

AZƏRBAYCAN RESPUBLİKASI ELM VƏ TƏHSİL NAZİRLİYİ
AZƏRBAYCAN TEXNİKİ UNİVERSİTETİ
İNFORMASIYA VƏ TELEKOMMUNİKASIYA TEXNOLOGİYALARI
FAKÜLTƏSİ

Əlyazma hüququnda

İxtisasın şifri və adı: 060509 - Kompüter elmləri

İxtisaslaşmanın adı: Optimal idarəetmənin riyazi -iqtisadi üsulları

Mühəndis riyaziyyatı və süni intellekt

kafedrasının magistrantı

Aslanova Nuranə Orxan qızının

magistr elmi dərəcəsi almaq üçün

“KOLLEC TƏLƏBƏLƏRİNİN PRAKTİKİ İXTİSAS BİLİKLƏRİNİN
QİYMƏTLƏNDİRİLMƏSİ ÜÇÜN İNTELLEKTUAL METODLARIN
İŞLƏNMƏSİ
mövzusunda

DİSSERTASIYA İŞİ

Kafedra müdiri:

t.e.d., professor Ağayev N.B

Elmi rəhbər:

r.ü.f.d., dosent Zamanova R.B

Bakı-2023

MÜNDƏRİCAT

GİRİŞ.....	4
İXTİSARLARIN SİYAHISI.....	8
I FƏSİL. PRAKTİKİ İXTİSAS BİLİKLƏRİNİN YOXLANILMASININ TƏNZİMLƏNMƏSİ VƏ ONUN BƏZİ MÜHÜM NƏZƏRİ ASPEKTLƏRİ.....	9
1.1. Praktiki ixtisas biliklərinin yoxlanılmasının ümumi testi və tənzimləyici çərçivəsi.....	9
1.2. Praktiki ixtisas biliklərinin yoxlanması testinin bəzi nəzəri aspektləri.....	10
1.3. Praktiki ixtisas biliklərinin yoxlanması üçün yaradılmış intellektual sistemin əhəmiyyəti.....	13
II FƏSİL. PRAKTİKİ İXTİSAS BİLİKLƏRİNİN YOXLANILMASI ÜÇÜN YARADILMIŞ İNTELLEKTUAL SİSTEMİN ƏSAS KONFİQURASIYASI.....	15
2.1. İntellektual sistemin yaradılması üçün nəzərdə tutulmuş proqramlaşdırma dillərinin nəzəri əsasları	15
2.2. İntellektual sistem üçün nəzərdə tutulmuş verilənlər bazasının əsas strukturu və idarəedilməsinin əsasları.....	24
2.3. Yaradılmış intellektual sistemin təhlükəsizliyinin təmin edilməsi üçün görülmüş işlər.....	26
2.4. Yaradılmış intellektual sistem nəzərdə tutulan auditoriyanın ehtiyaclarına nə dərəcədə cavab verir ?	30
III FƏSİL. YARADILMIŞ İNTELLEKTUAL SİSTEMİN KOD NÜMUNƏLƏRİ.....	34

3.1. Proyeğin ümumi strukturu.....	34
3.2. Veb layihəsinin ümumi strukturunda "app" kataloq faylları.....	43
3.3. Veb layihəsinin ümumi strukturunda "models" kataloq faylları.....	47
NƏTİCƏ.....	51
İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT.....	52

GİRİŞ

Mövzunun aktuallığı. Praktik peşə biliklərinin yoxlanılması müxtəlif səbəblərə görə çox aktualdır. Bir çox sahələrdə praktiki bacarıq və biliklər peşəkar səriştənin vacib komponentləridir və bu bacarıqların yoxlanılması tələbələrin öz vəzifələrini effektiv və təhlükəsiz şəkildə yerinə yetirməyə hazır olmalarını təmin etməyin vacib hissəsidir. Praktik peşə biliklərinin yoxlanılmasının müvafiq olduğu bəzi xüsusi üsullar bunlardır:

İctimai təhlükəsizliyin təmin edilməsi: Bir çox peşələr əhalinin təhlükəsizliyinə və rifahına birbaşa təsir göstərir. Məsələn, həkimlər, tibb bacıları və digər səhiyyə işçiləri təhlükəsiz və effektiv qayğı göstərmək üçün xəstəliklərin diaqnostikası, dərmanların idarə edilməsi və cərrahi prosedurların yerinə yetirilməsi kimi bir sıra praktiki bacarıqları yerinə yetirməyi bacarmalıdırlar. Eynilə, pilotlar, mühəndislər və mürəkkəb texnika və ya nəqliyyatla əlaqəli sahələrdə çalışan digər mütəxəssislər ictimai təhlükəsizliyi təmin etmək üçün praktiki bacarıqlar nümayiş etdirə bilməlidirlər. Tənzimləyici orqanlar praktiki bacarıqları yoxlayaraq, bu sahələr üzrə mütəxəssislərin lazımı qaydada təlim keçməsinə və öz vəzifələrini təhlükəsiz və effektiv şəkildə yerinə yetirə bilmələrinə kömək edə bilər.

Peşəkarlığın artırılması: Bir çox sahələrdə praktiki bacarıqlar peşəkar səriştə və etibarlılığın mühüm tərkib hissəsidir. Təcrübəçilərdən praktiki biliklərini nümayiş etdirmələrini tələb etməklə, tənzimləyici orqanlar peşəkarlıq və hesabatlılıq mədəniyyətinin inkişafına kömək edə bilər. Bu, praktikantların davamlı öyrənmə və inkişafa sadıq olmalarını və onların yüksək keyfiyyət və performans standartlarına riayət etmələrini təmin etməyə kömək edə bilər.

Hərəkətliliyin asanlaşdırılması: Bəzi peşələrdə tələbələr işləmək üçün müxtəlif şəhərlər və ya ölkələr arasında hərəkət etməli ola bilər. Tənzimləyici orqanlar praktiki bacarıqları sınaqdan keçirməklə praktikantların təhsil və ya təlimi harada almasından asılı olmayaraq eyni səriştə standartlarına cavab verməsini təmin etməklə bu mobilliyi asanlaşdırmağa kömək edə bilər. Bu, peşəkar səriştənin qiymətləndirilməsində ardıcillıq və ədalətliliyi təşviq etməyə kömək edə bilər. Həmçinin tələbələrin təcrübə keçirdikləri yerdə eyni yüksək standartlara riayət olunmasını təmin etməklə ictimaiyyəti qorumağa kömək edə bilər.

Ümumilikdə, praktiki peşə biliklərinin yoxlanılması praktikantların öz vəzifələrini təhlükəsiz və effektiv şəkildə yerinə yetirməyə hazır olmalarının və onların yüksək peşəkarlıq və hesabatlılıq standartlarına riayət etmələrinin təmin edilməsinin vacib hissəsidir.

