyatın idarə edilməsi sistemlərində, ağıllı binalarda, tullantıların idarə edilməsi sistemlərində, su idarəetmə sistemlərində və bir çox başqa sahələrdə istifadə edilə bilər.

2. Süni intellekt (AI): Süni intellekt ağıllı şəhərlərdə daha yaxşı qərarlar vermək üçün istifadə edilə bilər. Məsələn, trafik axınını idarə etmək üçün istifadə edilən sistem süni intellektdən istifadə edərək, trafik vəziyyətini proqnozlaşdırma və daha tez cavab verə bilər.

3. Blockchain: Blockchain texnologiyası ağıllı şəhərlərdə təhlükəsiz və şəffaf məlumat mübadiləsi üçün istifadə edilə bilər.

4. Virtual reallıq (VR) və artırılmış reallıq (AR): Virtual reallıq texnologiyası insanlara şəhərlərin tarixi, mədəniyyəti və görməli yerləri haqqında daha çox məlumat əldə etməyə kömək edə bilər.

Bu texnologiyalar ağıllı şəhərlərin inkişafında mühüm rol oynayır və şəhərlərin daha səmərəli, dayanıqlı və insan yönümlü olmasına kömək edir.

1.9. Ağıllı şəhərin inkişafı üçün trendlər və proqnozlar

Ağıllı şəhərin inkişafı sürətlə irəliləyir və gələcəkdə daha da əhəmiyyət kəsb edəcəyi gözlənilir. Bu sahədə texnoloji inkişafın sürətli inkişafı və artan əhəlinin gətirdiyi urbanizasiya meyilləri kimi amillər təsirli olur. Ağıllı şəhərin inkişafı üçün bəzi proqnozlaşdırılan tendensiyalar və proqnozlar bunlardır [3]:

1. Əşyaların İnterneti (IoT) texnologiyalarından daha çox istifadə olunacaq: IoT texnologiyaları ağıllı şəhərlərin inkişafında əsas rol oynayır. Bu texnologiyalar sayəsində sensorlar vasitəsilə toplanan məlumatlardan istifadə edərək şəhərlərə daha yaxşı nəzarət mümkündür.
2. Süni intellektin (AI) istifadəsi artacaq: Ağıllı şəhərlərdə süni intellekt alqoritmlərindən istifadə etməklə sensorlardan toplanan məlumatları daha effektiv emal etmək və təhlil etmək mümkün olacaq.
3. Yaşıl texnologiyalardan istifadə artacaq: Ağıllı şəhərlərin inkişafı zamanı ətraf mühitin davamlılığını artırmaq üçün yaşıl texnologiyalardan istifadə artacaq. Bu çərçivədə, bərpa olunan enerji mənbələri, enerji səmərəliliyi həlləri və tullantıların idarə edilməsi sistemləri kimi bir çox yaşıl texnologiyalardan istifadə ediləcək.

1.10. Gələcəkdəki inkişaf üçün potensial maneələr və problemlər

Ağıllı şəhərlərin inkişafı texnoloji irəliləyişlər və artan əhali və şəhər konsentrasiyası kimi amillər sayəsində sürətlə irəliləyir. Bununla belə, gələcək artım üçün potensial maneələr və problemlər də mövcuddur. Onlardan bəzilərini təqdim edirik [1]:

1. İnvestisiya problemləri: Ağıllı şəhərlərin inkişafı böyük investisiyalar tələb edir. Bu investisiyaları təmin etmək, dövlət və özəl sektor arasında əməkdaşlıq və maliyyə mexanizmlərini inkişaf etdirmək lazımdır.

2. Hüquq və idarəetmə problemləri: Ağıllı şəhərlərin inkişafı məlumatların toplanması, istifadəsi və idarə edilməsi kimi bir çox hüquqi və idarəetmə məsələlərini gündəmə gətirir. Bu problemlərin həlli üçün yeni qaydalar və idarəetmə mexanizmləri hazırlanmalıdır.

3. İnfrastruktur problemləri: Ağıllı şəhərlər lazımi infraquruktura malik olmadıqda uğursuz ola bilər. Xüsusilə, simsiz şəbəkələrin genişləndirilməsi və enerji mənbələrinə çıxış ağıllı şəhərlərin inkişafı üçün çox vacibdir.

4. Məlumat təhlükəsizliyi və məxfilik: Ağıllı şəhərlər böyük həcmdə məlumat toplayır və istifadə edir. Bu məlumatların təhlükəsizliyi və məxfiliyi ilə bağlı narahatlıqlar var. Bu problemləri həll etmək üçün məlumatların təhlükəsizliyi və məxfilik qaydaları və standartları yaratmaq lazımdır.

Bu problemlər və maneələr ağıllı şəhərlərin gələcək inkişafına təsir göstərə bilər. Bununla belə, bu problemlərin həlli üçün həll yollarının yaradılması və maraqlı tərəflər arasında əməkdaşlıq ağıllı şəhərlərin inkişafına töhfə verə bilər.

AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

Əlyazma hüququnda

Məmmədov Rasim Famil oğlu
Məmmədli Ülvü Aftandil oğlu
Hümbətov Qara Ələkbər oğlu
Nəbiyev Şahin Ramin oğlu

**Airbnb və bənzəri turistik xidmətlər təklif edən layihələr və onların işləmə
mexanizmi**

mövzusunda

MAGİSTRİK DİSSERTASIYASI

İxtisas: 060632 – “İnformasiya texnologiyaları və sistemləri mühəndisliyi”

İxtisaslaşma: “Tətbiqi program təminatı”

Elmi rəhbər:

t.e.d., professor Əliyev Ələkbər Əliqə oğlu

BAKİ – 2023

II FƏSİL. AIRBNB VƏ BƏNZƏRİ TURİSTİK XİDMƏTLƏR TƏKLİF EDƏN LAYİHƏLƏR VƏ ONLARIN İŞLƏMƏ MEXANİZMİ

Airbnb (ABNB) evlərini icarəyə vermək istəyənləri müəyyən yerlərdə yaşayış yeri axtaran insanlarla birləşdirən onlayn bazardır. Şirkət 2007-ci ildən, həmtəsisçiləri pullu qonaqları qonaq otağında qalmağa dəvət etmək ideyasını ilk dəfə irəli sürdükdən sonra uzun bir yol keçmişdir. Airbnb-nin son məlumatlarına görə, dünyanın 100.000-dən çox şəhər və qəsəbəsini və 220-dən çox ölkəni əhatə edən altı milyondan çox xidmət təklifinə malikdir. Airbnb qonaqpərvərlik sənayesində inqilab etdi. 2008-ci ilə qədər səyahətçilər, ehtimal ki, başqa bir şəhərə səyahət etmək üçün otel və ya yataqxana sifariş etmiş olardılar. Hal-hazırda, eyni insanların çoxu Airbnb-ni seçir. Airbnb-nin arxasında duran fikir sadədir: Yerli insanlar üçün ehtiyat evlərini və ya otağını əraziyə gələn insanlara icarəyə verməklə əlavə pul qazanmağın bir yolunu tapın. Bu platformadan istifadə edən ev sahibləri, böyük bir şirkətin ödənişləri həyata keçirəcəyinə və lazım olduqda dəstək təklif edəcəyinə əminliklə dünya üzrə milyonlarla insana kirayələrini reklam edə bilirlər. Və qonaqlar üçün Airbnb daha çox xarakter daşıyan ev kimi qalmaq üçün yer təklif edə bilər, bəlkə də çöldə yemək yeməmək üçün mətbəxi olsa belə, çox vaxt otellərin ödədiyi qiymətdən aşağı qiymətə.

Qonaqlar üçün addım-addım istifadə qaydası:

Rezervasiya etmək üçün ümumiyyətlə aşağıdakı addımları atmalısınız:

- Airbnb veb saytını açın.
- Daxil olun və ya hesab yaradın, əgər bunu etməmişsinizsə. Qeydiyyatdan keçmək bir az vaxt apara bilər. Digər şeylərlə yanaşı, siz telefon nömrənizi doğrulamalı və şəxsiyyətin müəyyən formasını yükləməlisiniz.
- Axtardığınız yeri və tarixi (-ləri) göstərin və sonra axtarışınızı fərdiləşdirmək üçün saytın müxtəlif filtrlərindən istifadə edərək mövcud yeri axtarmağa başlayın.
- İdeal yaşayış yeri tapdığınız zaman rezervasiya edin və ya rezerv edin - bəzən ev sahibi qəbul edəne qədər rezervasiyalar tam təsdiqlənməyəcək.

- Yaşayış haqqını ödəyin və qalacağınız ünvan daxil olmaqla, rezervasiya haqqında bildiriş alın. Uzunmüddətli qalmaq üçün ilkin depoziti, qalanını isə hissə-hissə ödəyəcəyiniz bir ödəniş planı təşkil etmək mümkündür [10].

2.1. Bizim layihə necə özünü maliyyələşdirəcək və pul qazanacaq?

Bizim tətbiqin biznes modeli olduqca sərfəlidir. Şirkət, Uber, Lyft və başqaları kimi, paylaşma iqtisadiyyatından kapitallaşdıraraq, əslində sahib olmadığı əmlakı icarəyə verməklə pul qazanıb. Hər dəfə rezervasiya ediləndə tətbiqimiz endirim edir. Mülk üzərinə kliklədiyiniz zaman, səhifənin sağında siz davam edib rezervasiya etsəniz, sizdən tutulacaq ödənişlərin bölgüsü ilə tanış olacaqsınız. Bu ödənişlərdən biri platformanın idarə edilməsi və dəstəyin göstərilməsi xərclərini əhatə edən xidmət haqqıdır; bu, əsasən bizim tətbiqin gəlirinin əsas hissəsini təşkil edir. Şirkətin veb saytına görə, qonaqlara xidmət haqlarının əksəriyyəti rezervasiya məbləğinin 14,2%-dən azdır. Ev sahiblərindən isə hər rezervasiya üçün qazanılan ümumi məbləğin 3%-i və ya daha çoxu tutulur [2].

Bizim tətbiq həm də zaman-zaman görə biləcəyiniz alternativ ödəniş strukturunu idarə edir. Dünyanın müəyyən regionlarında yerləşən mehmanxanalar və bəzi xüsusi host növləri (o cümlədən proqram təminatı ilə əlaqəli olanlar) üçün xidmət haqqını qonaqlara bölmək əvəzinə tam ödəmələri məcburidir. Belə olduqda, yaşayış yeri təklif edən şəxsdən adətən rezervasiya məbləğinin təxminən 14%-16%-i tutulacaq. İstisnalara yalnız ev sahibi üçün ödənişin 10% səviyyəsində müəyyən edildiyi materik Çin daxildir.

2.2. Layihənin qarşısındakı əsas məqsədlər və həll edəcəyi əsas çətinliklər

Bizim tətbiq ölkəmizin böyük tətillər icarəsi platformalarından biri olmaq hədəfi ilə yola çıxmışdır. O, insanlara unikal və fərdiləşdirilmiş təcrübə təqdim etməklə, səyahət və yeni yerdə qalma tərzində inqilab etdi. Tətbiqimizin iş prosesi sadə və başa düşüləndir. Birincisi, əmlak sahibi öz əmlakını tətbiqimizdə siyahıya alır. Onlar otaqların sayı, vanna otağı və qonaqlar üçün mövcud şərait kimi əmlak haqqında təfərrüatları təqdim edirlər. Sahib həmçinin əmlakın qiymətini təyin edir və əmlakın ən yaxşı xüsusiyyətlərini nümayiş etdirmək üçün fotosəkillər çəkir. Daha sonra şəhərdə qalmaq üçün yer axtaran səyahətçilər tətbiqdə uyğun mülkləri axtara bilirlər. Onlar

axtarış nəticələrini məkana, qiymət diapazonuna, qonaqların sayına və əmlakın növünə görə filtrləyə bilər. Bəyəndikləri əmlakı tapdıqdan sonra rezervasiya detallarını müzakirə etmək üçün sahibi ilə əlaqə saxlaya bilərlər. Sahibi və qonaq rezervasiyadan razıdırsa, qonaq ödənişi bizim tətbiqin veb-saytı vasitəsilə həyata keçirə bilər. Platforma sahibi və qonaq arasında vasitəçi rolunu oynayır və hər iki tərəfin sifariş prosesi boyunca qorunmasını təmin edir. Qonaq əmlaka daxil olana qədər ödəniş bizim tətbiq tərəfindən tutulur. Bizim tətbiq insanların gündəlik həyatlarında bir sıra problemləri həll edir. Birincisi, bu, səyahət edənlər üçün sərfəli və çevik yaşayış seçimini təmin edir. Xüsusilə məşhur turizm yerlərində otellər bahalı ola bilər. Bizim tətbiq səyahətçilərə büdcələrinə və seçimlərinə uyğun qalmağa yer tapmağa imkan verir. Platforma həmçinin ənənəvi otel saytlarında mövcud olmayan mənzillər, villalar və ağac evləri kimi unikal yerləşdirmə variantları təklif edir. İkincisi, bizim tətbiq mülk sahiblərinə ehtiyat otaqlarını və ya bütün əmlaklarını icarəyə verməklə əlavə gəlir əldə etməyə imkan verir. Bu, ikinci ev kimi istifadə edilməmiş əmlakı olan, lakin onu satmaq istəməyən insanlar üçün xüsusilə dəyərli ola bilər. Bizim tətbiq onlara əmlaklarını səyahətçilərə icarəyə verməklə pul qazanmaq üçün platforma təqdim edir. Nəhayət, bizim tətbiq səyahətçilər üçün daha şəxsi və orijinal səyahət təcrübəsi təqdim edir. Yerlilərin evində qalmaq səyahətçilərə getdikləri şəhərin mədəniyyətini və həyat tərzini bir otelin təmin edə bilməyəcəyi şəkildə yaşamağa imkan verir. Bir çox ev sahibləri həmçinin şəhərdə görüləcək işlər üçün təkliflər və məsləhətlər verir, səyahətçilərə səyahətlərindən maksimum yararlanmağa kömək edir. Nəticə olaraq, bizim tətbiqin iş prosesi sadə və effektivdir və o, insanların gündəlik həyatında bir sıra problemlərin həllinə kömək edir. Səyahətçilər üçün sərfəli və unikal yaşayış variantları və mülk sahibləri üçün gəlir mənbəyi təqdim etməklə bizim tətbiq səyahət sənayesi üçün dəyərli platformaya çevrilib [1].

1. Əlverişli Yerləşdirmə: bizim tətbiq insanlara öz mülklərini səyahətçilərə icarəyə verməyə imkan verən onlayn əsaslı bazardır və ənənəvi otellərə sərfəli alternativ təqdim edir.

2. Orijinal Təcrübə: bizim tətbiq səyahətçilərə orijinal təcrübələrdən həzz almağa və yerli sakinlərlə əlaqə saxlamağa imkan verir. Yeni bir şəhərə və ya ölkəyə baş çəkmək

qorxuducu ola bilər, lakin bizim tətbiq ev sahibləri gizli daşlar tövsiyə edə, daxili məsləhətlər verə və ya hətta şəhər bələdçisi kimi çıxış edə bilər.

3. Fərdiləşdirmə: bizim tətbiq müxtəlif ehtiyaclara, üstünlüklərə və büdcələrə cavab verən xüsusiyyətlərə malikdir. Onlar müxtəlif ölçülər, yerlər və şərait təklif edir ki, bu da səyahətçilərə qalacaqlarını ehtiyaclarına, büdcələrinə və maraqlarına uyğun olaraq fərdiləşdirməyə imkan verir.

4. Təhlükəsizlik: bizim tətbiq həm ev sahibləri, həm də qonaqlar üçün təhlükəsiz və etibarlı platforma təmin edir. bizim tətbiq həm ev sahiblərinin, həm də qonaqların şəxsiyyətini yoxlayır və onların əməliyyatları üçün təhlükəsiz ödəniş sistemi təqdim edir.

5. Çeviklik: bizim tətbiq birbaşa ev sahibi ilə razılaşdırıla bilən giriş və çıxış vaxtları baxımından çeviklik təklif edir. Bu, tək saatlarda gələn və baqajlarını hardasa saxlamalı olan səyahətçilər üçün vaxta qənaət edir.

6. Davamlılıq: bizim tətbiq səyahətçiləri turistlərin qaynar nöqtələrində deyil, yerli məhəllələrdə qalmağa təşviq etməklə davamlılığını təşviq edir, bu da yerli biznesi dəstəkləyir və karbon izlərini azaldır.

7. Yerli İcmalara Dəstək: bizim tətbiq yerli sakinlərə iqtisadi səmərə verməklə yerli icmaları dəstəkləməkdə mühüm rol oynayır. Ev sahibləri əlavə gəlir əldə edirlər və səyahətçilər yerli restoranlarda yemək yeyərək, yerli mağazalarda alış-veriş etməklə və yerli nəqliyyatdan istifadə etməklə yerli iqtisadiyyatı stimullaşdırırlar.

8. Qrup Yerləşdirməsi: bizim tətbiq ailələr kimi səyahətçi qruplarını yerləşdirə bilən daha böyük müklər təklif edir. Bir çox hallarda, birdən çox otel otağı sifariş etməkdənsə, daha böyük bizim tətbiq əmlakını icarəyə götürmək daha sərfəli olur.

9. Heyvanlar üçün uyğun yerləşdirmə: Bir çox bizim tətbiq ev sahibləri ev heyvanlarını qarşılayır, bu, tüklü dostlarını tərk etmək istəməyən səyahətçilər üçün vacib amil ola bilər. Heyvanlar üçün əlverişli bizim tətbiq ailə təttilini tamamlaya bilər.

10. Əlçatanlıq: bizim tətbiq əlil arabası üçün əlçatanlıq kimi xüsusi əlçatanlıq ehtiyacları olan səyahətçilərə xidmət göstərir. Onlar səyahətçilərə öz ehtiyaclarına cavab verən mükləri tapmaqda kömək etmək üçün filtrlər təqdim edirlər ki, bu da hər kəsə qabiliyyətlərindən asılı olmayaraq səyahət etməkdən həzz almağa imkan verir.

2.3. Layihənin istifadəçilərinə və ağıllı şəhər ekosisteminə təmin etdiyi üstünlüklər

Bizim tətbiq ənənəvi mehmanxana və yerləşdirmə seçimlərinə alternativ təqdim edərək insanların səyahət tərzində inqilab etdi. bizim tətbiq-dən istifadə etməyin ən böyük üstünlüklərindən biri unikal və sərfəli seçimlərin geniş çeşididir. Səyahətçilərin mənzillər, villalar və hətta ev gəmiləri də daxil olmaqla müxtəlif növ evlərə çıxışı var ki, bu da ziyarətçilərə tipik bir otel otağında qalmaqla müqayisədə fərqli bir təcrübə təqdim edə bilər. bizim tətbiq həmçinin səyahətçilərə adətən hesab etmədikləri məhəllələrdə və icmalarda qalmağa imkan verir. Bundan əlavə, bizim tətbiq qiymətləri, xüsusən də qrup halında və ya uzun müddətə səyahət edərkən ənənəvi otellərdən daha ucuz olur. Bizim tətbiq ev sahibləri həmçinin öz yerli əraziləri haqqında dəyərli insayder məlumat verir və ziyarət etmək üçün ən yaxşı restoranları, barları və attraksionları tövsiyə edə bilərlər. Bundan əlavə, bizim tətbiq orijinal təcrübələr üçün imkanlar yaradaraq yerli mədəniyyətlə əlaqə hissi yaradır. Nəhayət, bizim tətbiq-dən istifadə etmək asandır və sifariş və qeydiyyat vaxtları baxımından rahatlıq təmin edir. Ümumiyyətlə, bizim tətbiq insanların dünyanı təcrübədən keçirməsi və unudulmaz səyahət xatirələri yaratması üçün unikal və sərfəli üsul yaratmışdır [1].

Geniş seçim: bizim tətbiq ev sahibləri tək otaqlar, mənzillər və evlərdən tutmuş qayıqlara, karvanlara və hətta qalalara qədər bir çox müxtəlif əmlak növlərini siyahıya alır.

Pulsuz siyahılar: Ev sahibləri əmlaklarını siyahıya almaq üçün pul ödəməli deyillər. Siyahılara yazılı təsvirlər, başlıqları olan fotosəkillər və potensial qonaqların ev sahibləri haqqında bir az tanış ola biləcəyi istifadəçi profili daxil ola bilər.

Ev Sahibləri Öz Qiymətlərini Müəyyən edə bilərlər: Gecə, həftəlik və ya ayda nə qədər ödəniş edəcəyini hər bir ev sahibi qərar verə bilər.

Fərdiləşdirilə bilən Axtarışlar: Qonaqlar bizim tətbiq verilənlər bazasında axtarış edə bilərlər - yalnız tarix və yerə görə deyil, qiymətə, əmlakın növünə, şəraitə və ev sahibinin dilinə görə - və xüsusiləşdirməyə kömək etmək üçün açar sözlər ("Qız Qalasına yaxın" kimi) əlavə edə bilərlər. onların axtarışı.

Əlavə Xidmətlər: bizim tətbiq təkliflərinə təcrübələr və restoranlar daxildir. Məkan üzrə axtarış edən insanlar yerli bizim tətbiq ev sahibləri tərəfindən təklif olunan dərsələr və görməli yerləri görmək kimi təcrübələrin siyahısını görəcəklər. Restoran siyahılarına bizim tətbiq ev sahiblərinin rəyləri də daxildir.

Qonaqlar və Ev Sahibləri üçün Mühafizələr: bizim tətbiq hər qonağın ödənişini qeydiyyatdan sonra 24 saat ərzində saxlayır və pulu ev sahibinə verir. Ev sahibləri üçün bizim tətbiq əmlaka dəymiş əsassız ziyanı ödəmək üçün 1.000.000 dollara qədər pul verir. Bu qoruma heç bir əlavə xərc tələb etmir, lakin hər şeyi əhatə etmir.

2.4. Layihənin rəqiblərindən üstün cəhətləri

Aşağıda verilən nüanslar diqqətə alınacaq kodlaşdırma mərhələsində. Bizim tətbiqin əsas məqsədi daha təkmilləşdirilmiş və istifadəçilərə daha rahat və sadə istifadəçi təcrübəsi təqdim etməkdir [9].

1. Doğrulama prosesi: Airbnb-də yoxlama prosesi mövcud olsa da, həm qonaqların, həm də ev sahiblərinin təhlükəsizliyini təmin etmək üçün o, gücləndirilə bilər.
2. Müştəri xidməti: Bəzi istifadəçilər sifarişləri ilə işlərin pis getdiyi zaman müştəri xidmətində çətinlik çəkdiklərini bildirdilər. Airbnb bu vəziyyətlərdə cavab müddətlərini və təcrübələrini yaxşılaşdırma bilər.
3. Axtarış funksiyası: Airbnb-nin axtarış funksiyası istifadəçi hər dəfə siyahı axtarıqda daha dəqiq nəticələr vermək üçün təkmilləşdirilə bilər.
4. Ləğv siyasətləri: Ev sahibləri və qonaqlar tez-tez müxtəlif ləğv siyasətlərinə giriş əldə edirlər, bu da ehtiyac yarandıqda çətinliklərə səbəb ola bilər. Airbnb daha aydın və standartlaşdırılmış siyasətlər yarada bilər.
5. Təmizlik standartları: Yüksək təmizlik standartlarına riayət etmək lazımdır və Airbnb ev sahiblərinin evdə təmizlik və hazırlıq protokollarına əməl etməsinə və tətbiq etməsinə əmin ola bilər.
6. Qiymətləndirmə: Bəzi siyahılar həddən artıq baha görünə bilər və Airbnb mövsümlilik, yer və şərait kimi amilləri nəzərə alaraq daha dəqiq qiymətlər təqdim edə bilər.
7. Şəffaflıq: Bəzi istifadəçilər gizli ödənişləri bildirdilər və onlar sifariş zamanı göstərilən təsvirlərə uyğun əmlak tapmadılar. Airbnb bütün qiymətləri aydın şəkildə

göstərməklə və siyahıda mövcud olan dəqiq imkanları təfərrüatlandırmaqla istifadəçilər üçün şəffaf ola bilər.

2.5. Metodologiya

Project Management - metodologiyalarının istifadəsi akademik tezislərin yazılması xeyli vaxt, səy və vəsait tələb edən mürəkkəb bir prosesdir. Tezis illərin tədqiqi və tədqiqatının kulminasiya nöqtəsini təmsil edir və akademik təhsilin kritik tərkib hissəsidir. Akademik dissertasiyanın yazılmasında layihənin idarə edilməsinin rolunu qiymətləndirmək olmaz, çünki o, layihənin uğurla başa çatdırılmasında və bütün zəruri tələblərə cavab verməsində həlledici rol oynayır. Akademik tezis üçün layihənin idarə edilməsində ilk addım layihənin əhatə dairəsini müəyyən etməkdir. Buraya tədqiqata rəhbərlik edəcək tədqiqat suallarının, istifadə olunacaq metodologiyanın və tamamlanma müddətinin müəyyən edilməsi daxildir. Bu mərhələdə düzgün planlaşdırma layihənin bütün proses boyu diqqət mərkəzində və yolda qalmasını təmin etməyə kömək edəcək. Bundan əlavə, hər bir komanda üzvünün rol və məsuliyyətlərini müəyyən etmək və hər kəsin layihədə öz rolundan xəbərdar olmasını təmin etmək üçün aydın ünsiyyət kanalları yaratmaq vacibdir [8].

Əhatə dairəsi müəyyən edildikdən sonra layihə meneceri layihənin hər bir mərhələsi üçün tapşırıqları, son tarixləri və mərhələləri əks etdirən ətraflı layihə cədvəli yaratmalıdır. Layihənin gedişatı haqqında hər kəsi məlumatlandırmaq üçün bu cədvəl bütün komanda üzvlərinə və maraqlı tərəflərə çatdırılmalıdır. Akademik tezislərin yazılmasında layihə menecmentinin digər mühüm rolu risklərin idarə edilməsidir. Risklərə maliyyə çatışmazlığı, məlumat itkisi və ya tamamlanmada gecikmələr kimi məsələlər daxildir. Layihə meneceri potensial riskləri müəyyən etməli və onları azaltmaq üçün fəvqəladə hallar planları yaratmalıdır. Planlaşdırma və risklərin idarə edilməsi ilə yanaşı, layihənin idarə edilməsi keyfiyyətə nəzarətin təmin edilməsində mühüm rol oynayır. Layihə meneceri həm proses, həm də son məhsul üçün standartlar və təlimatlar yaratmalıdır. Buraya tədqiqat etikası, məlumatların təhlili və yazı üslubu üçün təlimatlar daxil ola bilər. Layihə meneceri bu standartların yerinə yetirilməsini təmin etmək üçün irəliləyişə nəzarət etməli və lazım gəldikdə düzəlişlər etməlidir.

Nəhayət, layihənin idarə edilməsi akademik dissertasiyanın verilən vaxt daxilində tamamlanmasını təmin etmək üçün mühüm rol oynayır. Layihə meneceri layihənin yolunda qalmasını təmin etmək üçün layihə cədvəlini mütəmadi olaraq nəzərdən keçirməlidir. Lazım gələrsə, layihənin vaxtında və büdcə daxilində tamamlanmasını təmin etmək üçün qrafikə düzəlişlər edilə bilər.

Yekun olaraq, akademik dissertasiyanın yazılmasında layihənin idarə edilməsinin rolunu qiymətləndirmək olmaz. Layihənin düzgün idarə edilməsi layihənin diqqət mərkəzində olmasını, yaxşı planlaşdırılmasını və müəyyən edilmiş standartlara uyğun tamamlanmasını təmin edir. O, həmçinin riskləri azaltmağa, keyfiyyətə nəzarəti təmin etməyə və layihənin vaxtında və büdcə daxilində tamamlanmasını təmin etməyə kömək edir. Müvafiq layihə idarəetmə üsulları və yanaşmaları ilə akademik tezis daha asan şəkildə fəth edilə bilər.

2.6. İstifadə olunan texnologiyalar

Ağıllı Şəhər konsepsiyası üzrə yaradacaq olduğumuz proyekt veb layihədir və bu layihənin hazırlanması prosesi ərzində istifadə olunacaq metodologiya və tətbiqlərdən söz açmaq önəmli bir nüansdır. Bu hissədə veb layihəni hazırlayarkən istifadə etdiyimiz texnologiyalar və onların istifadə sahəsi haqqında məlumatlar yer alacaqdır. Veb layihə frontend (ön, klient tərəfi) və backend (arxa, server tərəfi) hissələrindən ibarətdir. Hər bir hissəsində də fərqli texnologiyalar istifadə olunmuşdur. Proqramlaşdırma dilləri olaraq python və javascript, framevörk olaraq fastapi və react tətbiq edilmişdir [8].

Python - Python geniş istifadəçi bazasına malik yüksək səviyyəli proqramlaşdırma dilidir. Sadə və oxunaqlı sintaksisi sayəsində kodu tez və effektiv yazmağa imkan verir. Modul quruluşu ilə Python geniş kitabxana və alətlər ekosisteminə malikdir, buna görə də müxtəlif sahələrdə olan problemlərin həll olunmasında istifadə olunur. Həm əməliyyat sistemləri, həm də platformalar arasında daşınma qabiliyyətini dəstəkləyən Python veb inkişafı, məlumat analitikası, süni intellekt və maşın öyrənməsi kimi bir çox sahədə populyardır. O, həm də böyük bir cəmiyyət tərəfindən dəstəklənir və daim inkişaf etdirilən bir dildir.