İşin əsas məqsədi. Praktiki peşə biliklərinin yoxlanmasının əsas məqsədi fərdlərin öz peşə vəzifələrini səriştəli və təhlükəsiz şəkildə yerinə yetirmək üçün lazımi bacarıq və biliyə malik olub-olmamasının qiymətləndirməkdir. Praktiki peşə biliklərinin yoxlanılmasının bəzi məqsədləri bunlardır:

Bacarıqların təmin edilməsi: Tənzimləyici orqanlar praktiki bacarıqları yoxlayaraq praktikantların lazımi qaydada təlim keçməsinə və öz vəzifələrini təhlükəsiz və effektiv şəkildə yerinə yetirə bilmələrini təmin edə bilər.

Standartların müəyyən edilməsi: Bu, peşəkar səriştənin qiymətləndirilməsində ardıcillıq və ədalətliliyi təşviq etməyə kömək edə bilər, həmçinin praktikantların yüksək keyfiyyət və performans standartlarına riayət etmələrini təmin etməklə ictimaiyyəti qorumağa kömək edə bilər.

Peşəkar inkişafın təşviqi: Təcrübəçilərdən müntəzəm olaraq öz praktiki biliklərini

nümayiş etdirmələrini tələb etməklə, onların ixtisaslarını nə dərəcədə öyrəndikərinə nəzarət edə bilər.

İşin praktiki əhəmiyyəti. Aydındır ki, müxtəlif ixtisas qruplarında təhsil alan tələbələr üçün müəyyən müddət aralığında tədris olunan riyazi və ya digər praktiki ixtisas biliklərinin yoxlanılması tələbələrin keçirilən fənn və ya daha detallı desək, mövzunu qavrama qabiliyyətinin yoxlanılması vacib aspektlərdən biridir. Buna nəzərən yaradılmış intellektual sistem müntəzəm olaraq keçirilməsinin çətinlik yarada biləcəyi praktiki biliklərin yoxlanılması üçün nəzərdə tutulmuş testlər hər hansı bir müdaxilə olmadan ədalətli keçirilməsini özündə ehtiva edir.

Tədqiqat işinin metodoloji və nəzəri əsasları. Mövzu əsasən kollec tələbələrinin praktiki ixtisas biliklərinin yoxlanılması üçün yaradılmış intellektual sistemi əhatə edir. İntellektual sistemin ümumi quruluşu tələbə, müəllim və əsas tənzimləyici orqan tərəfindən idarə oluna bilən model olaraq kodlaşdırılmış və nəzərdə tutulan X kollecinin ixtisas fənnlərini əhatə edir.

Dissertasiya işinin həcmi, quruluşu və qısa məzmunu. Dissertasiya işinin həcmi 51 səhifədir. İş girişdən, üç fəsildən, nəticələrdən, istifadə olunmuş ədəbiyyat siyahısından ibarətdir.

Girişdə yerinə yetirilmiş dissertasiya işinin mövzusunun aktuallığı, onun predmeti və əsas məqsədi əsaslandırılmış, elmi yeniliyi barəsində aydın mühakimələr aparılmışdır. “praktiki ixtisas biliklərinin yoxlanılmasının tənzimlənməsi və onun bəzi mühüm nəzəri aspektləri” adlanan birinci fəsildə praktiki ixtisas biliklərinin yoxlanılması ön plana çəkilərək onun zəruriliyi əsaslandırılmış, bu istiqamətdə yaradılan intellektual sistemlərin prioritet istiqamətlərinin strategiyası təhlil olunmuşdur.

“Praktiki ixtisas biliklərinin yoxlanılması üçün yaradılmış intellektual sistemin əsas konfigurasiyası” adlanan ikinci fəsildə isə proyektin istifadəçi

interfeysini əhatə edən əsas struktur elementləri qeyd olunmuş və ümumi kataloq faylları haqqında mülahizələr qeyd olunmuşdur.

“Yaradılmış intellektual sistemin kod nümunələri” adlanan üçüncü fəsildə isə kataloq fayllarında yerləşən php uzantılı faylların kod nümunələrinə nəzər salınmış, vacib kodu nümunələrinin işləmə prinsipləri izah edilmişdir.

Nəticə bölməsində tədqiq olunan mövzu barəsində magistrın bəzi şəxsi fikir və mülahizələri öz əksini tapmışdır.

İXTİSARLARIN SİYAHISI

HTML – Hyper Text Markup Language

CSS – Cascading Style Sheet

PHP- Personal Home Page

HTTP - HyperText Transfer Protocol

API - Application Programming Interface

SQL – Structured Question Language

SSL - Secure Sockets Layer

TLS - Nəqliyyat Layeri Təhlükəsizliyi

AES - Qabaqcıl Şifrələmə Standartı

SHA- Təhlükəsiz Hash Alqoritmi

MVC - Model-View-Controller

ORM - Object Relation Mapping

FƏSİL 1

PRAKTİKİ İXTİSAS BİLİKLƏRİNİN YOXLANILMASININ TƏNZİMLƏNMƏSİ VƏ ONUN BƏZİ MÜHÜM NƏZƏRİ ASPEKTLƏRİ

1.1. Praktiki ixtisas biliklərinin yoxlanılmasının ümumi testi və tənzimləyici çərçivəsi

Praktik ixtisasların yoxlanılması üçün ümumi test və normativ baza yoxlanılan peşə və ya sahədən asılı olaraq dəyişə bilər. Bununla belə, bir çox praktiki imtahan növlərinə aid olan bəzi ümumi prinsiplər və təcrübələr mövcuddur. Praktiki ixtisasların yoxlanılması üçün test və normativ bazanın bəzi əsas komponentləri bunlardır:

Testin dizaynı: Təcrübəli imtahanın hazırlanmasında ilk addım testin özünü dizayn etməkdir. Bu, yoxlanılan peşə və ya sahəyə uyğun olan əsas bacarıq və bilik sahələrinin müəyyən edilməsini və bu səlahiyyətləri qiymətləndirən bir sıra test tapşırıqlarının hazırlanmasını əhatə edir. Test ədalətli, etibarlı olmalıdır, yəni qiymətləndirmək üçün nəzərdə tutulmuş bilik və bacarıqları dəqiq ölçür və zamanla ardıcıl nəticələr verir.