Fastapi - FastAPI Python ilə hazırlanmış müasir veb freymvörkdür. Yüksək performanslı və sürətli işləyən asinxron quruluşu ilə diqqət çəkir. O, sürətli program inkişaf etdirməyə imkan verən sadə və oxunaqlı sintaksisə malikdir. FastAPI avtomatik olaraq API sənədləri yarada, məlumatları strukturunu yoxlaya bilən Pydantic ilə inteqrasiya edir. Bu yolla o, güclü məlumatların yoxlanılması və serializasiya imkanları təklif edir. Asinxron dəstək miqyaslılığı təmin edir və yüksək trafikli tətbiqlər üçün uyğun bir həll təqdim edir. Həmçinin, FastAPI açıq mənbə kodlarından ibarətdir və icma tərəfindən dəstəklənən böyük bir layihədir.

Pydantic - Pydantic, Python dilində istifadə olunan məlumatların yoxlanılması və serializasiya kitabxanasıdır. Bu kitabxana məlumat modellərini təyin edərkən statik növləri dəstəkləyir və avtomatik olaraq məlumatların yoxlanılması, növün yoxlanılması və serializasiya/deserializasiyanı həyata keçirir. Pydantic məlumatların yoxlanılması xüsusiyyətləri ilə ardıcıl məlumat bütövlüyünü təmin edir və səhv məlumat daxil edilməsinin qarşısını alır. Həmçinin, Python ilə yazılan layihələrə asanlıqla inteqrasiya oluna bilir və Python ekosistemində geniş istifadə olunan kitabxanadır.

Sqlalchemy - SQLAlchemy, Python proqramlaşdırma dili üçün hazırlanmış ORM (Object-Relational-Mapper) kitabxanasıdır. Verilənlər bazası idarəsi və qarşılıqlı əlaqə üçün istifadə olunur. SQLAlchemy, əlaqəli verilənlər bazalarının obyekt yönümlü şəkildə istifadəsini təmin etməklə verilənlər bazası əməliyyatlarını asanlaşdırır. Declarative API sayəsində verilənlər bazası cədvəllərini Python siniflərinə çevirərək verilənlər bazası əməliyyatlarını daha oxunaqlı və başa düşülən edir. SQLAlchemy verilənlər bazası müstəqilliyini dəstəkləməklə müxtəlif verilənlər bazası sistemləri ilə işləyə bilər. Güclü sorğu imkanları verilənlər bazası əməliyyatlarını yerinə yetirməyi asanlaşdırır və daha səmərəli edir. SQLAlchemy böyük bir istifadəçi icması tərəfindən dəstəklənir və geniş yayılmış verilənlər bazası vasitəsidir.

Alembic - Alembic Python-da istifadə olunan verilənlər bazası köçürmə vasitəsidir. Verilənlər bazası sxemlərinin idarə edilməsi və saxlanması üçün istifadə olunur. Alembic verilənlər bazası miqrasiyasını idarə etmək üçün sadə və intuitiv API təqdim edir. Dəyişiklikləri izləmək, yeniləmələri tətbiq etmək və geri qaytarmaq üçün

istifadəçi yönümlü bir yol təqdim edir. Alembic miqrasiya ssenarilərini saxlamaq üçün versiya sistemə malikdir. O, həmçinin çoxlu verilənlər bazası mühərrikləri ilə uyğun gəlir və geniş verilənlər bazası sistemlərini dəstəkləyir. Alembic bir çox Python əsaslı layihələrdə geniş istifadə olunan verilənlər bazası köçürmə vasitəsidir.

Postgresql - PostgreSQL açıq mənbəli verilənlər bazası idarəetmə sistemidir. Etibarlılığa, məlumatların bütövlüyünə və performansla diqqət yetirməklə hazırlanmışdır. PostgreSQL geniş xüsusiyyətlər dəstinə malikdir və ACID (atomicity, consistency, isolation, durability) uyğun əməliyyatları dəstəkləyir. O, mürəkkəb sorğuları səmərəli şəkildə emal edə və yüksək performanslı böyük verilənlər toplularını idarə edə bilər. Bundan əlavə, PostgreSQL genişlənmə və yüksək əlçatanlıq üçün müxtəlif replikasiya və ehtiyat nüsxə (backup) üsullarını dəstəkləyir. PostgreSQL böyük bir icma tərəfindən dəstəklənən verilənlər bazası sistemidir və daim inkişaf etdirilir. Açıq mənbə quruluşu, rahatlığı və sərbəst istifadəsi ilə üstünlük verilən verilənlər bazası həllidir.

Docker - docker konteyner əsaslı proqram platformasıdır. Tətbiqləri konteynerlər şəklində yığmaq və yaymaq üçün istifadə olunur. Docker proqramların daşınma qabiliyyətini artırır və onların müxtəlif əməliyyat sistemləri və platformalarında problemsiz işləməsini təmin edir. Konteynerlər təcrid olunmuş mühitlərdə işləyir və proqramın bütün asılılıqlarını ehtiva edir, bu da onun lazım olan şəkildə işləməsini təmin edir. Docker sürətli yerləşdirmə, miqyaslaşdırma və yüksək səmərəlilik təklif edir. Siz həmçinin Docker konteynerlərini asanlıqla konfigurasiya edə, idarə edə və paylaşa bilərsiniz. Docker müasir proqram təminatının işlənilməsində geniş istifadə olunan bir vasitədir və böyük bir icma tərəfindən dəstəklənir.

Github - Git kodu izləmək, dəyişiklikləri idarə etmək və müxtəlif versiyaları saxlamaq üçün proqram təminatının hazırlanması prosesində istifadə olunur. Git birdən çox tərtibatçının eyni layihə üzərində birlikdə işləməsini asanlaşdırır və dəyişiklikləri izləməyə imkan verir. Hər bir dəyişiklik ayrıca "commit" kimi saxlanılır və layihənin müxtəlif nöqtələrinə qaytarıla bilər. Git budaq (branch) və birləşmə (merge) xüsusiyyətləri ilə müxtəlif inkişaf axınlarını dəstəkləyir. Git sürətli, çevik və güclü

versiyaya nəzarət sistemi kimi məşhurdur və geniş istifadəçi icması tərəfindən dəstəklənir. GitHub isə Git əsaslı layihələrin yerləşdiyi platformadır. Git local kompüterdə layihənin versiyaya nəzarətini təmin edir və dəyişiklikləri izləyir. GitHub, digər tərəfdən, bu Git repository (anbar) üçün bulud əsaslı paylaşılan və əməkdaşlıq platformasıdır. GitHub, layihələri paylaşmaq, əməkdaşlıq etmək və icmanın iştirakını təşviq etmək üçün tərtibatçılar arasında geniş istifadə olunan xidmətdir.

Veb layihələr ilk yarandığı dövrlərdə, ancaq istifadəçilərin məlumatları oxumaq üçün istifadə edəcəkləri formada təyinatlanmışdı. Daha sonra isə daha dinamik hal almağa başlayan veb saytlar durmadan artan məlumatlar və istifadə yönündə yaranan ehtiyac və tələblər sayəsində günü gündən daha da mürəkkəbləşmişdir. Artıq veb layihələr tək bir subyektin əli ilə deyil bir komandanın müştərək çalışması ilə ərsəyə gəlir. Veb layihənin hazırlanmasının texniki prosesi adətən: Frontend və Backend üzrə ixtisaslaşmış kadrların sinxron çalışmasından ibarətdir. Bəs nədir bu frontend və backend?

Backend – özünü layihənin dinamik strukturunda büruzə verir. Serverlərin yaradılması, CRUD əməliyyatları (Create, Read, Update, Delete - məlumatların istifadəçi və ya adminin idarəsi ilə dinamik olaraq əlavə edilməsi, verilənlər bazasından lazımi məlumatların gətirilməsi, var olan cari məlumatın yenilənməsi, redaktə edilməsi və silinməsi), routing və s. Backendə – C#, C++, Java, Python, Ruby, Golang, NodeJS kimi proqramlama dillərini misal göstərə bilərik.

Frontend – Veb layihənin və ya tətbiqin istifadəçiyə görünən hissəsinin kodlanmasından, backend-dən gələn datanın idarə edilməsi işlərindən ibarətdir. Əgər saytımızın dinamikliyə ehtiyacı yoxdusa, backend-ə ehtiyac olmadan sadəcə frontend alətləri ilə statik saytlar hazırlamaq mümkündür. Frontend-də istifadə olunan dillər və alətlər: HTML, CSS, Javascript (framework-ləri və s.) [4, 5, 6].

HTML (Hypertext Marking Language) – bir proqramlaşdırma dili deyil, hipermətnləri işarələmə dilidir. Veb saytların ümumi strukturu, elementləri HTML kodları vasitəsi ilə hazırlanır.

CSS (Cascading Style Sheets) - saytı daha gözoxşayan etmək və istifadəsini rahatlaşdırmaq üçün CSS dilindən istifadə edəcəyik. CSS kodlarını HTML faylına

inteqrasiya etmək üçün 3 üsul mövcuddur. “Inline CSS” – kodlar cari HTML elementinin “style” atributuna dəyər verməklə dizaynı dəyişdirmək olur. “Internal CSS” – HTML faylının “head” teqləri daxilində style elementinin arasında CSS kodları əlavə edilir. “External CSS” – xarici bir CSS faylına HTML link elementi vasitəsilə istinad edilir. Orta və böyük layihələrdə ən optimal metod sonuncu versiyadır.

JavaScript – veb səhifənin dinamikliyi, istifadəçinin yerinə yetirdiyi əməliyyatların icrası üçün lazım olan funksiyalar, serverdən verilənlərin çəkilməsi və s. üçün bizə gərəkli olan proqramlaşdırma dilidir. JavaScriptin imkanları yuxarıda qeyd olunanlarla bitmir. JavaScript, veb proqramçıların veb səhifələri, serverləri, veb tətbiqləri və hətta oyunları da yaradarkən daha dinamik qarşılıqlı əlaqə yaratmaq üçün istifadə etdiyi sinxron və single-thread proqramlaşdırma (skript) dilidir. Yəni bu dil işləmə prinsipi olaraq sətir-sətir çalışıb, daha üstdəki prosesləri icra etmədən aşağıdakı funksiyalara keçmir. Javascript Google V8 motorunu istifadə edərək brauzerdə çalışa bilən yeganə proqram dilidir. Əlavə olaraq Javascriptin Node.JS frameworku ilə server tərəfində çalışan backend proyektləri də yaratmaq mümkündür. Günümüzdə inkişaf etdirilən veb layihələrin həcmi çox olduğu üçün, uzun Javascript kodları yazıb vaxt itirməkdən bizi xilas edən bir çox JS (Javascript) kitabxanaları var. React JS, Angular, JQuery, Svelte, Vue JS və s. bunlara misaldır. Bu kitabxanalar ilə Javascript prinsipi ilə çalışan, sintaktik olaraq asanlaşmış kodlar yazmaq mümkündür. Biz bu proyektde React JS kitabxanasından istifadə edəcəyik. Bu səbəbdən React JS nədir, iş prinsipi nədir, üstün cəhətləri nədir bunlara nəzər salmaq [5].

React JS - Facebook şirkəti tərəfindən böyük proyektləri inkişaf etdirərkən standartlara uyğun iş mühiti yaratmaq və vaxt qazanmaq üçün yaradılan, istifadəçi interfeysləri yaratmaq üçün Javascript kitabxanasıdır. Niyə React? React JS stackoverflow.com saytının 2021-2022-ci il statistikasına görə Asp.Net frameworkündən sonra proqramçılar tərəfindən ən çox sorğu aparılan, ən çox maraq göstərilən ikinci veb frameworkudur. Asp.Net backend texnologiyası olduğu üçün, rahatlıqla deyə bilərik ki frontend tətbiqləri üçün hal hazırda ən uyğun seçim React kitabxanasıdır [6].

React bizə çalışmaq üçün HTML və Javascript faylları ilə əlaqələnmiş hazır iş mühiti təqdim edir ki, bu imkanlar bizə həm vaxt qazandırır həm də yazdığımız kodun daha modul formada daha anlaşılabilən və standartlara uyğun olmasını təmin edir. Reactın təməl prinsipi JavaScript kodları istifadə edərək, veb səhifədə JSX formatlı komponentlər render etməkdir. JSX (Javascript XML) formatı bizə səhifə yeniləməyə ehtiyac duymadan, səhifədəki elementlərin dəyişilməsi, fərqli səhifələrə yönləndirmə kimi üstünlüklər verir. Amma brauzerlər sözügedən JSX formatlı kodları oxumaqda aciz qalır. Onda belə bir sual ortaya çıxır: bəs bu kodlar necə işə düşür?

Babel – Javascript ES6, ES7 standartlarında yazılan kodları, həmçinin React kodlarını brauzerin oxuya biləcəyi ES5 standartına çevirə bilən bir translyasiya alətidir. React üçün iş mühiti yaradılarkən “node modules” adlı qovluq içində hazır babel konfiq faylları yüklü olur. Daha öncə qeyd etdiyimiz kimi React, JavaScript kodlarından istifadə edir, yəni JavaScript kodlarının içində XML kodları yazırıq (qısaca JSX). Biz brauzerdə ancaq Javascript kodlarının çalışdığını desək də, Javascriptin də çatışmayan bəzi cəhətlərinin olduğunu unutmayaq. Bu zəif cəhətlərdən biri də dəyişən və ya sabit dəyərlərin data tipinin təyin edilməsidir. Sadə bir misal olaraq, yanlışlıqla müştəri tərəfindən daxil olan “string” tipli verilən üzərində riyazi əməliyyat aparmağa çalışsaq bir çox xəta ilə üzləşə bilərik ki, bu xətanın təsbit edilməsi bəzən saatlarla vaxt alır. Bu problemə qarşılıq olaraq köməyimizə Typescript proqram dili çatacaq. Bəs nədir bu Typescript?

Typescript - Javascriptin bütün xüsusiyyətlərini özündə saxlayan və bizə daha anlaşılabilir və standart qəlibə uyğun güclü kod yazmağa yarayan proqramlama dilidir. Sintaks olaraq JavaScriptə bənzər bənzəsə də özündə bir neçə yenilik sığdırır. Bu fərqlər nələrdir?. Obyekt yönümlü proqramlamaya uyğundur, yəni klasslar, modullar, miras və s. özəllikləri dəstəkləyir.