Testin idarə edilməsi: Test tərtib edildikdən sonra, namizədlərə ardıcıl və standartlaşdırılmış şəkildə idarə edilməlidir. Bu, namizədlərin imtahan verə biləcəyi test mərkəzlərinin və ya yerlərin yaradılmasını, namizədlərin test təlimatlarına əməl etmələrini təmin etmək üçün xüsusi nizamlayıcı sistemin təmin edilməsini və

imtahan nəticələrinin qiymətləndirilməsi və hesabat verilməsi üçün uyğun model və sistemin yaradılması, prosedurlarının qurulmasını əhatə edə bilər.

Tənzimləyici çərçivə: Praktik ixtisasların yoxlanılması çox vaxt peşəkar təşkilatlar, dövlət qurumları və ya digər qurumlar tərəfindən tənzimləyici nəzarətə məruz qalır. Bununla belə nəzərdə tutulmuş intellektual sistem kollecin nəznində olduğu əsas götürülərək, əvvəlcədən müəyyən edilmiş nizamlayıcılar tərəfindən sadəcə uyğun qrup tələbələrinin və müəllimlərinin girişinə icazə verəcək. Normativ bazaya imtahanın işlənilib hazırlanmasını və idarə edilməsini tənzimləyən qaydalar, və ya standartlar, eləcə də kimin imtahan vermək hüququna malik olduğunu və imtahan nəticələrinin necə istifadə olunacağını müəyyən edən meyarlar daxildir.

Ümumilikdə, praktiki ixtisasların yoxlanılması fərdlərin öz peşə vəzifələrini səriştəli və təhlükəsiz şəkildə yerinə yetirmək üçün lazımi bacarıq və biliklərə malik olmasını təmin etmək üçün hazırlanmış test və normativ bazanı əhatə edir. Tənzimləyici orqanlar praktiki bacarıq və bilikləri yoxlayaraq praktikantların lazımi qaydada təlim keçməsinə və öz vəzifələrini təhlükəsiz və effektiv şəkildə yerinə yetirə bilmələrini təmin edə bilər.

1.2. Praktiki ixtisas biliklərinin yoxlanması testinin bəzi nəzəri aspektləri

Praktiki ixtisas imtahanı fərdin konkret peşə və ya sahədə praktiki bacarıq və biliklərini qiymətləndirmək üçün nəzərdə tutulub. Bununla belə, imtahanı hazırlayarkən və idarə edərkən nəzərə alınması vacib olan bir neçə nəzəri aspekt də var. Praktiki kvalifikasiya imtahanının bəzi əsas nəzəri aspektləri bunlardır:

Koqnitiv Mürəkkəbliik: Praktiki ixtisas testinin nəzəri cəhətlərindən biri idrak mürəkkəbliiyidir. Bu, tapşırıqları yerinə yetirmək və ya imtahan suallarına cavab vermək üçün tələb olunan düşüncə səviyyəsinə aiddir. Təhlil, sintez və ya qiymətləndirmə kimi daha yüksək səviyyəli düşüncə tələb edən tapşırıqlar, ümumiyyətlə, sadə xatırlama və ya tanınma tələb edən tapşırıqlardan idrak baxımından daha mürəkkəb hesab olunur. Testin koqnitiv mürəkkəbliik səviyyəsi yoxlanılan peşə və ya sahədə tələb olunan təcrübə səviyyəsinə uyğun olaraq tərtib edilməsi düşünülmüş və nəzərdə tutulmuş şəxslər üçün qaydalar əvvəlcədən tərtib ediləcəkdir.

Quruluşun etibarlılığı: Praktiki kvalifikasiya testinin digər mühüm nəzəri aspekti konstruksiya etibarlılığıdır. Bu, testin ölçülməsi nəzərdə tutulan konstruksiya və ya konsepsiyayı ölçmə dərəcəsinə aiddir. Məsələn, əgər test müəyyən bir riyazi tənliyin həll edilməsi qabiliyyətini ölçmək üçün nəzərdə tutulubsa, test tapşırıqları birbaşa həmin tapşırıqla əlaqəli olmalı və müvafiq bilik və bacarıqları qiymətləndirməlidir.

Etibarlılıq: Etibarlılıq praktiki ixtisas imtahanının digər əsas nəzəri aspektidir. Bu, test nəticələrinin zamanla və müxtəlif test iştirakçıları arasında ardıcılığına aiddir. Etibarlı test, testin nə vaxt və harada aparılmasından və kimin imtahan verməsindən asılı olmayaraq düzgün nəticələr verməlidir. Buna testin idarə edilməsinin, qiymətləndirmənin və şərhin standartlaşdırılması yolu ilə nail olmaq olar.

Ayrı-seçkilik: Ayrı-seçkilik testin tələb olunan bilik və bacarıqlara malik olan və olmayan şəxsləri ayırd etmək qabiliyyətinə aiddir. Yüksək ayrı-seçkilik yaradan test, sözügedən tapşırıqları yerinə yetirmək üçün uyğun olanları daha yaxşı müəyyən edə bilir. Ayrı-seçkilik bilik və bacarıq səviyyələrini fərqləndirən yaxşı tərtib edilmiş test tapşırıqlarından istifadə etməklə gücləndirilə bilər.

Ədalətlik: Ədalət praktiki ixtisas imtahanının mühüm nəzəri aspektidir. Test, keçmişdən və ya digər xüsusiyyətlərindən asılı olmayaraq, bütün imtahan verənlər üçün ədalətli şəkildə tərtib edilməli və idarə edilməlidir. Bu, test tapşırıqlarının mədəni cəhətdən neytral olmasını, əlilliyi olan şəxslər üçün şəraitin təmin edilməsini və imtahan nəticələrinin qiymətləndirilməsində və ya şərhində heç bir qərəzliliyin olmamasından ibarət ola bilər.

Gizlilik: Nəhayət, gizlilik praktiki ixtisas biliklərinin yoxlanılması üçün nəzərdə tutulmuş intellektual sistemin əsas aspektlərindən biridir. Yaradılan intellektual sistemin əsas diqqət verdiyi məqamlardan biri sistemin tənzimləyici şəxsdən başqa girişə icazə verməməsi, imtahanın gedişatına kimsənin müdaxilə etməməsi, nəticələrin sadəcə müəllim və imtahan verən şəxs tərəfindən görülməsinə və nəzərdə tutulan test tapşırıqlarının müəllimlər tərəfindən tərtib edilməsi nəzərdə tutulmuşdur.

Ümumiyyətlə, yüksək keyfiyyətli və effektiv imtahan hazırlamaq və idarə etmək üçün praktiki ixtisas imtahanının nəzəri aspektlərini başa düşmək vacibdir. Tənzimləyici orqanlar koqnitiv mürəkkəblilik, konstruktiv etibarlılıq, etibarlılıq, ayrışçılıq və ədalətlik kimi amilləri nəzərə alaraq testin müəyyən bir peşə və ya sahədə təhlükəsiz və effektiv təcrübə üçün tələb olunan bacarıq və bilikləri dəqiq qiymətləndirməsinə kömək edə bilər.

1.3. Praktiki ixtisas biliklərinin yoxlanılması üçün yaradılmış intellektual sistemin əhəmiyyəti

İlk öncə qeyd edim ki, praktik peşə biliklərinin yoxlanılması üçün yaradılmış intellektual sistem müəyyən kollec tələbələrinin lazımi bacarıq və biliklərə malik olmasını təmin etmək üçün mühüm vasitə olaraq nəzərdə tutulmuşdur. İntellektual sistemin vacib olmasının bəzi əsas səbəbləri bunlardır:

Standartlaşdırma: İntellektual sistemin əsas üstünlüklərindən biri praktiki peşəkar biliklərin qiymətləndirilməsinin standartlaşdırılmasına kömək etməsidir. Standartlaşdırılmış test tapşırıqları toplusundan istifadə etməklə sistem imtahan verən bütün şəxslərin eyni meyarlar əsasında qiymətləndirilməsini təmin edir. Müxtəlif zaman aralıklarında imtahan verə biləcək tələbələrə nəzərə alaraq – test suallarının yayılma ehtimalına qarşı bazada nəzərdə tutulmuş müəllimlər tərəfindən imtahan üçün düşünülmüş test suallarının 3 dəfə artıq test yerləşdirilməsi əsas fəaliyyətlərdən biridir. Eyni zamanda imtahan verəcək olan tələbələr isə imtahan prosesində eyni suallarla deyil, bazada yerləşən sualların random şəkildə

paylanmasına əsasən qurulmuş test toplusundan imtahan verəcək. Bu, imtahan nəticələrinin ədalətli və dəqiq olmasına və imtahandan keçən şəxslərin ardıcıl bilik və bacarıq səviyyəsini nümayiş etdirməsinə kömək edir.

Keyfiyyətə nəzarət: İntellektual sistemin digər mühüm faydası, yoxlanılan peşə və ya sahədə keyfiyyətə nəzarəti təmin etməyə kömək etməsidir. Təcrübə üçün tələb olunan bilik və bacarıqlar üçün yüksək zolağı təyin etməklə sistem yalnız ixtisaslı şəxslərin bu sahədə təcrübə keçməsinə kömək edir. Bu, ictimaiyyətin təhlükəsizliyini qorumağa və peşə və ya sahənin nüfuzunu artırmağa kömək edə bilər. Eyni zamanda təhsil alan tələbənin tədris olunan fənlərə və mövzulara nə dərəcədə yiyələnməsi üçün statistika da təyin oluna bilər.

Peşəkar inkişaf: İntellektual sistem də peşəkar inkişafda mühüm rol oynaya bilər. Təcrübə üçün tələb olunan əsas bilik və bacarıqları müəyyən etməklə sistem fərdlərə əlavə təlim və ya təhsilə ehtiyac duyduqları sahələri müəyyən etməyə kömək edə bilər. Bu, peşə və ya sahə daxilində davamlı öyrənmə və inkişafı təşviq etməyə kömək edə bilər ki, bu da öz növbəsində tələbələr üçün daha yaxşı nəticələrə səbəb ola bilər.

Karyera yüksəlişi: İntellektual sistem həm də peşə və ya sahədə karyera yüksəlişi üçün yol təmin edə bilər. Fərdlərdən praktiki imtahan vasitəsilə bilik və bacarıqlarını nümayiş etdirmələrini tələb etməklə, sistem liderlik və ya ixtisaslaşmış rollar üçün ən uyğun olanları müəyyən etməyə kömək edə bilər. Bu, karyera yüksəlişi və peşəkar inkişaf üçün imkanlar yaratmağa kömək edə bilər.

Ümumiyyətlə, praktiki peşə biliklərinin yoxlanılması üçün yaradılmış intellektual sistem müəyyən bir peşə və ya sahədə çalışan şəxslərin öz işlərini səmərəli və təhlükəsiz şəkildə yerinə yetirmək üçün lazımi bacarıq və biliklərə malik olmasını

təmin etmək üçün mühüm vasitədir. Standartlaşdırmanı, keyfiyyətə nəzarəti, peşəkar inkişafı, karyera yüksəlişini və qanunlara uyğunluğu təşviq etməklə sistem praktikantların müştərilərinin və ya xəstələrinin ehtiyaclarını ödəmək üçün yaxşı hazırlanmış və ixtisaslı olmasını təmin etməyə kömək edə bilər.

FƏSİL 2

PRAKTİKİ İXTİSAS BİLİKLƏRİNİN YOXLANILMASI ÜÇÜN YARADILMIŞ İNTELLEKTUAL SİSTEMİN ƏSAS KONFİQURASIYASI

2.1. İntellektual sistemin yaradılması üçün nəzərdə tutulmuş proqramlaşdırma dillərinin nəzəri əsasları

İntellektual sistem dedikdə veb saytın yaradılması nəzərdə tutulmuşdur. Veb sayt internet vasitəsilə daxil olan və veb-brauzerdə nümayiş etdirilən veb səhifələr toplusudur. O, adətən mətn, şəkillər, videolar və interaktiv elementlər kimi strukturlaşdırılmış şəkildə təşkil edilən və təqdim edilən məlumat və ya məzmunu ehtiva edir. Veb-saytın əsas strukturu ana səhifə, naviqasiya menyuları və digər daxili səhifələrdən ibarət olaraq tərtib edilmişdir. İntellektual sistemin yaradılması üçün nəzərdə tutulmuş proqramlaşdırma dilləri dinamik, interaktiv və responsive (bütün kateqoriyalara uyğun gələn) veb-saytın inkişafını asanlaşdırmaq məqsədi daşıyan bir neçə nəzəri əsaslara əsaslanır. Veb-saytın strukturunun qurulma prinsiplərinə əsasən istənilən cihazdan – planşet, notbuk, mobil telefon və komputerdən giriş mümkündür və dizayn hər bir cihaz üçün ayrılıqda tərtib edilmiş və sınaqdan keçirilmişdir. İntellektual sistemi yaratmaq üçün istifadə olunan proqramlaşdırma dillərinin əsas nəzəri əsaslarından bəziləri bunlardır:

Müştəri-Server Arxitekturası: Müştəri-server arxitekturası müştərinin (adətən veb-brauzer) serverdən (veb server) xidmətlər və ya resurslar tələb etdiyi hesablama modelidir. Server müştərinin sorğularına cavab verir və lazımi resursları təmin edir. Müştəri-server arxitekturası veb-saytın qurulmasında istifadə olunan ümumi modeldir. Bu arxitektura veb-sayt iki əsas komponentə bölünür: müştəri tərəfi və server tərəfi.