Kompilyasiya - Javascript interpretasiya əsaslı dildir, yəni tərtib etmə mərhələsinə sahib deyil. Bu səbəbdən bizim kodumuz çalışmadığı müddətcə proqramda səhvlərin və ya xətalara olub olmamasını aşkar edə bilmirik. Kod çalışdıqda isə xəta çıxarsa, daha əvvəl qeyd etdiyimiz kimi o xətanı axtarmaq xeyli vaxt apara bilər. Typescript ilə çalışarkən isə belə hal yaşanmır. Typescript kodları real

vaxt ərzində koddakı xətalara aşkar edib proqramçıya bildirir. Daha öncə qeyd etmişdik ki, brauzerdə ancaq Javascript kodları çalışır. Bəs Typescript kodların brauzer üzərində necə çalışdırmalıyıq?. Typescript kodlarını brauzer üzərində çalışdırmaq üçün Webpack adlanan alətdən istifadə edilir. Webpack node paket meneceri (npm) vasitəsilə komputerdə qlobal olaraq yüklənir və Typescript faylını bu texnologiya vasitəsilə JS formatına çevirmək mümkündür.

Bu sadaladığımız bütün bu texnologiyaları kompleks şəkildə gətirmək üçün Vite adlanan alətin köməyinə ehtiyacımız var. Vite ilə React faylını yaratmaq üçün aşağıdakı addımları icra etmək lazımdır.

1. Əvvəlcə işlərimizi icra edəcəyimiz qovluğun içinə keçib, terminalı açırıq: terminalda “npm create vite” komandasını icra etdikdən sonra fayl adını qeyd edib framework olaraq Reactı seçirik (Şəkil 1).

```
PS C:\Users\ASUS\Desktop\Smart City> npm create vite
✓ Project name: ... ./
✓ Package name: ... smart-city
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
  Vue
>  React
  Preact
  Lit
  Svelte
  Others
```

Şəkil 1. Texnologiya təklif edilən terminal pəncərəsi.

Daha sonra isə template olaraq Typescripti seçib iş mühitinin şablonunu yaradıırıq.

2. Birinci addımı tamamilən icra etdikdən sonra terminalda iki əmr yazırıq (Şəkil 2).

```
# smart-city adlı qovluğa daxil olun
cd smart-city

# Projeğin tələblərini yükləyin
npm install

# Projeği Visual Studio Code ilə açın
code .

# Projeği yerli serverdə başlatmaq üçün:
npm run dev
```

Şəkil 2. Layihənin yüklənməsinə kömək edən əmrlər.

React – Typescript faylının strukturu. React faylının strukturu ümumi olaraq aşağıdakı qaydadadır. Faylımızda yerləşən index.html bizim kök (root) faylımızdır. Bütün proseslər məhz bu fayl üzərindən aparılır. Belə ki ekranda göstərilən bütün elementlərə, index.html içərisindəki “root” id-li element sahiblik edir. Proyektin dinamikliyindən cavabdeh olan main.ts faylında yuxarıda qeyd edilən root selektorlu element seçilib və App adlı react komponentini render edir (Şəkil 3) [5, 6]:

```
src > TS main.tsx
1  import React from 'react'
2  import ReactDOM from 'react-dom/client'
3  import App from './App.tsx'
4  import './index.css'
5
6  ReactDOM.createRoot(
7    document.getElementById('root') as HTMLElement
8  ).render(
9    <React.StrictMode>
10   <App />
11 </React.StrictMode>,
12 )
```

Şəkil 3. Layihənin “kök” (root) strukturu.

Stilləndirmə olaraq isə ən məşhur CSS kitabxanalarından biri olan Tailwind-dən istifadə edəcəyik. Tailwind – HTML elementlərinin class selektorunun köməyi ilə daha az CSS kodu yazaraq istədiyimiz dizaynda veb səhifəmizi tərtib etməyə yardım edir. Tailwindi proyektimizə inteqrasiya etmək üçün aşağıdakı addımlar icra edilir:

1. Cari qovluq içində terminalı açıb: `npm install tailwindcss@latest` komandası vasitəsilə tailwindi faylımıza əlavə edirik
2. `tailwind.config.js` faylının içində aşağıdakı konfigurasiya əməliyyatlarını icra edək (Şəkil 4):

```
// tailwind.config.js
const defaultTheme = require('tailwindcss/defaultTheme')

module.exports = {
  theme: {
    extend: {
      fontFamily: {
        sans: ['Inter var', ...defaultTheme.fontFamily.sans],
      },
    },
  },
  // ...
}
```

Şəkil 4. Tailwind kitabxanasının konfigurasiya edilməsi.

AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ

AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ

Əlyazma hüququnda

Məmmədov Rəsim Fəmil oğlu
Hümbətov Qara Ələkbər oğlu
Nəbiyev Şahin Rəmin oğlu

**Airbnb və bənzəri turistik xidmətlər təklif edən layihələr və onların işləmə
mexanizmi**

mövzusunda

MAGİSTRİK DİSSERTASİYASI

İxtisas: 060632 – “İnformasiya texnologiyaları və sistemləri mühəndisliyi”

İxtisaslaşma: “Tətbiqi proqram təminatı”

Elmi rəhbər:

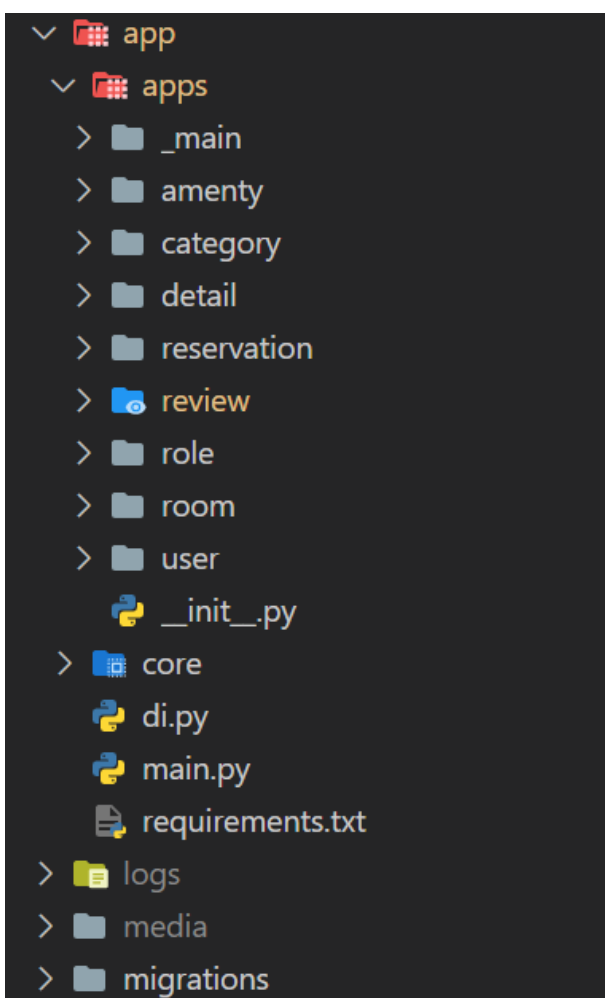
t.e.d., professor Əliyev Ələkbər Əliqə oğlu

BAKİ – 2023

III FƏSİL. TƏTBİQİN HAZIRLANMASI

3.1. Backend (server tərəfi)

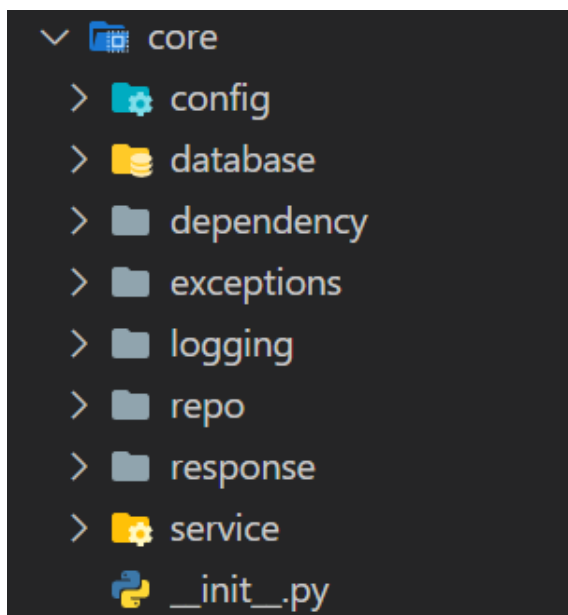
Qovluq strukturu - tətbiqin qovluq strukturu "app", "logs", "media" və "migrations" kimi bir neçə qovluqdan ibarətdir. Qovluq strukturu aşağıdakı şəkildə ətraflı (Şəkil 5) təsvir olunmuşdur. Ümumilikdə, bu qovluq strukturu müxtəlif komponentlər, əsas funksionallıq, qeydlər, media faylları və verilənlər bazası miqrasiyaları üçün ayrı-ayrı qovluqlarla proqramın kod bazasının modul təşkilini göstərir.



Şəkil 5. Server tərəfinin qovluq strukturu.

"app" qovluğunun içərisində iki əsas alt qovluq var: "apps" və "core". "apps" qovluğunda əlavə olaraq "_main", "amenty", "category", "detail", "reservation", "review", "role", "room" və "user" kimi alt qovluqları var. "apps" qovluğundakı "amenty", "category", "detail", "reservation", "review", "role", "room" və "user" alt

qovluqları, tətbiqin xüsusi funksiyalarını və ya xüsusiyyətlərini təmsil edir, kateqoriyaların, rezervasiyaların, rəylərin, rolların, otaqların və istifadəçilərin idarə edilməsi kimi funksionallıqları mövcuddur (Şəkil 6).



Şəkil 6. Core alt qovluğunun strukturu.

"app" qovluğundakı "core" qovluğu "config", "database", "dependency", "exceptions", "logging", "repo", "response" və "service" kimi alt kataloqlardan ibarətdir. Bu alt kataloqlar, tətbiqin əsas komponentlərini, konfigurasiyalarını və utilitlərini saxlayır. Bu komponentlər bütün tətbiq boyunca istifadə olunduğu üçün xüsusi core yəni nüvə adlı qovluqda toplanmışdır [8].

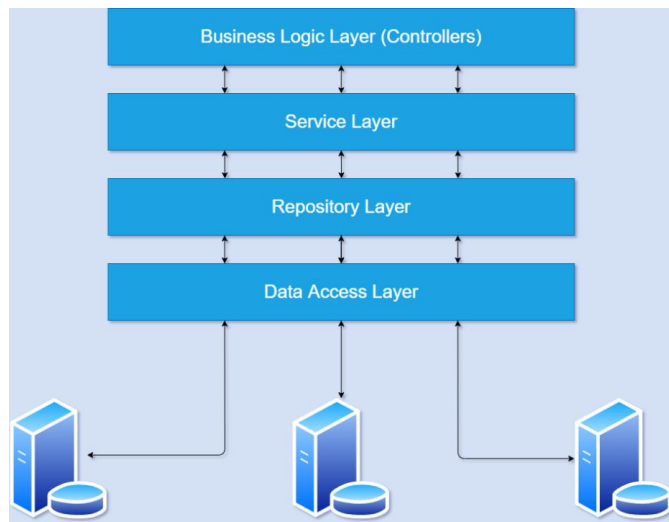
"logs" qovluğu ayrıca qovluq kimi mövcuddur, proqramın işləmə vaxtı ərzində yaratdığı jurnal fayllarını (loq mesajlarını) saxlamaq üçün nəzərdə tutulub.

"media" kataloqu, tətbiqə aid şəkillər, videolar və ya sənədlər kimi media fayllarını saxlamaq üçün nəzərdə tutulub.

"migration" kataloqunda "versiyalar" adlı alt kataloq var. Bu qovluğun altında verilənlər bazasının dəyişmələrinin hər bir versiyası saxlanılır. Bununla da ehtiyac olarsa daha verilənlər bazasının əvvəlki versiyasına asanlıqla qayıtmaq mümkündür.

3.2. Repository service pattern (repository və servis ilə dizayn)

Repository Service Pattern proqramda verilənlərə girişi və davamlılığını idarə etməyə kömək edən proqram dizayn nümunəsidir. Bu nümunə tətbiqin biznes məntiqi ilə əsas məlumat saxlama mexanizmi (adətən verilənlər bazası) arasında abstraksiya qatını təmin edir. Bu dizayn nümunəsi verilənlərə çıxış üçün cavabdeh olan komponentləri biznes məntiqindən ayıraraq kod bazasını daha modular şəkildə saxlanılmasını təmin edir. Repository komponenti verilənlərə çıxış və manipulyasiya əməliyyatlarını müəyyən edən interfeys kimi xidmət edir (Şəkil 7). O, sorğu, yaratmaq, yeniləmək və silmək kimi obyektlər üzərindəki manipulyasiya əməliyyatlarını əhatə edir. Digər tərəfdən, service komponenti tətbiqin biznes məntiqini təmsil edir [8]. O, repository komponentindən istifadə edir və tətbiqin istifadə vəziyyətlərinə uyğun olaraq məlumatların axtarışını, çevrilməsini və manipulyasiyasını təşkil edir.



Şəkil 7. Repository - service dizayn nümunəsinin təsviri.

Yuxarıda qeyd olunan dizayn nümunəsinin tətbiqdə təsviri (Şəkil 8, Şəkil 9, Şəkil 10) aşağıdakı kimidir.

Nümunə istifadəçinin yaradılması servisinə aiddir.

Controller


```

@user_router.post("/", status_code=201)
@Inject
async def create_user(
    obj_in: UserCreate,
    db_session = Depends(get_db),
    user_service = Depends(Provide[UserContainer.user_service])
) -> Response[UserOut]:
    result = await user_service.create_user(db_session, obj_in)
    return {
        "msg": "User created successfully",
        "result": result,
        "is_success": True
    }

```

Şəkil 8. Kontroller səviyyəsində məlumat qəbulu və servis səviyyəsinə ötürülməsi.

Service

```

class UserService(BaseSQLAlchemyServiceWithLogging[UserRepo]):
    async def create_user(self, db_session, user_in: UserCreate) -> UserOut:
        # create user
        db_obj = await self.repo.create(db_session, user_in.dict(exclude={"permissions"}))
        user = await self.get_user(db_session, id=db_obj.id)

        await db_session.commit()
        return UserOut(**user.__dict__)

```

Şəkil 9. İstifadəçinin yaradılması üçün biznes məntiqi.