Müştəri tərəfi veb-saytın istifadəçinin kompüterində və ya mobil cihazında işləyən hissəsinə aiddir. O, istifadəçi interfeysini təqdim etmək və istifadəçi qarşılıqlı əlaqələrini idarə etmək üçün məsuliyyət daşıyır. Tipik olaraq, müştəri tərəfi komponentləri HTML, CSS və JavaScript kimi veb texnologiyalarından istifadə etməklə qurulur. Bu komponentlər istifadəçinin brauzerinə çatdırılır və onun cihazında işləyir.

Server tərəfi veb-saytın serverdə işləyən hissəsinə, yəni veb-saytı yerləşdirən kompüter və ya kompüterlər qrupuna aiddir. Server tərəfi istifadəçi sorğularının işlənməsi, cavabların yaradılması və məlumatların saxlanması üçün idarə edilməsinə cavabdehdir. Tipik olaraq, server tərəfi komponentlər PHP, Python və ya Java kimi server tərəfi proqramlaşdırma dillərindən istifadə etməklə qurulur. Bizim tərəfimizdən yaradılmış intellektual sistem – veb-saytda PHP proqramlaşdırma dilindən istifadə edilmişdir.

Müştəri tərəfi və server tərəfi arasında əlaqə HTTP sorğuları və cavablarından istifadə etməklə asanlaşdırılır. İstifadəçi veb-saytla əlaqə qurduqda, məsələn, düyməni klikləməklə və ya formanı doldurmaqla, müştəri tərəfi server tərəfinə HTTP sorğusu göndərir. Daha sonra server tərəfi sorğunu emal edir, cavab yaradır və HTTP cavabı şəklində müştəri tərəfinə geri göndərir.

İstifadəçi tərəfindən server tərəfinə HTTP sorğusu göndərilən zaman baş verən hadisə bu şəkildə izah edilir.

Müştəri tərəfi server tərəfinə HTTP sorğusu göndərdikdə, bu o deməkdir ki, müştəri tərəfi server tərəfdən müəyyən bir hərəkəti yerinə yetirməyi və ya serverdən məlumat əldə etməyi xahiş edir. HTTP, veb-saytın müştəri tərəfi və server tərəfi arasında məlumat mübadiləsi üçün qaydalar toplusunu təşkil edən Hypertext Transfer Protocol deməkdir. Budur, server tərəfinə HTTP sorğusu göndərən müştəri tərəfinin nümunə ssenarisi:

Tutaq ki, istifadəçi – tələbə və ya müəllim veb-sayta daxil olur və hesab məlumatlarına baxmaq üçün düyməni klikləyir. İstifadəçi düyməni kliklədikdə, müştəri tərəfi (istifadəçinin veb brauzeri) həmin istifadəçi üçün hesab təfərrüatlarını tələb edən server tərəfinə HTTP sorğusu göndərir.

HTTP sorğusu adətən bir neçə komponentdən ibarətdir:

HTTP metodu: Bu, edilən sorğunun növünü göstərir. Ümumi HTTP metodlarına GET (məlumatları əldə etmək üçün), POST (məlumat göndərmək üçün) və DELETE (məlumatları silmək üçün) daxildir.

URL: Bu, sorğu edilən server tərəfindəki resursun yerini müəyyən edir.

Başlıqlar: Bunlar göndərilən və ya qəbul edilən məlumatın növü kimi sorğu haqqında əlavə məlumat verir və autentifikasiya və keşləmə üçün istifadə edilə bilər.

Bədən: Bu, forma məlumatları və ya JSON kimi server tərəfinə göndərilməsi lazım olan hər hansı məlumatları özündə ehtiva edir.

Server tərəfi HTTP sorğusunu qəbul etdikdən sonra sorğunu emal edəcək və HTTP cavabı yaradacaq.

Cavab adətən aşağıdakı formalarda verilir:

HTTP status kodu: Bu sorğunun uğurlu olub-olmadığını və ya xəta ilə qarşılaşdığını göstərir.

Başlıqlar: Bunlar məzmun növü və uzunluğu kimi cavab haqqında əlavə məlumat verir.

Əsas mətn: Bu, hesab təfərrüatları və ya xəta mesajı kimi müştəri tərəfinə qaytarılan məlumatları ehtiva edir.

Müştəri tərəfi daha sonra HTTP cavabını alacaq və veb saytında tələb olunan məlumatları göstərərək məlumatları istifadəçiyə təqdim edəcək.

Nəticə olaraq, müştəri tərəfi server tərəfinə HTTP sorğusu göndərdikdə o, server tərəfdən məlumat və ya hərəkət tələb edir. Sonra server tərəfi sorğunu emal edir və müştəri tərəfinə geri göndərilən HTTP cavabı yaradır. Bu məlumat və fəaliyyət mübadiləsi bir çox veb-saytların və veb proqramların işləməsi üçün vacibdir. Müştəri-server arxitekturası tərtibatçılara veb saytın müxtəlif komponentlərini yaratmaq üçün müxtəlif texnologiyalardan və proqramlaşdırma dillərindən istifadə etməyə imkan verir, dizayn və icrada çeviklik təmin edir.

Bütövlükdə, müştəri-server arxitekturası çoxlu sayda istifadəçini idarə edə bilən, təhlükəsizlik təhdidlərindən qoruya bilən, məzmun və xidmətlərin çatdırılması üçün çevik və genişləndirilə bilən platforma təmin edən veb-saytlar yaratmaq üçün güclü vasitədir.

Veb saytın ön proqramlaşdırma hissəsi – frontend proqramlaşdırma veb-saytın istifadəçi interfeysi və istifadəçi təcrübəsi yaratmaq üçün nəzərdə tutulmuşdur. Bu, istifadəçilərin düymələr, menyular, formalar və digər qrafik komponentlər kimi gördüyü və qarşılıqlı əlaqədə olduğu elementlərin layihələndirilməsini və kodlaşdırılmasını nəzərdə tutur.

Frontend proqramlaşdırması adətən veb-saytın strukturunu, üslubunu və interaktivliyini yaratmaq üçün HTML, CSS və JavaScript kimi dillərdən istifadə etməyi əhatə edir. HTML veb-səhifənin strukturunu və məzmununu müəyyən etmək üçün, CSS isə veb-səhifənin üslubunu yaratmaq və onu vizual olaraq cəlbedici etmək üçün istifadə olunur. JavaScript veb-səhifədə pop-up pəncərələr, açılan menyular və animasiyalar kimi interaktiv və dinamik elementlər yaratmaq üçün istifadə olunur.

HTML (Hypertext Markup Language) frontend proqramlaşdırmada veb-səhifənin məzmununu yaratmaq və strukturlaşdırmaq üçün istifadə edilən standart işarələmə dilidir. Frontend proqramlaşdırmada HTML kodlarının əsas funksiyaları bunlardır:

Veb səhifənin strukturunun müəyyən edilməsi: HTML tərtibatçılara başlıqları, paraqrafları, siyahıları, şəkilləri və digər məzmun növlərini təyin etmək üçün bir sıra teqlərdən istifadə etməklə veb-səhifənin strukturunu müəyyən etməyə imkan verir. HTML teqlərindən istifadə edərək veb-səhifənin məzmununu strukturlaşdırmaqla tərtibatçılar CSS və JavaScript-dən istifadə edərək üslub və manipulyasiya edilə bilən elementlərin iyerarxiyasını yarada bilərlər.

Məzmun üçün semantik məna təmin etmək: HTML teqləri veb-səhifənin məzmununa semantik məna verir ki, bu da axtarış sisteminin optimallaşdırılması və əlçatanlıq üçün faydalı ola bilər. Məsələn, veb səhifənin başlıq hissəsini qeyd etmək

üçün <header> teqindən istifadə axtarış motorlarına və köməkçi texnologiyalara bu bölmənin səhifənin əsas başlığı olduğunu bildirir.

Linklərin və naviqasiyanın yaradılması: HTML tərtibatçılara müxtəlif səhifələr və ya veb səhifənin bölmələri arasında keçid etmək üçün istifadə oluna bilən <a> (lövbər) teqindən istifadə edərək veb-səhifələr arasında bağlantılar yaratmağa imkan verir. Tərtibatçılar həmçinin siyahılar və keçidlərdən istifadə edərək naviqasiya menyuları yarada bilirlər.

Multimedia məzmununun yerləşdirilməsi: HTML tərtibatçılara müvafiq olaraq , <video> və <audio> teqlərindən istifadə edərək şəkillər, videolar və audio faylları kimi multimedia məzmununu veb səhifəyə yerləşdirməyə imkan verir. Bu, istifadəçilərə birbaşa veb-səhifədə müxtəlif növ media məzmununa baxmaq və onlarla qarşılıqlı əlaqədə olmaq imkanı verir.

İstifadəçi daxiletməsi üçün formaların yaradılması: HTML <input>, <select> və <textarea> kimi forma elementləri toplusunu təmin edir ki, bu da tərtibatçılara istifadəçi girişi üçün giriş formaları, axtarış formaları və əlaqə formaları kimi formalar yaratmağa imkan verir.

Bütövlükdə, HTML strukturlaşdırılmış və semantik cəhətdən mənalı veb-səhifələrin yaradılması üçün əsas təmin etdiyi üçün frontend proqramlaşdırma üçün əsas dildir. HTML teqlərindən istifadə etməklə tərtibatçılar veb-səhifənin strukturunu, məzmununu və funksionallığını müəyyən edə və istifadəçilərin qarşılıqlı əlaqədə olması üçün istifadəçi dostu və interaktiv interfeys yarada bilirlər.

CSS (Cascading Style Sheets) ön proqramlaşdırmada HTML sənədlərinin təqdimatını və üslubunu təsvir etmək üçün istifadə edilən üslub cədvəli dilidir. Frontend proqramlaşdırmada CSS kodlarının əsas funksiyaları bunlardır:

Veb səhifənin dizaynı: CSS tərtibatçılara veb-səhifənin HTML elementlərinə üslublar, rənglər, şriftlər və digər vizual effektlər əlavə etməyə imkan verir. CSS seçicilərindən istifadə edərək HTML elementlərinə üslublar tətbiq etməklə tərtibatçılar veb səhifə üçün vizual cəlbedici və vahid dizayn yarada bilərlər.

Dizayn və yerləşdirmənin yaradılması: CSS tərtibatçılara veb-səhifədə HTML elementlərinin yerləşdirilməsinə və ölçüsünə nəzarət etməyə imkan verən bir sıra tərtibat və yerləşdirmə xüsusiyyətlərini təmin edir. Buraya mürəkkəb planlar və cavab verən dizaynlar yaratmaq üçün istifadə edilə bilən display, mövqe, float və grid kimi xüsusiyyətlər daxildir.

Responsiv dizayn: CSS, tərtibatçılara müxtəlif ekran ölçüləri və cihazlarına uyğunlaşan planlar və üslublar yaratmağa imkan verən bir sıra media sorğuları və responsiv dizayn üsulları təqdim edir. Bu, veb-səhifəyə masaüstü kompüterlərdən mobil telefonlara qədər müxtəlif cihazlarda baxmağa və istifadə etməyə imkan verir.

Animasiya və interaktivlik: CSS tərtibatçılara veb-səhifədə dinamik və interaktiv effektlər yaratmağa imkan verən bir sıra animasiya və keçid xüsusiyyətlərini təmin edir. Buraya animasiyalar, hover effektləri və digər interaktiv elementlər yaratmaq üçün istifadə edilə bilən transformasiya, keçid və animasiya kimi xüsusiyyətlər daxildir.

Əlçatanlıq: CSS mətn kontrastını təyin etmək, düzgün şrift ölçülərindən istifadə etmək və şəkillərə alternativ mətn təsvirləri əlavə etmək kimi veb-səhifənin əlçatanlığını yaxşılaşdırma bilən bir sıra xüsusiyyətlər və üsullar təqdim edir.

Ümumilikdə, CSS ön proqramlaşdırma üçün vacib bir dildir, çünki o, tərtibatçılara veb-səhifənin HTML məzmununa vizual üslub və effektlər əlavə etməyə və istifadəçilərin qarşılıqlı əlaqədə olması üçün vizual cəlbedici və istifadəçi dostu interfeys yaratmağa imkan verir. CSS-dən istifadə etməklə tərtibatçılar istifadəçilərinin ehtiyaclarına və seçimlərinə cavab verən cavab verən, əlçatan və cəlbedici veb səhifələr yarada bilərlər.

JavaScript veb-səhifələrdə ön proqramlaşdırma üçün istifadə olunan yüksək səviyyəli proqramlaşdırma dilidir. Frontend proqramlaşdırmada JavaScript kodlarının əsas funksiyaları bunlardır:

Dinamik veb səhifə davranışı: JavaScript istifadəçi qarşılıqlı əlaqəsinə və hadisələrə cavab verə bilən dinamik veb səhifələr yaratmaq üçün istifadə olunur. JavaScript ilə tərtibatçılar istifadəçi təcrübəsini və veb-səhifədə əlaqəni artırma bilən açılan menyular, pop-up pəncərələr və slayd şouları kimi interaktiv funksiyalar əlavə edə bilərlər.