Repository

```

class UserRepo(BaseSQLAlchemyRepo):
    model = User

    async def create(self, db_session, obj_in):
        """
            Create user
        """
        try:
            return await super().create(db_session, obj_in)
        except IntegrityError as e:
            raise RepoException(e.orig.__cause__.detail, e)

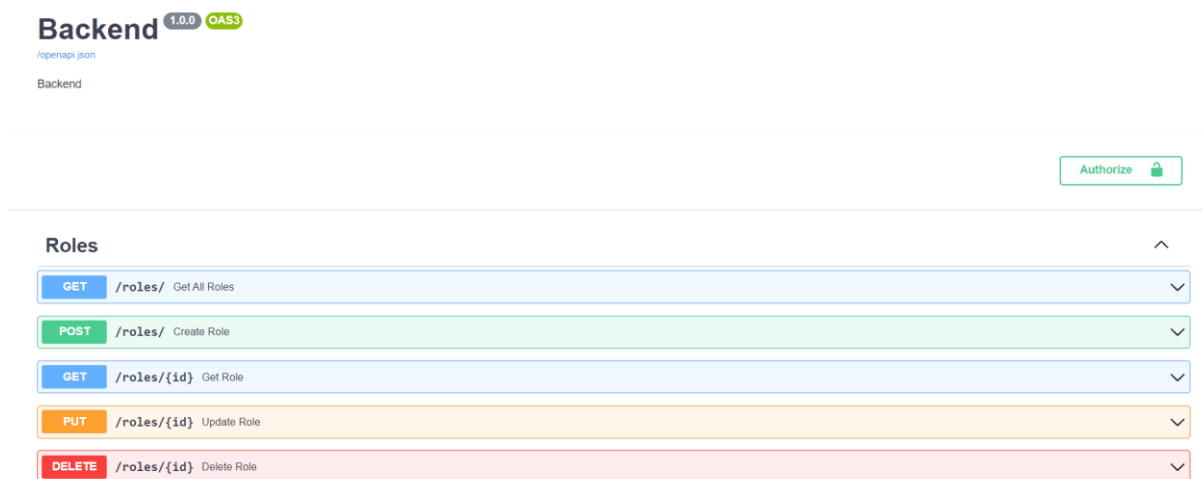
```

Şəkil 10. Baza ilə əlaqə saxlamaq üçün repository səviyyəsi.

Bu nümunədə controller daxilində olan klient tərəfin istifadə etdiyi api interfeysidir. Onun daxilində biznes məntiqi icra edən servis metodunu çağırırıq və həmən servis metodu daxilində verilənlər bazası ilə əlavə saxlayan repository metodunu çağırır. Bu proses boyunca heç bir xəta alınmazsa, istifadəçi uğurla yaranmış olur. Hər hansı bir xəta olarsa, onu klient tərəfin anlaya biləcəyi şəkildə emal edib json formatında ötürülür.

3.3. Swagger

Swagger RESTful API-lərin qurulması, sənədləşdirilməsi və istehlakı üçün açıq mənbə çərçivəsidir (framework). O, tərtibatçılara API son nöqtələrini (endpoint), sorğu/cavab nümunələrini və autentifikasiya mexanizmlərini müəyyən etməyə və vizuallaşdırmağa imkan verən bir sıra alətlər və xüsusiyyətlər təqdim edir. Swagger API tərifinə əsaslanan interaktiv sənədlər, müştəri SDK-ları və server kodu yaratmaqla standartlaşdırma və API inteqrasiyasını asanlaşdırır. FastAPI swagger ilə inteqrasiya olduğundan bizə avtomatik şəkildə api sənədlərini yaratmağa imkan verir. Aşağıda bizim api sənədindən bir hissəni nümunə (Şəkil 11) şəkində görə bilərsiniz.



Şəkil 11. Swagger ilə API dokumentasiyası.

Yuxarıdakı nümunədə rolların yaradılmasının crud əməliyyatları təsvir olunmuşdur.

3.4. Dependency Injection

Dependency injection (asılılıq inyeksiyası) obyektin yaradılmasına və asılılığı həmin obyektləri istifadə edən koddan ayırmağa imkan verən proqram dizayn nümunəsidir. Bu model modulluq, sınaqdan keçirilə bilən və təkrar istifadə oluna bilən komponentlər yaratmaq məqsədi daşıyır. Python Dependency Injector, Python proqramlarında geniş istifadə olunan asılılıq inyeksiya çərçivəsidir. Bu, asılılıq inyeksiya nümunələrinin həyata keçirilməsini asanlaşdıran alətlər və dekoratorlar dəsti təqdim edir. Python Dependency Injector ilə tərtibatçılar öz asılılıqlarını müəyyən edə, həyat dövrlərini idarə edə və avtomatik həll edə bilərlər. Tətbiqin hər bir modulunun daxilində konteyner adlanan python faylları mövcuddur və onlar həmən modul

daxilində olan asılılıqları və onların həyat dövrlərini idarə etmək üçün nəzərdə tutulmuşdur (Şəkil 12). Yuxarıda qeyd olunan dekoratorlar vasitəsi (Şəkil 13) ilə biz həmin obyektləri ehtiyac olan yerdə əldə edə bilərik [8].

```
class ReviewContainer(containers.DeclarativeContainer):
    """
    Review dependencies
    """
    wiring_config = containers.WiringConfiguration(packages=["app.apps"])

    review_repo = providers.Singleton(ReviewRepo)
    review_service = providers.Singleton(ReviewService, repo=review_repo, logger=CoreContainer.app_logger)
```

Şəkil 12. Asılılıqların mərkəzi konteynerdə qeydiyyatdan keçirilməsi.

```
@review_router.delete("/{id}")
@Inject
async def delete_review(
    id: int,
    db_session = Depends(get_db),
    review_service = Depends(Provide[ReviewContainer.review_service]),
    user = Depends(get_current_user)
) -> Response:
    await review_service.delete_review(db_session, id=id)
    return {
        "msg": "Review deleted successfully",
        "result": None,
        "is_success": True
    }
```

Şəkil 13. Asılılıqların istifadə olunması.

Bu nümunə otaqlara verilən rəylər modulundandır. Həmən modul daxilində bizə rəylərlər bağlı servis və verilənlərə müraciət etmək üçün repository lazımdır. Bundan əlavə hər hansı xəta baş verərsə həmən xətalrı loq mesajları şəklində saxlamaq üçün də logger-dən istifadə olunmuşdur. Bu asılılıqların hər biri burada qeydiyyatdan keçirilir və onları istifadə etmək istədikdə inject adlanan dekoratordan istifadə olunur.

3.5. Json veb token

JWT (JSON Web Token) JSON formatında tərəflər arasında məlumatı təhlükəsiz ötürmək üçün kompakt və müstəqil formatdır. Üç hissədən ibarətdir: başlıq, əsas məlumat və imza. JWT-lər tez-tez veb tətbiqlərində autentifikasiya və avtorizasiya məqsədləri üçün istifadə olunur. JWT-lər server tərəfində seans idarəçiliyini tələb etmədən istifadəçiləri etibarlı şəkildə autentifikasiya etmək və avtorizasiya etmək üçün istifadə olunur. O, istifadəçi ID, rol və son istifadə tarixi kimi məlumatları ehtiva edən sorguları daşıya bilər və serverlərə bu məlumatlara etibar etməyə imkan verir. JWT-nin təhlükəsizliyi tətbiqin həyata keçirilməsindən və mühafizə tədbirlərindən asılıdır.

Düzgün tətbiq edildikdə, JWT autentifikasiya və avtorizasiya üçün təhlükəsiz mexanizm təmin edə bilər. Bununla belə, güclü şifrələmə alqoritmlərindən istifadə etmək, istifadə olunan şəxsi açarı qorumaq və ehtiyat tədbirləri görmək vacibdir. Token-in yaradılması alqoritmi aşağıda təsvir olunmuşdur (Şəkil 14) və token daxilində məlumat olaraq istifadəçinin id, email və rolu saxlanılmışdır.

```
def create_access_token(data: dict, expires_delta: timedelta | None = None):
    """
    jwt token generation
    """
    to_encode = data.copy()

    if expires_delta:
        expire = datetime.utcnow() + expires_delta
    else:
        expire = datetime.utcnow() + timedelta(minutes=app_settings.ACCESS_TOKEN_EXPIRE_MINUTES)

    to_encode.update({"exp": expire})
    encoded_jwt = jwt.encode(to_encode, app_settings.SECRET_KEY, algorithm=app_settings.ALGORITHM)
    return encoded_jwt
```

Şəkil 14. JWT tokenin yaradılması.

JWT-nin tətbiqindən asılı olaraq zəiflikləri vardır. JWT ilə əlaqəli ümumi risklərə token sızması, təhlükəsiz saxlama və icra xətalrı daxildir. Təhlükəsizliyi yaxşılaşdırmaq üçün HTTPS-dən istifadə, tokenin bitmə vaxtını məhdudlaşdırmaq və tokenlərin doğruluğunu yoxlamaq vacibdir.

3.6. ORM

Tətbiq də ORM olaraq SQLAlchemy istifadə olunmuşdur. SQLAlchemy, verilənlər bazası əməliyyatlarını səmərəli və effektiv şəkildə idarə etmək üçün FastAPI kimi çərçivələrlə (framework) birlikdə istifadə olunur.

FastAPI daxilində SQLAlchemy-dən istifadə bir sıra üstünlüklər vardır:

1. SQLAlchemy tərtibatçılara verilənlər bazası modellərini Python sinifləri kimi müəyyən etməyə imkan verir, burada hər bir sinif verilənlər bazasında cədvəli təmsil edir. Bu abstraksiya obyekt yönümlü proqramlaşdırma paradigması ilə əlaqəli verilənlər bazası modeli arasında problemsiz keçidə imkan verir.
2. SQLAlchemy PostgreSQL, MySQL, SQLite və başqaları daxil olmaqla bir çox verilənlər bazası mühərriklərini dəstəkləyir. O, müxtəlif verilənlər bazaları ilə qarşılıqlı əlaqə yaratmaq üçün API təmin edir.
3. SQLAlchemy, tərtibatçılara Python metodları və operatorlarından istifadə edərək mürəkkəb verilənlər bazası sorğuları qurmağa imkan verən yüksək

səviyyəli API (sorgular üçün) təmin edir. Bu yanaşma sorguların qurulması prosesini sadələşdirir və SQL ilə bağlı olan təhlükəsizlik risklərini azaldır.

4. SQLAlchemy, obyekt istinadlarından istifadə edərək müxtəlif verilənlər bazası cədvəlləri arasında əlaqələrin müəyyən edilməsinə imkan verən güclü əlaqələr sistemi təklif edir. O, qoşulma, istəkli yükləmə və kaskad əməliyyatları kimi əlaqəli məlumatların sorgulanması və manipulyasiyası üçün rahat üsullar təqdim edir.
5. SQLAlchemy Alembic adlı verilənlər bazası miqrasiya alətini inteqrasiya edir ki, bu da tərtibatçılara verilənlər bazası sxemindəki dəyişiklikləri və versiyanı idarə etməyə imkan verir. O, verilənlər bazası miqrasiyasını yaratmaq və tətbiq etmək üçün sadə bir yol təqdim edir, zamanla verilənlər bazası sxemini təkmilləşdirməyi asanlaşdırır.

FastAPI daxilində SQLAlchemy istifadə edərək, tərtibatçılar proqramın biznes məntiqinə diqqət yetirə və verilənlər bazası əməliyyatlarını ORM çərçivəsinə buraxa bilərlər. SQLAlchemy, Python obyektləri və verilənlər bazası arasında tərcüməni idarə edir.

SQLAlchemy üstünlüyünü görmək üçün aşağıdakı nümunəyə (Şəkil 15) baxmaq olar. Nümunə ana səhifədə otaqların məlumatlarını gətirmək üçün istifadə olunur. Verilmiş sorğu üçün əsas tələblər: bütün otaqları ilk şəkilləri ilə bərabər seçməkdir. Mövcud vəziyyətdə bir otaq üçün bir neçə şəkil mövcud ola bilər, bu nümunədə isə yalnız ilk şəkili gətirmək lazımdır [4].

```

async def list(self, session):
    subq = (
        select(
            func.min(Photo.id).label('photo_id'),
            Photo.room_id
        )
        .group_by(Photo.room_id)
    ).subquery()

    stmt = select(self.model)\
        .options(contains_eager(Room.photos))\
        .join(subq, and_(self.model.id == subq.c.room_id))\
        .join(Photo, and_(subq.c.photo_id == Photo.id))
    result = await session.execute(stmt)
    return result.unique().scalars().all()

```

Şəkil 15. Ssqlalchemy ilə sql sorgusunun hazırlanması və icrası.

3.7. Database (verilənlər bazası)

Postgresql - açıq mənbəli əlaqəli verilənlər bazası idarəetmə sistemidir. İlk dəfə 1986-cı ildə Kaliforniya Universitetində hazırlanmışdır və hazırda bütün dünyada geniş istifadə olunan verilənlər bazası sistemidir. Əlaqəli verilənlər bazası idarəetmə sistemləri verilənlərin cədvəllərdə təşkil edildiyi və cədvəllər arasında əlaqələrin qurulduğu bir quruluşa malikdir. Bu strukturundan istifadə edərək PostgreSQL məlumatları saxlayır, idarə edir və verilənlər bazası proqramlarının işləməsini təmin edir. Həm kiçik miqyaslı layihələrdə, həm də böyük müəssisə proqramlarında istifadə olunur. Həmçinin, PostgreSQL böyük bir cəmiyyətə malikdir və istifadəçilər arasında dəstək və resurs mübadiləsini təmin edir.

PostgreSQL-in üstünlükləri - PostgreSQL-in bir çox üstünlükləri var. PostgreSQL-in əsas üstünlükləri bunlardır:

1) Açıq Mənbə və Pulsuz olması: PostgreSQL açıq mənbəli layihədir və tamamilə pulsuzdur. Bu o deməkdir ki, hər kəs PostgreSQL-i yükləyə və istifadə edə bilər. Həmçinin, mənbə kodu əlçatan olduğundan, istifadəçilər PostgreSQL-i ehtiyaclarına uyğun olaraq fərdiləşdirə və inkişaf etdirə bilərlər.

2) Geniş funksiyalar Dəsti: PostgreSQL bir çox inkişaf etmiş funksiyaları dəstəkləyir. JSON məlumat növləri, tam mətn axtarışı, geoməlumatlar, çoxsaylı versiyalar, daxili funksiyalar, əməliyyatların ardıcılığı və bir çox digər funksiyalar PostgreSQL-i güclü verilənlər bazası idarəetmə sisteminə çevirir.

3) Genişləndirmə və Plugin Dəstəyi: PostgreSQL istifadəçilərə xüsusiyyətləri və funksionallığı genişləndirmək üçün genişləndirmələr və plaginlərdən istifadə etməyə imkan verir. Bu, istifadəçilərə PostgreSQL-i ehtiyaclarına uyğun olaraq uyğunlaşdırmağa və fərdiləşdirməyə imkan verir.

4) Yüksək Performans və Ölçüləndirilmə qabiliyyəti: PostgreSQL sürətli və yüksək performanslı verilənlər bazası idarəetmə sistemidir. Təkmilləşdirilmiş sorğu optimallaşdırılması, sorğuların paralel işlənməsi, indeksləşdirmə və məlumatların sıxılması kimi xüsusiyyətlərlə verilənlər bazası performansını artırır. Bundan əlavə, PostgreSQL yüksək trafikli tətbiqlərdə miqyaslılığı təmin edir.