Məlumatların yoxlanılması və formaların idarə edilməsi: JavaScript formalarda istifadəçi daxiletmələrini yoxlamaq və forma təqdimatlarını idarə etmək üçün istifadə edilə bilər. Bu, istifadəçilər tərəfindən daxil edilmiş məlumatların dəqiq və tam olmasını təmin etməyə kömək edə bilər və xətalara və təhlükəsizlik zəifliklərinin qarşısını ala bilər.

Brauzer uyğunluğu: JavaScript brauzer imkanları və uyğunluq problemlərindəki fərqləri aşkar etmək və idarə etmək üçün istifadə edilə bilər. Bu, veb-səhifənin müxtəlif brauzerlər və platformalarda düzgün işləməsini təmin etməyə kömək edə bilər və bütün istifadəçilər üçün ardıcıl istifadəçi təcrübəsi təmin edə bilər.

API və kitabxanalar: JavaScript veb-səhifə üçün əlavə funksionallıq və məlumat təmin edə bilən API-lərə (Tətbiq Proqramlaşdırma İnterfeysləri) və xarici kitabxanalara daxil olmaq və onlarla qarşılıqlı əlaqədə olmaq üçün istifadə edilə bilər. Bura sosial media platformaları, hava məlumatları və digər veb xidmətləri üçün API-lər daxildir.

Ümumilikdə, JavaScript frontend proqramlaşdırma üçün kritik dildir, çünki o, tərtibatçılara veb səhifəyə dinamik və interaktiv xüsusiyyətlər əlavə etməyə, istifadəçi daxiletmələrini və hadisələri idarə etməyə, real vaxt rejimində HTML məzmununu və strukturunu dəyişdirməyə imkan verir. JavaScript-dən istifadə etməklə tərtibatçılar cəlbedici, cavab verən və bütün istifadəçilər üçün qüsursuz istifadəçi təcrübəsi təmin edən veb səhifələr yarada bilərlər.

Frontend proqramlaşdırmanın məqsədi istifadəçilərə veb sayt və onun xüsusiyyətləri ilə qarşılıqlı əlaqə yaratmağa imkan verən vizual cəlbedici və istifadəsi asan interfeys yaratmaqdır. Frontend tərtibatçıları üçün veb saytı tərtib edərkən və kodlaşdırarkən istifadəlik, əlçatanlıq və cavabdehlik kimi amilləri nəzərə almaları vacibdir.

Ümumilikdə, frontend proqramlaşdırma veb saytın inkişafının vacib hissəsidir, çünki veb saytın istifadəçilərə necə göründüyünü və hiss etdiyini müəyyən edir və istifadəçi təcrübəsinə və veb saytın uğuruna böyük təsir göstərə bilər.

2.2. İntellektual sistem üçün nəzərdə tutulmuş verilənlər bazasının əsas strukturu və idarəedilməsinin əsasları

Verilənlər bazaları müasir veb-saytların vacib komponentidir, çünki onlar tərtibatçılara veb-saytı gücləndirmək üçün istifadə olunan məlumatları saxlamağa və idarə etməyə imkan verir. Veb sayt yaratarkən istifadə edilə bilən bir neçə növ verilənlər bazası var, hər birinin öz güclü və zəif tərəfləri var. Veb saytın inkişafı üçün ən çox istifadə olunan verilənlər bazalarından bəziləri bunlardır:

Əlaqəli verilənlər bazaları: Bunlar verilənləri cədvəllər şəklində təşkil etmək üçün əlaqəli modeldən istifadə edən verilənlər bazalarıdır, hər bir cədvəl müəyyən bir məlumat növünü təmsil edir. Relational verilənlər bazaları verilənlər bazasından məlumatları manipulyasiya etmək və əldə etmək üçün SQL (Strukturlaşdırılmış Sorğu Dili) istifadə edir. Veb inkişafında istifadə olunan populyar relyasiya verilənlər bazalarına MySQL, PostgreSQL və Oracle daxildir.

NoSQL verilənlər bazaları: Bunlar məlumatları təşkil etmək üçün sabit sxemdən istifadə etməyən verilənlər bazalarıdır və bunun əvəzinə məlumatları açar-dəyər cütləri və ya sənədlər kimi çevik və genişlənə bilən formatda saxlayır. NoSQL

verilənlər bazaları böyük həcmdə strukturlaşdırılmamış məlumatları idarə etmək üçün nəzərdə tutulmuşdur və məzmun idarəetmə sistemləri, sosial şəbəkələr və e-ticarət platformaları da daxil olmaqla geniş proqramlar üçün istifadə edilə bilər. Veb inkişafında istifadə edilən məşhur NoSQL verilənlər bazalarına MongoDB, Couchbase və Cassandra daxildir.

Obyekt yönümlü verilənlər bazaları: Bunlar verilənləri təşkil etmək üçün obyekt yönümlü modeldən istifadə edən verilənlər bazalarıdır, hər bir obyekt sistemdə müəyyən bir obyekti və ya obyekti təmsil edir. Obyekt yönümlü verilənlər bazaları multimedia və veb proqramlar kimi mürəkkəb məlumat strukturları tələb edən proqramlar üçün çox uyğundur. Veb inkişafında istifadə olunan məşhur obyekt yönümlü verilənlər bazalarına db4o, ObjectStore və Versant daxildir.

Verilənlər bazaları istifadəçi məlumatı, məzmun və metadata daxil olmaqla müxtəlif növ məlumatların saxlanması və idarə edilməsi üçün veb inkişafında istifadə olunur. Veb-sayt yaradarkən tərtibatçılar verilənlər bazası ilə qarşılıqlı əlaqədə olmaq və müxtəlif tapşırıqları yerinə yetirmək üçün verilənlər bazası idarəetmə sistemlərindən (DBMS) istifadə edirlər, məsələn, cədvəllər və sahələr yaratmaq, verilənlərin daxil edilməsi və yenilənməsi, məlumat üçün verilənlər bazasını sorğulamaq.

Verilənlər bazaları həmçinin axtarış funksionallığı, məzmunun idarə edilməsi və e-ticarət əməliyyatları kimi veb-saytda müxtəlif funksiyaları və funksionallığı gücləndirmək üçün istifadə olunur. Məsələn, e-ticarət veb-saytı müştəri məlumatlarını, məhsul kataloqlarını və sifariş tarixçələrini saxlamaq üçün əlaqəli verilənlər bazasından istifadə edə bilər, sosial şəbəkə isə istifadəçi profillərini, yazılarını və media məzmununu saxlamaq üçün NoSQL verilənlər bazasından istifadə edə bilər.