5) Data Integrity: PostgreSQL verilənlərin bütövlüyünü qorumaq üçün müxtəlif məhdudiyyətlər və mexanizmlər təklif edir. O, ilkin açarlar, xarici açarlar, unikal məhdudiyyətlər kimi xüsusiyyətlərlə məlumatların bütövlüyünü təmin edir.

6) Təhlükəsizlik: PostgreSQL təhlükəsizlik sahəsində möhkəm təmələ malikdir. O, verilənlər bazasına giriş, istifadəçi icazəsi və şifrələmə kimi təhlükəsizlik tədbirləri ilə məlumatların təhlükəsizliyini təmin edir.

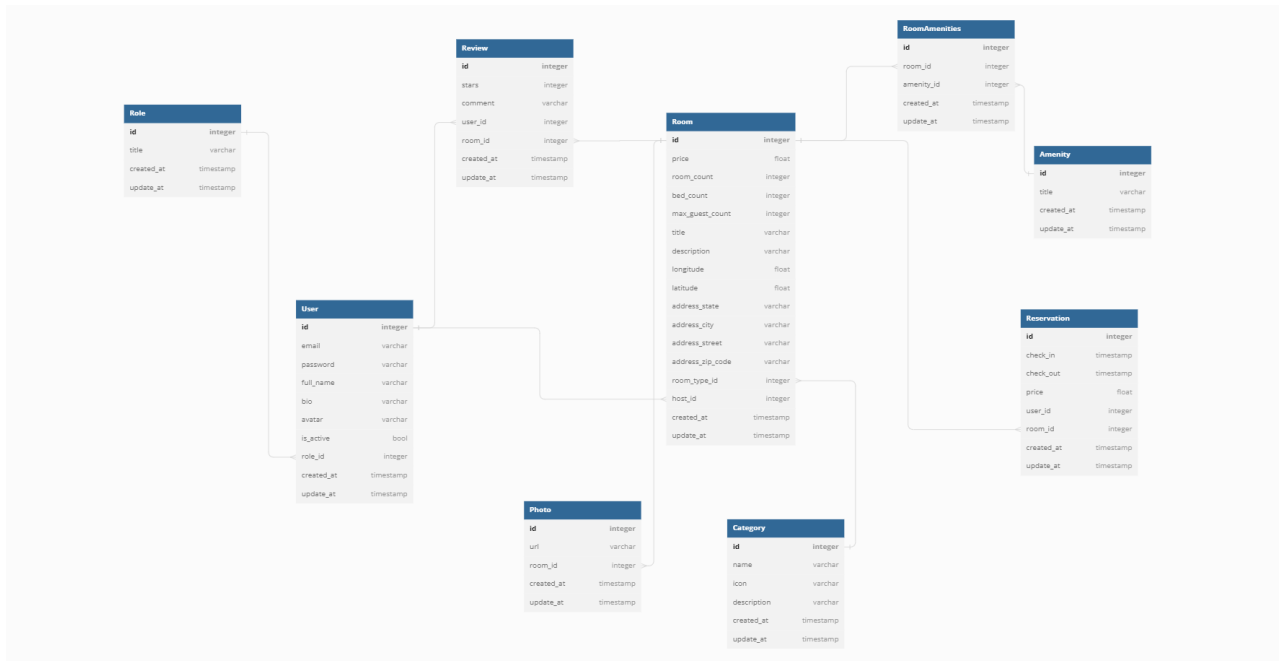
7) Aktiv Cəmiyyət və Dəstək: PostgreSQL böyük və aktiv istifadəçi cəmiyyətinə malikdir. Bu cəmiyyət istifadəçilərə dəstək verir, suallara cavab verir, problemləri həll edir və resursları paylaşır. Bundan əlavə, PostgreSQL-də çoxlu sənədlər və resurslar var.

Bu üstünlüklər PostgreSQL-i bir çox müxtəlif layihələrdə üstünlük verilən verilənlər bazası idarəetmə sisteminə çevirir. Tətbiqimizdə postgresql-dən istifadə olunmuşdur. Aşağıdakı cədvəllər (Şəkil 16) mövcuddur [5].

```
test_db=# \dt
          List of relations
Schema |          Name          | Type  | Owner
-----+-----+-----+-----
public | alembic_version       | table | test_user
public | amenty                 | table | test_user
public | category               | table | test_user
public | photo                  | table | test_user
public | reservation            | table | test_user
public | review                 | table | test_user
public | role                   | table | test_user
public | room                   | table | test_user
public | roomamenities         | table | test_user
public | roomdetail             | table | test_user
public | user                   | table | test_user
(11 rows)
```

Şəkil 16. Cədvəllərin adları (postgresql konteyner daxilində).

Cədvəllər arasındakı əlaqə (Şəkil 17):



Şəkil 17. Verilənlər bazasında cədvəllər arası əlaqə.

Tranzaksiya - PostgreSQL-də "tranzaksiya" bir məntiqi əməliyyat kimi işləmək üçün çoxlu verilənlər bazası əməliyyatlarını birləşdirməyə imkan verən bir konsepsiyadır. Tranzaksiya tamamilə yerinə yetirilən və ya ümumiyyətlə yerinə yetirilməyən bir sıra verilənlər bazası əməliyyatlarından ibarətdir. Tranzaksiyalar verilənlər bazasında ardıcillıq və bütövlüyü təmin etmək üçün mühüm mexanizmdir.

Tranzaksiyalar ümumiyyətlə ACID prinsiplərinə əsaslanır. Bu prinsiplər deməkdir:

1) **Atomluq**: Əməliyyat vahid məntiqi əməliyyat hesab olunur və ya tamamilə yerinə yetirilir, ya da heç yerinə yetirilmir. Bütün əməliyyatlar uğurla başa çatdıqda, əməliyyatın nəticəsi qalıcıdır. Hər hansı bir əməliyyat uğursuz olarsa, əməliyyat geri qaytarılır və bütün dəyişikliklər geri qaytarılır.

2) **Ardıcillıq**: Əməliyyat verilənlər bazasının müəyyən ardıcillığını saxlamalıdır. Verilənlər bazasının başlanğıc və son vəziyyətləri arasında ardıcıl vəziyyətdə olmasını təmin etmək üçün tədbirlər görülür.

3) **İzolyasiya**: Bir əməliyyatın digər əməliyyatlarla qarşılıqlı əlaqəsi təcrid olunmalıdır. Əməliyyata digər əməliyyatların əməliyyatları təsir etməməlidir.

İzolyasiya səviyyələri əməliyyatın digər əməliyyatlardan necə təcrid olunduğunu müəyyən edir.

4) Davamlılıq: Əməliyyat başa çatdıqdan sonra baş verən dəyişikliklər qalıcı olmalıdır. Bu o deməkdir ki, əməliyyatın nəticəsi həmişə verilənlər bazasında saxlanılır və gələcək səhvlərə və ya fasilələrə baxmayaraq onu əldə etmək mümkün deyil.

PostgreSQL bu ACID prinsiplərini dəstəkləyən verilənlər bazası idarəetmə sistemidir. O, əməliyyatların başlanması, əməliyyatların birləşdirilməsi, əməliyyatların ləğvi və ya tamamlanması kimi əməliyyatların idarə edilməsi əməliyyatlarını təmin edir. Bu yolla verilənlər bazasında ardıcılıq, bütövlük və etibarlılıq təmin edilir [5].

Prosedur - bir və ya daha çox SQL ifadəsi və ya kod blokundan ibarət parametrləri qəbul edə bilən adlandırılmış verilənlər bazası obyektidir. Prosedurlar verilənlər bazasında təkrarlanan tapşırıqları yerinə yetirmək üçün istifadə olunur və istifadəçilərə verilənlər bazası üçün xüsusi iş məntiqi və ya əməliyyatlarını müəyyən etməyə imkan verir. Prosedurlar verilənlər bazası tərəfində saxlanılan və sorğu dilində (SQL) yazılmış kod parçaları ehtiva edir. Prosedur adla müəyyən edilir və müəyyən bir məqsəd və ya funksiyanı yerinə yetirir. Prosedurlar daha mürəkkəb əməliyyatları yerinə yetirmək, verilənləri manipulyasiya etmək, sorğuları yerinə yetirmək və ya digər verilənlər bazası əməliyyatlarını yerinə yetirmək üçün istifadə edilə bilər. PL/pgSQL (PostgreSQL-in mövcud prosedur dili genişlənməsi) tez-tez PostgreSQL-də prosedurlar yaratmaq üçün istifadə olunur. PL/pgSQL prosedurları müəyyən etmək, dəyişənlərdən istifadə etmək, döngələr və şərtlər yaratmaq kimi xüsusiyyətləri dəstəkləyir. Parametrlər proseduru təyin edərkən prosedur daxilində müəyyən edilə və istifadə edilə bilər. Prosedur tərəfindən yerinə yetiriləcək əməliyyatlar BEGIN və END blokları arasında yerləşir.

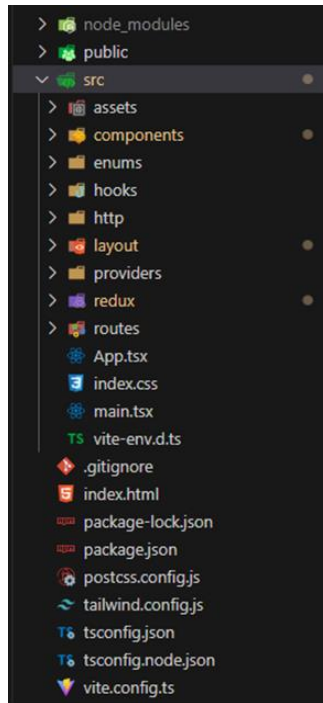
Prosedurlar verilənlər bazasında saxlanılır və lazım olduqda çağırıla bilər. Prosedurlar verilənlər bazası idarə edilməsində və tətbiqlərin inkişafında səmərəlilik, təkrar istifadə və əməliyyatların idarə edilməsi baxımından böyük üstünlük təmin edir.

3.8. Frontend (istifadəçi tərəfi)

Tətbiqdə istifadə edilən texnologiyalar - ağıllı şəhər ekosistemi çərçivəsində turistlərə dəstək veb layihəsinin hazırlanması zamanı istifadə etdiyimiz texnologiyalar

bunlardır: JavaScript proqramlaşdırma dili (kitabxana olaraq React JS), verilən dəyərlərin tiplərini təyin etmək, obyekt yönümlü proqramlaşdırma imkanlarından istifadə etmək üçün TypeScript proqramlaşdırma dili. Həmçinin dəyərlərin (state) idarə edilməsi üçün React Redux Toolkit kitabxanası (ingilis dilində - “state management”), dinamik məlumatları əldə etmək, servise sorğu göndərə bilməyimiz üçün Axios kitabxanasından istifadə edilmişdir. Bundan əlavə, istifadəçinin tarix məlumatlarını əlavə edə bilməsi və komponentlərin bu tarixi proqramın başa düşə bilməsi üçün Date-fns kitabxanası, istifadəçinin layihə daxilində istifadə edilən xəritədən istifadə edə bilməsi üçün Leaflet kitabxanası, layihə daxilində məlumatların detallarına baxan zaman, həmin məlumatın url ünvanını təyin etmək üçün “query-string” kitabxanası işlədilmişdir. İstifadəçinin tarix aralıqlarını seçim etmək üçün React Date Range kitabxanası, istifadəçinin daxil edəcəyi, məlumatları dolduracağı form üçün React Hook Form kitabxanası, bildiriş məlumatlarının istifadəçiyə göstərilməsi üçün React Hot Toast, seçim edə bilmək üçün istifadə edilən komponentin yaradılması məqsədilə React Select kitabxanası kömək olmuşdur. Səhifələr arası keçidi təmin etmək üçün isə React Router kitabxanası, stilləndirmə proseslərinin asanlaşdırılması məqsədilə CSS dilinin kitabxanası olan Tailwind CSS istifadə edilir. Bütün bu kitabxanaların yüklənməsi, bir qovluq altında birləşməsi və paket halına salınması üçün isə Vite JS adında kitabxanadan istifadə edilir. React JS kitabxanasının istifadə edilməsinə ən başlıca səbəb, orta və böyük həcmli layihələrdə, istifadə edilən element, komponent sayları çox olduğundan onları vahid şəkildə qruplaşdırmaq, təkrarlanmaların sayını azaltmaq, həmçinin layihənin brauzerdə yüklənməsi zamanı resurslara qənat etmək, lazım olduğu halda elementin yüklənməsini təmin etməkdir. Bu məqsədlə hazırlanan layihənin qovluq strukturunu maksimal şəkildə sadələşdirmək, başa düşülən halda qurmaq önəm daşıyır [7].

Layihənin fayl strukturu – layihənin fayl strukturu (və ya arxitekturası) Şəkil 18-də göstərildiyi kimidir.



Şəkil 18. İstifadəçi interfeysinin qovluq strukturu.

“Public” qovluğu daxilində layihənin istənilən hissəsindən birbaşa müraciət edə biləcəyimiz fayllar yerləşdirilir. “Src” qovluğu isə digər əsas faylların yerləşdiyi hissədir. Burada “assets” qovluğunda şəkillər, video və musiqi, sənəd formatında fayllar saxlanıla bilər. “Components” faylında isə layihə hazırlanarkən, istifadəçinin brauzer üzərində gördüyü elementlərin saxlanılır. Nümunə olaraq, veb saytın başlığı (header), kontentləri, istifadəçinin klik etdiyi hər hansı düymə, yazı sahələri, şəkillər, köməkçi pəncərələr və s. “Enums” qovluğu daxilində sabit, statik məlumatlar vardır. Həmçinin ən önəmli nüans, bizim yaratdığımız “api.tsx” adlı fayldır. Bu faylın xidmət sahəsi ondan ibarətdir ki, biz servise göndərdiyimiz sorğular üçün, lazım olan URL adresini mərkəzi olaraq buradan idarə edə bilərik. URL adres bizə layihənin müxtəlif fayllarında lazım ola bilər. Mərkəzləşdirilmiş bu struktur, gələcəkdə yarana biləcək hər hansı dəyişikliyə bizim bütün fayllar üzərində ediləcək proseslərdən azad edir.

“Hooks” qovluğu haqqında məlumat verməzdən öncə “hook” anlayışı haqqında məlumat vermək doğru olar. React JS kitabxanası ilə proqramçıların tanıdığı “hook”-lar o işə xidmət edir ki, təkrarlanan prosesləri vahid formaya gətirir. Biz bu təkrarlanmaları, hansı ki kod sətirləri yüzlərlə ola bilər, vahid funksiya şəklində hazırlayıb, istifadəsi lazım olan hissələrdə yalnız 1 sətirlə yekunlaşdırıla bilərik [5, 6].

Bizim bu layihəmizdə isə ölkə adlarını gətirmək, köməkçi ekranların açılıb-bağlanması üçün istifadə edirik.

“HTTP” qovluğu layihəmizin ən önəmli proseslərindən birini yerinə yetirir. Layihəmiz veb layihə olduğu üçün, həmçinin yalnız turistlərə deyil ümumi cəmiyyətə açıq olduğu üçün hər kəs daxil olmaq imkanına sahibdir. Bu səbəbdən istifadəçilərin təhlükəsizliyini, onlara məxsus özəlliklərin aktivləşdirilməsi və digər səbəblərdən avtorizasiya prosesinin yerinə yetirilməsi böyük önəm daşıyır. Veb saytımızı daxil olan istifadəçi həm qeydiyyatdan keçib özünə yeni bir şəxsi hesab yarada bilər, həm də öncədən yaratmış olduğu hesabına yenidən daxil ola bilər. Əgər istifadəçi bu prosesləri yerinə yetirməyibsə, onda müəyyən özəlliklər vardır ki, nümunə olaraq ev rezervasiyası, bu əməliyyatı etmək hüququ əlindən alınır. HTTP – HyperText Transfer Protocol – brauzerlə server arasında əlaqə qurmaq üçün istifadə edilən protokoldur. HTTP tərəfindən göndərilən sorğular request, qəbul edilən dəyərlər response adlanır. HTTP sorğuları URL (Uniform Resource Locator) adresini özündə saxlayır və HTTP metodlarından istifadə edərək response istəyir. HTTP metodları – GET, POST, PUT, DELETE-dir. HTTP sorğusu tərkibində Header-da ola bilər. Header istifadəçidən şəxsiyyətini doğrulamaq üçün şəxsiyyət məlumatlarını və ya requestdə hansı formada data göndəriləcəyini təyin edir. Response olaraq – HTML, JSON, XML kimi datalar gələ bilər.

Axios, JavaScriptdə istifadə edilən bir kitabxana olub HTTP sorğularını göndərmək və qəbul etmək üçün istifadə edilir. Axios eyni anda birdən çox sorğunu göndərə və qəbul edə bilər. Axios daxilində interceptor adlanan xüsusiyyət mövcuddur. İnterceptorlar vasitəsilə http sorğularından əvvəl və sonra əməliyyatlar yerinə yetirmək mümkündür. Məsələn HTTP sorğusu göndərilməzdən əvvəl, Authorization Header göndərmək kimi və ya sorğu bitdikdən sonra başqa əməliyyatlar yerinə yetirmək kimi [6].

- HTTP sorğusundan əvvəl əməliyyatları yerinə yetirmək üçün `instance.interceptors.request.use()`
- Sonra üçün `instance.interceptors.response.use()` – istifadə edilir. Aşağıda ətraflı görəcəksiniz.

`axios.create()` – axios kitabxanasının tərkibində olan bir metoddur və axios instance yaratmaq üçün istifadə edilir. Bir proqramda eyni apidən istifadə edərək sorğu göndərmək lazım olduqda hər dəfə hər sorğu üçün təkrar təkrar eyni strukturu yazmaq əvəzinə əvvəldən `axios.create()` dən istifadə edərək instance yaradaraq daha asan istifadə etmək mümkündür.

Qısaca `axios.create()` bizim parametrlərimizə uyğun olaraq axios sorğusu hazırlayır. Məsələn aşağıdakı parametrlərə malik olan 10 sorğumuz olacaqsa, əvvəldən standart struktur yaradaraq istifadə etmək olar. Artıq sonradan servis adı dəyişdikdə bir yerdən dəyişərik, bütünüənə təsir edə bilərik (Şəkil 19).

```
import axios from 'axios';

const instance = axios.create({
  baseURL: 'https://example.com/api',
  headers: {
    Authorization: `Bearer ${jwtToken}`,
  },
});

instance.get('/users')
  .then(response => console.log(response.data));
```

Şəkil 19. Axios create metodu vasitəsilə instance yaradılması.

İstifadə etmək istədikdə sadəcə `instance.get(/users)` yazmaq kifayətdir. Burada GET http sorğu növüdür. Mötərizə daxilində yazdığımız isə `baseURL` in sonuna yazılacaq url-in davamını göstərir. `https://example.com/api/users` url-in son forması belə olur. Hər sorğu üçün `baseURL` ve `headers`-də Auth istifadə edəcək.

Authorization bir istifadəçinin şəxsiyyətinin doğrulanması prosesidir. Authorization prosesi ilə şəxsiyyət doğrulama edərək istifadəçilərə müəyyən icazələr verilə bilər. Web applikasiyada Authorizasiya prosesi aşağıdakı kimi yerinə yetirilir: Auth prosesi üçün hər sorğu göndərməzdən əvvəl gərək `interceptors.request` ilə Authorization headerində token göndərərək şəxsiyyəti doğrulayaq. HTTP sorğularında istifadə edilən headerdir. Bu istifadəçinin şəxsiyyətini doğrulamaq üçün və veb saytda hər hansı bir əmri yerinə yetirmək üçün ona icazəsi olub olmadığını “access token” göndərərək müəyyən edə bilər.

`Interceptors.request.use()` - HTTP sorğusu işə düşmədən əvvəl `instance.interceptors.request` işə düşür. Config və Error dəyərlərindən ibarətdir. Bu şəkildə sorğu göndərməzdən əvvəl sorğunun Header-in də düzəliş edərək Authorization header əlavə edir və dəyər olaraq local storage da saxladığımız tokeni göndərir (Şəkil 20).

```
instance.interceptors.request.use(
  (config) => {
    const token = localStorage.getItem("access_token");
    if (token) {
      config.headers["Authorization"] = token;
    }
    return config;
  },
  (error) => {
    return Promise.reject(error);
  }
);
```

Şəkil 20. Sorğudan əvvəl işə düşən kod.

İlk parameter olan config - interceptors dəyişənidir. Configin daxilində Axios instances-in bütün xüsusiyyətləri parametrləri saxlanır (obyekt şəklində). Şəkildə `config.headers()` edərək Authorization leave edir ve dəyər verir [4]. Error - interceptors dəyişənidir. Əgər interceptor prosesi zamanı xəta çıxarsa `Promise.reject(error)` qaytaracaq. `Interceptors.request` prosesi olduqdan sonra HTTP sorğusu prosesi işə düşür. Sorğu bitikdən sonra isə `interceptor.response.use()` funksiyası yerinə yetirilir. Funksiyadan error və ya response dəyəri qayıda bilər. `Response.use`-da iki dəyər vardır: response və error (Şəkil 21).

```
(response) => {
  return response;
},
```

Şəkil 21. Sorğu nəticəsində əldə edilən nəticə.

Burada response dəyəri - HTTP sorğusundan geri qayıdan response-dur. Əgər problemsiz response gəlibsə response-u qaytarır. Error - Http sorğusu zamanı problem yaranarsa bizə qayıdan errordur (Şəkil 22).

```

axios.interceptors.response.use(
  response => {
    return response;
  },
  error => {
    console.error(error.config);
    return Promise.reject(error);
  }
);

```

Şəkil 22. Sorğu bitdikdən sonra işə düşən kod.

Burada error.config xəyata səbəb olan sorğunun strukturunu verir. Tərkibində sorğuya aid bütün parametrlər - headerlər, url, data və s. olur. Error.response – bizə errordan gələn cavabı qaytarır. Error.response daxilində error-un statusu, error-un adı və s. ola bilər.originalConfig._retry nədir? - _retry sorğunun təkrar yenidən yerinə yetirildiyini göstərir. Əvvəlcə dəyəri false olur çünki sorğu təkrar yerinə yetirilməyib. Təkrar yerinə yetirmək üçün dəyərini true vermək olar. Try {} daxilində yenidən access token almaq üçün refresh tokenə istifadə edərək sorğu göndərilir. Ardınca return instance(originalConfig) edərək əvvəlki etdiyimiz sorğu təkrarlanır [4].

React Redux Toolkit, React və Redux ilə birlikdə istifadə edilən bir JavaScript kitabxanasıdır. Bu kitabxana, Redux ilə sərfəli və effektiv bir şəkildə vəziyyət (dəyər, state) idarəetməsini və məlumat axınıni təşkil etməyə kömək edir. Aşağıdakı məlumatlar, React Redux Toolkit barəsində əsas məlumatları və istifadə nümunəsini təqdim edir. React Redux Toolkit, Redux-un quruluş və işlətmə məntiqini sadələşdirir və çox fərqli funksiyalar və utiliteler təqdim edir. Bu kitabxana, Redux-ın quruluşu üçün gələnək variantlardan daha sərfəli bir alternativdir və bir çox tez-tez təkrarlanan kod hissələrini avtomatlaşdıraraq effektivliyi artırır. Layihəmizdə Redux üçün qovluq

strukturunu 3 əsas hissəyə bölmüşük: Store, Actions və Reducers. Store daxilində yaratdığımız Slice-ları (hissələri) istifadə etmək üçün kodlar əlavə edilir (Şəkil 23).

```
import { combineReducers, configureStore } from "@reduxjs/toolkit";
import auth from "./reducers/AuthSlice";
import room from "./reducers/RoomSlice";

const rootReducer = combineReducers({
  auth,
  room,
});

export const store = configureStore({
  reducer: rootReducer,
});

export const setupStore = () => {
  return store;
};

export type RootState = ReturnType<typeof rootReducer>;
export type AppStore = ReturnType<typeof setupStore>;
export type AppDispatch = typeof store.dispatch;
```

Şəkil 23. Redux Toolkit strukturu.

Reducers daxilində bizim ilkin dəyərlərin saxlandığı bölmələr, Slice-in (hissənin) adı, dəyərlərin məlumatlarının dəyişdirilməsi üçün funksiyalar və Actions-lar vasitəsilə sorğulardan gələn nəticələrin dəyərlərə mənimsədilməsi var [6]. Actions faylında isə kitabxananın bizə təqdim etdiyi createAsyncThunk metodu vasitəsilə HTTP sorğularını göndərmək əməliyyatlarını yerinə yetiririk (Şəkil 24).

```
export const login = createAsyncThunk(
  "auth/login",
  async (user: any, thunkAPI) => {
    try {
      const response = await axios.post(`${API.MAIN_URL}/auth/login`, user);
      const { data } = response;
      localStorage.setItem("token", data.result.access_token);
      localStorage.setItem("token_type", data.result.token_type);

      return data;
    } catch (e: any) {
      return thunkAPI.rejectWithValue("Error");
    }
  }
);
```

Şəkil 24. Redux Toolkit Action təsviri.

React Router v6, React əsaslı bir yönləndirmə kitabxanasıdır və çox müxtəlif funksionalitetlər təklif edir. "Public" root-lar, hər kəs tərəfindən əlçatan edilən səhifələr üçün nəzərdə tutulub. Bu, məlumatların göstərilməsi, məqalələrin oxunması və kateqoriyalara baxılması kimi amillər üçün istifadə oluna bilər. "Private" köklər isə adətən girişli olan istifadəçilər üçün olan məlumatları, hesab məlumatlarını idarə etmək və gizlilik tələb edən funksiyaları özündə birləşdirir [6].

Dinamik ID yönləndirməsi, React Router v6-nın güclü funksionalitetlərindən biridir. Bu proses, URL-dəki dəyişən ID parametri vasitəsilə hər bir səhifənin fərqli məzmun göstərməsinə imkan verir. Məsələn, bir məhsul səhifəsinin URL-dəki dinamik ID, məhsulun xüsusiyyətlərini və əlavə məlumatlarını göstərmək üçün istifadə oluna bilər. Böyük həcmli tətbiqlərdə dinamik ID yönləndirməsi, saytların daha yalın və tənzimləmə bilən olmasına imkan verir.

Veb saytımızda səhifələr arası keçid əməliyyatlarını yerinə yetirmək üçün isə React Router kitabxanasından istifadə olunubdur. Bu kitabxananın “Private Router”, “Public Router” və dinamik ID xüsusiyyətləri istifadə edilib. Private və Public router anlayışları istifadə edilməsində məqsəd, istifadəçinin kimliyinin doğrulanıb-döğrulanmadığını təsdiq etmək üçün istifadə edilib. Belə ki, layihəmizdə istifadəçi öz evini yerləşdirmək istəyirsə və ya başqa bir evi kirə almaq istəyirsə bu zaman istifadəçinin kimliyinin doğrulanmasını, yəni şəxsi hesabına daxil olması tələb olunur. Bunun üçün istifadəçiyə köməkçi qeydiyyat və ya daxil ol pəncərəsi açılır. İstifadəçinin etmək istədiyi əməliyyatların qarşısını sırf Private və Public route ilə alırlar [7]. Öncəki bölmələrdə qeyd etdiyimiz kimi, istifadəçi öz hesabına daxil olduqdan sonar “access token” əldə edir. Biz bu tokeni brauzerin müvəqqədi yaddaşında saxlayırıq. Daha sonar router fayllarımızda yaddaşda saxlanılan dəyərlər müqayisə olunur. Əgər həmin dəyərlər istifadəçinin doğruluğunu təyin edirsə lazım olan səhifələrə giriş imkanı aktivləşir (Şəkil 25).

```
import { useEffect } from "react";
import { Outlet, useNavigate, useLocation } from "react-router-dom";
import { useAppSelector } from "../hooks/redux";

const ProtectedRoutes = (props: any) => {
  const { isAuthenticated } = useAppSelector((state) => state.auth);
  let navigate = useNavigate();
  let location = useLocation();
  useEffect(() => {
    if (isAuthenticated === false) {
      navigate("/login", { state: location.pathname });
    }
  }, [isAuthenticated]);
  return isAuthenticated ? <Outlet /> : null;
};

export default ProtectedRoutes;
```

Şəkil 25. Protected Routes strukturu.

İstifadəçi öz şəxsiyyətini təsdiq etdikdən sonra qarşısına çıxacaq evlər haqqında ümumi məlumatlar əldə edəcəkdir. Müxtəlif senarilərə uyğun olaraq, müxtəlif evlər vardır. İstifadəçinin əməliyyatına uyğun olaraq hər hansı bir ev haqqında geniş məlumat əldə etmək istədikdə, bu vaxt bizə React Router kitabxanasının “Dinamik Routing” anlayışı kömək olur. Servis üzərindən bizə hər bir ev haqqında məlumat gəldikdə, unikal olaraq evlərin ID məlumatlarını da əldə etmiş oluruq. Biz kitabxananın təqdim etdiyi metoda həmin ID məlumatı göndərərək (Şəkil 26), həmçinin servise GET request göndərərək, ev haqqında detallı məlumatları bizə geri dönmüş olur.

```
const App = () => {
  const dispatch = useAppDispatch();

  useEffect(() => {
    dispatch(usersMe());
  }, []);

  return (
    <Layout>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/listings/:id" element={<Listing />} />
      </Routes>
    </Layout>
  );
};
```

Şəkil 26. Router prosesinin App faylı üzərində tətbiqi.

NƏTİCƏ

Ağıllı şəhərlər texnoloji inkişaf, artan əhali və urbanizasiya meylləri səbəbindən sürətlə inkişaf edir. Bu şəhərlər səmərəlilik, davamlılıq, təhlükəsizlik, həyat keyfiyyəti və vətəndaşların iştirakı kimi bir çox üstünlüklər təklif edir. Bununla belə, bu inkişaf, təhlükəsizlik, məxfilik, məlumatların idarə edilməsi, uyğunluq, maliyyələşdirmə və investisiya problemləri kimi potensial maneələr və problemlər də təqdim edə bilər. Gələcəkdə ağıllı şəhərlərin inkişafı və böyüməsi davam edəcək. Yeni texnologiyalar, məlumatların təhlili, süni intellekt və əşyaların interneti kimi sahələrdə irəliləyişlər ağıllı şəhərlərin daha da inkişafına səbəb olacaq və onların təsirini artıracaq. Lakin bu inkişafın idarə edilməsində və həyata keçirilməsində çətinlikləri nəzərə alaraq, bu prosesi təşkil etmək və idarə etmək lazımdır. Nəticə etibarilə, ağıllı şəhərlər gələcəkdə daha da vacib olacaq və bir çox üstünlüklər təqdim etsə də, bu inkişafın idarə edilməsi təhlükəsizlik, məlumatların idarə edilməsi, uyğunluq və maliyyələşdirmə nəzərə alınmaqla həll edilməlidir.

Əsas məqamların ümumiləşdirilməsi. Ağıllı şəhərlər texnoloji inkişaf sayəsində səmərəlilik, davamlılıq, təhlükəsizlik, həyat keyfiyyəti və vətəndaşların iştirakı kimi bir çox fayda təmin edir. Bununla belə, bu inkişaf, təhlükəsizlik, məxfilik, məlumatların idarə edilməsi, uyğunluq, maliyyələşdirmə və investisiya problemləri kimi potensial maneələr və problemlər də təqdim edə bilər. Gələcəkdə ağıllı şəhərlərin inkişafı və böyüməsi davam edəcək. Lakin bu prosesin idarə edilməsində və həyata keçirilməsində çətinlikləri nəzərə almaqla həll etmək lazımdır [2]. Nəticə etibarilə, ağıllı şəhərlər gələcəkdə daha da vacib olacaq və bir çox üstünlüklər təqdim etsə də, bu inkişafın idarə edilməsi təhlükəsizlik, məlumatların idarə edilməsi, uyğunluq və maliyyələşdirmə nəzərə alınmaqla həll edilməlidir.

Yaşayış olmayan ərazilərdə smart şəhər konspeti. Dünyanın müxtəlif yerlərində ağıllı şəhərlərin inkişafı istiqamətində davam edən tendensiya sakinlərin həyat keyfiyyətinin yaxşılaşdırılmasına yönəlmiş bir sıra yeniliklər gətirib. Bu texnoloji inqilab tədricən cəmiyyətin bütün sahələrinə nüfuz edir və ağıllı şəhər komponentlərinin inteqrasiyasının getdikcə daha çox göründüyü bir sahə ölkənin yaşayış olmayan əraziləridir. Ağıllı şəhərlərin əsas məqsədlərindən biri enerji, su və

tullantıların idarə olunması kimi resurslardan səmərəli istifadə etməkdir. Lakin bu üstünlüklər təkcə şəhər yerləri ilə məhdudlaşmır. Ağıllı şəhər həlləri daimi insan nəzarətinin olmaması resursların səmərəli idarə olunmasını çətinləşdirən yaşayış olmayan ərazilərdə eyni dərəcədə effektiv ola bilər. Məsələn, yaşayış olmayan ərazilərdə fermerlərin üzləşdiyi əsas problemlərdən biri məhsullarının lazımi miqdarda su almasını təmin etməkdir. Bununla belə, ağıllı suvarma sistemlərinin inteqrasiyası ilə fermerlər bitkilərinin böyümənin müxtəlif mərhələlərində tələb etdiyi optimal məbləğə qərar verərək, sudan istifadəni effektiv şəkildə idarə edə bilərlər [1]. Bu sistemlər fermerlərin su istehlakı ilə bağlı daha yaxşı qərarlar qəbul etmələrini təmin etmək üçün torpaq rütubəti sensorları, hava proqnozları və məhsulun monitorinqi alətləri kimi müxtəlif mənbələrdən məlumat toplamaq üçün nəzərdə tutulub. Ağıllı şəhər komponentlərinin yaşayış olmayan ərazilərdə effektiv şəkildə inteqrasiya oluna biləcəyi başqa bir sahə nəqliyyatdır. Belə ərazilərdəki böyük məsafələri nəzərə alsaq, nəqliyyat həm fiziki şəxslər, həm də müəssisələr üçün əsl problem ola bilər. Bununla belə, avtonom dronlar kimi ağıllı texnologiyaların inteqrasiyası ilə yük maşınları və ya vertolyotlar kimi bahalı və uzun daşınma variantlarına ehtiyac olmadan yüklərin ucqar ərazilərə çatdırılması artıq mümkündür. Ağıllı enerji həlləri ağıllı şəhər komponentlərinin inteqrasiyasının əhəmiyyətli təsir göstərə biləcəyi başqa bir sahədir. Yaşayış olmayan ərazilərdə enerji, xüsusən də şəbəkəyə hazır və daimi çıxışın olmaması səbəbindən baha başa gəlir. Bununla belə, ağıllı günəş enerjisi həllərinin inteqrasiyası enerji xərclərini kəskin şəkildə azalda və davamlı və etibarlı enerji mənbəyi təmin edə bilər. Yekun olaraq, ölkənin yaşayış olmayan ərazilərində ağıllı şəhər komponentlərinin inteqrasiyası bu bölgələrin idarə olunmasında inqilab etmək potensialına malikdir. IoT sensorları, süni intellekt və blokçeyn kimi müxtəlif ağıllı texnologiyalardan istifadə etməklə indi resursları səmərəli idarə etmək, məhsuldarlığı artırmaq və bu ərazilərdə yaşayan insanların həyat keyfiyyətini yaxşılaşdırmaq mümkündür. Buna görə də bu texnologiya həm fərdlər, həm bizneslər, həm də hökumət üçün böyük potensiala malikdir.

Oxucular üçün yekun fikir və təkliflər. Ağıllı şəhərlər şəhərlərin inkişafında növbəti təkamüldür. Bu texnoloji inkişaf şəhər yerlərində yaşayanların həyatını

asanlaşdırmaq və daha davamlı etmək üçün bir çox imkanlar təqdim edir. Bununla belə, ağıllı şəhərlərin uğurlu inkişafı üçün müəyyən çətinliklər və maneələr var. Oxucular ağıllı şəhərlərin gətirəcəyi potensial faydalar və dəyişikliklər haqqında daha çox öyrənmə bilirlər. Onlar həmçinin texnoloji tendensiyaları və irəliləyişləri izləməklə ağıllı şəhərlərin gələcək inkişafına töhfə verə bilirlər. Ağıllı şəhərin inkişafını davam etdirmək üçün hökumətlər, yerli hökumətlər və özəl sektor əməkdaşlıq etməlidir. Bundan əlavə, ictimaiyyətin iştirakı və qarşılıqlı əlaqə ağıllı şəhərlərin uğurunun açarıdır. Ağıllı şəhərlərin davamlı inkişafı üçün ictimaiyyətin ehtiyac və gözləntilərinin nəzərə alınması vacibdir. Nəticədə, ağıllı şəhərlərin inkişafı daha yaxşı həyat keyfiyyəti, davamlılıq, səmərəlilik və təhlükəsizlik təmin etmək potensialına malikdir. Ona görə də ağıllı şəhərlərin inkişafının davam etdirilməsi şəhərlərin gələcək inkişafında mühüm rol oynayacaq [10].

Gələcək işlər – mobil tətbiq, sponsorluqlar. Bu günün sürətli rəqəmsal dövründə veb-sayt layihələri iləri getmək üçün görünməmiş çətinliklərlə üzləşir. Daim inkişaf edən texnologiya ilə ayaqlaşmaq üçün veb-sayt layihə komandamız daim öz strategiyalarımızı təkmilləşdirmək üzrə çalışacağıq. Veb sayt layihələrinin uğuru üçün getdikcə kritik hala gələn perspektiv strategiyalardan biri gələcəkdə mobil proqramlar və tərəfdaşlıqların işə salınması planıdır. Mobil proqramlar son illərdə əksər müəssisələrin rəqəmsal strategiyasının təməl daşına çevrilib. Mobil cihazların artması insanlara istənilən vaxt, hər yerdə və istənilən cihazdan internetə çıxış imkanı verib. Bu, onların gözləntilərini və davranışlarını kəskin şəkildə dəyişdirdi. İnsanlar indi mobil proqramlar vasitəsilə sevimli brendləri və veb-saytları ilə əlaqə qura biləcəklərini gözləyirlər. Mobil proqram həlləri təklif etməyən veb-saytlar öz auditoriyasını itirmək riski daşıyır. Veb sayt layihəsinin uğurunu təmin etmək üçün mobil proqramların hazırlanmasını nəzərə almaq vacibdir. Mobil tətbiqlər auditoriyamızın veb saytımla daha immersiv və istifadəçi dostu platformada əlaqə saxlamasına imkan verəcək. Mobil proqramlar push bildirişləri, oflayn giriş və daha sürətli yükləmə vaxtları kimi üstünlüklər təklif edir. Mobil proqramların hazırlanması həm də istifadəçi cəlbini və sədaqətini artırmağa kömək edə bilər, nəticədə veb-saytımızın trafikini artırır. Tərəfdaşlıq veb-sayt layihələrinin uğuru üçün də çox vacibdir [11]. Tərəfdaşlarla

işləmək müəssisələrdə tez-tez mövcud olan maddi çətinlikləri aradan qaldırmağa kömək edə bilər və şirkətlərin məqsədlərinə çatmasını asanlaşdırır. Tərəfdaşlıqlar sizə başqalarının təcrübəsindən yararlanmağa, yeni gəlir mənbələri tapmağa və yeni müştəri bazarlarına daxil olmağa imkan verə bilər. Tərəfdaşlıqları uğurla başlamaq üçün veb-sayt layihəsi potensial tərəfdaşları müəyyən etməli və təbliğat strategiyası hazırlamalıdır. Tərəfdaşların sizdən nəyə ehtiyacı olduğunu və onlara necə dəyər verə biləcəyinizi öyrənin. Tərəfdaşlıq üçün hədəflerinizi müəyyən etmək də vacibdir. Məqsəd gəliri artırmaq, yeni bazara çıxmaq və ya müştəri təcrübəsini təkmilləşdirməkdir? Məqsədlərinizi bilmək düzgün tərəfdaşlar tapmağa və düzgün strategiya hazırlamağa kömək edəcək. Maraqlı tərəfdaşlıq fərsətlərindən biri e-ticarət saytlarıdır. Bir çox müəssisə onlayn hərəkət edir və e-ticarət saytları daha çox müştəriyə çatmaq üçün əla yoldur. E-ticarət saytları ilə əməkdaşlıq veb saytınızın gəlirini, müştəri bazasını və brendin görünməsinə artırmağa kömək edə bilər. Siz həmçinin müştərilərinizə eksklüziv endirimlər, promosyonlar və ya loyallıq proqramları kimi dəyər təklif edə bilərsiniz. Digər potensial tərəfdaşlıq sosial media platformaları ilə bağlıdır. Sosial media son bir neçə ildə çox böyüdü və güclü marketinq alətinə çevrildi. Sosial media platformaları ilə əməkdaşlıq veb saytınızın əhatə dairəsini və brendin görünməsinə artırmağa kömək edə bilər. Sosial media platformaları həmçinin hədəf auditoriyanızın davranışları, üstünlükləri və maraqları haqqında dəyərli fikirlər təqdim edə bilər ki, bu da sizə daha effektiv marketinq strategiyası hazırlamağa kömək edə bilər. Nəticə olaraq, veb-sayt layihələrinin gələcəyi onların innovasiya etmək və əyridən qabaqda qalmaq qabiliyyətinə bağlıdır. Mobil proqramların inkişafı və tərəfdaşlıq rəqəmsal dünyada rəqabətə davamlı olmaq istəyən veb-sayt layihələri üçün iki mühüm strategiyadır. Veb saytınızı səfərbər etmək və sənaye liderləri ilə tərəfdaşlıq veb saytınızın istifadəçi təcrübəsini, gəlirini, müştəri bazasını və brendin görünməsinə əhəmiyyətli dərəcədə yaxşılaşdırmağa kömək edə bilər və nəticədə uğurlu və davamlı veb-sayt layihəsinə gətirib çıxara bilər. Beləliklə, veb-sayt layihələri daim inkişaf edən rəqəmsal ekosistemdə rəqabət aparmağa ümid edirlərsə, bu gün planlaşdırmağa başlamalıdırlar.

İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT

1. Caprotti F., Dayot Y., Schraven D., Sengers F., Joss S., The Smart City as Global Discourse: Storylines and Critical Junctures Across 27 Cities, *Journal of Urban Technology*, 2019, 26(1), 3-34.
2. Cowley R., Joss S., Dayot Y., The smart city and its publics: insights from across six UK cities, *Urban Research & Practice*, DOI:10.1080/17535069.2017.1293150.
3. Hammersmith Group, The Internet of Things: Networked Objects and Smart Devices, February 2010, <https://www.theinternetofthings.eu>.
4. "Login Authentication" – Santa Clara University
5. *Eloquent JavaScript*, 3rd edition (2018). Written by Marijn Haverbeke.
6. *Learning React: Functional Web Development with React and Redux 1st Edition*
7. *The Complete Guide to React User Authentication with Auth0* (November 2022)
8. C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, Foundations For Smarter Cities, *IBM Journal of research and development*, Vol 54, Number 4, pages 1-16, 2010.
9. P. Neirotti, A. De Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, Current Trends in Smart City Initiatives: Some Stylised Facts, *Cities*, Vol 38, pages 25-36, 2014.
10. S. Alawadhi, A. Aldama-Nalda, H. Chourabi, J R. Gil-Garcia, S. Leung, S. Mellouli, T. Nam, T. A Pardo, H. J Scholl, and S. Walker, Building Understanding of Smart City Initiatives, *International Conference on Electronic Government*, pages 40-53, 2012.
11. S. P Mohanty, U. Choppali, and E. Kougianos, Everything You Wanted to Know About Smart Cities: The Internet of Things is The Backbone, *IEEE Consumer Electronics Magazine*, Vol 5, Number 3, pages 60-70, 2016.