Verilənlərin saxlanması və idarə edilməsi ilə yanaşı, verilənlər bazası veb saytın performansını və miqyasını yaxşılaşdırmağa kömək edə bilər. Keşləmə və indeksləşdirmə üsullarından istifadə etməklə tərtibatçılar verilənlər bazası sorğularını optimallaşdırırlar və serverdəki yükü azalda bilərlər, bu da veb-sayta daha çox trafik idarə etməyə və daha yaxşı istifadəçi təcrübəsi təmin etməyə imkan verir.

Ümumiyyətlə, verilənlər bazaları müasir veb-saytların kritik tərkib hissəsidir və tərtibatçılara məlumatları genişləndirə bilən və səmərəli şəkildə saxlamağa və idarə etməyə imkan verir. Tərtibatçılar öz xüsusi ehtiyacları üçün düzgün verilənlər bazasından istifadə etməklə sürətli, etibarlı və bütün ziyarətçilər üçün qüsursuz istifadəçi təcrübəsi təmin edən veb-saytlar yarada bilərlər.

Praktiki ixtisas biliklərinin yoxlanılması üçün nəzərdə tutulmuş verilənlər bazası əlaqəli verilənlər bazası – SQL sorğu dilidir.

2.3. Yaradılmış intellektual sistemin təhlükəsizliyinin təmin edilməsi üçün görülmüş işlər

Veb saytın təhlükəsizliyinin təmin edilməsi üçün bir neçə vacib addım atılmışdır. İnternet üzərindən məlumat ötürülməsini qorumaq üçün istifadə edilən bir neçə növ təhlükəsiz şifrələmə protokolları mövcuddur. Bəzi nümunələrə baxaq:

SSL/TLS: SSL (Secure Sockets Layer) və onun varisi TLS (Nəqliyyat Layeri Təhlükəsizliyi) veb-trafikin təhlükəsizliyini təmin etmək üçün geniş istifadə olunan şifrələmə protokollarıdır. Onlar veb server və istifadəçinin veb brauzeri arasında

ötürülən məlumatları qorumaq üçün simmetrik və asimmetrik şifrələmənin birləşməsindən istifadə edirlər.

AES: Qabaqcıl Şifrələmə Standartı (AES) istirahət və tranzit zamanı məlumatların qorunması üçün geniş şəkildə istifadə edilən simmetrik şifrələmə alqoritmidir. SSL/TLS, VPN və digər təhlükəsiz rabitə protokollarında geniş istifadə olunur.

RSA: RSA, SSL/TLS və digər təhlükəsiz rabitə protokollarında açar mübadiləsi üçün istifadə edilən asimmetrik şifrələmə alqoritmidir. Açarı düz mətnlə ötürmədən təhlükəsiz açar mübadiləsini təmin edir.

SHA: Təhlükəsiz Hash Alqoritmi (SHA) məlumatların bütövlüyü və autentifikasiyası üçün geniş istifadə olunan hash funksiyaları ailəsidir. O, adətən SSL/TLS, VPN-lər və digər təhlükəsiz rabitə protokollarında istifadə olunur.

PGP: Olduqca Yaxşı Məxfilik (PGP) e-poçt rabitəsi üçün məşhur şifrələmə üsuludur. Mesajın məxfiliyini, bütövlüyünü və həqiqiliyini təmin etmək üçün simmetrik və asimmetrik şifrələmənin birləşməsindən istifadə edir.

Bunlar internet üzərindən məlumat ötürülməsini qorumaq üçün istifadə edilən şifrələmə protokollarının yalnız bir neçə nümunəsidir. Müxtəlif məqsədlər üçün və sözügedən proqramın və ya sistemin xüsusi təhlükəsizlik tələblərindən asılı olaraq müxtəlif protokollar istifadə olunur.

Yaradılmış intellektual sistemin təhlükəsizliyinin təmin edilməsi üçün HTTPS şifrələmə protokolundan istifadə olunmuşdur.

HTTPS şifrələməsi veb sayt və istifadəçinin veb brauzeri arasında ötürülən məlumatları şifrələmək üçün istifadə edilən təhlükəsizlik protokoludur. HTTPS

"Hypertext Transfer Protocol Secure" deməkdir. İstifadəçi HTTPS şifrələməsindən istifadə edən veb sayta daxil olduqda, onun brauzeri SSL/TLS (Secure Sockets Layer/Nəqliyyat Layeri Təhlükəsizliyi) adlı prosesdən istifadə edərək veb saytın serveri ilə təhlükəsiz əlaqə qurur. Bu şifrələmə, giriş məlumatları və ya kredit kartı məlumatı kimi hər hansı həssas məlumatın icazəsiz şəxslər tərəfindən ələ keçirilməsindən qorunmasını təmin edir. HTTPS şifrələməsinin istifadəsi getdikcə daha çox əhəmiyyət kəsb edir, çünki internet üzərindən daha çox həssas məlumatlar ötürülür.

HTTP və HTTPS internet üzərindən məlumat ötürmək üçün istifadə edilən hər iki protokoldur, lakin aralarında mühüm fərq var.

HTTP (Hypertext Transfer Protocol) veb server və istifadəçinin veb brauzeri arasında məlumat ötürmək üçün istifadə edilən ənənəvi protokoldur. O, məlumatları düz mətnlə göndərir, yəni üçüncü şəxslər tərəfindən tutula və oxuna bilər. Bu, giriş məlumatlarını, kredit kartı detalları və şəxsi məlumatlar kimi həssas məlumatların ötürülməsini təhlükəsiz edir.

HTTP və HTTPS arasındakı əsas fərq internet üzərindən ötürülən məlumatları qorumaq üçün şifrələmənin istifadəsidir. HTTPS əlavə təhlükəsizlik səviyyəsi təmin edir ki, bu da təcavüzkarların internet üzərindən ötürülən məlumatları ələ keçirməsini və oxumasını çətinləşdirir. Buna görə veb saytların həssas istifadəçi məlumatlarını qorumaq üçün HTTPS şifrələməsindən istifadə etmələri tövsiyə olunur.

Əgər yaradılmış intellectual sistem üçün təhlükəsizlik təmin olunmasaydı, gözlə görünə biləcək qədər xəbərdarlıq siqnalları ilə qarşılaşa bilərik. Aşağıda verilmiş

Şəkil 2.1 kimi giriş olunan veb saytın təhlükəsizliyi təmin etmədiyi barədə məlumatverici xəbərdarlıqla qarşılaşa bilərik.



Your connection is not private

Attackers might be trying to steal your information from **aslanovnurane.com** (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_DATE_INVALID



To get Chrome's highest level of security, [turn on enhanced protection](#)

Advanced

Back to safety

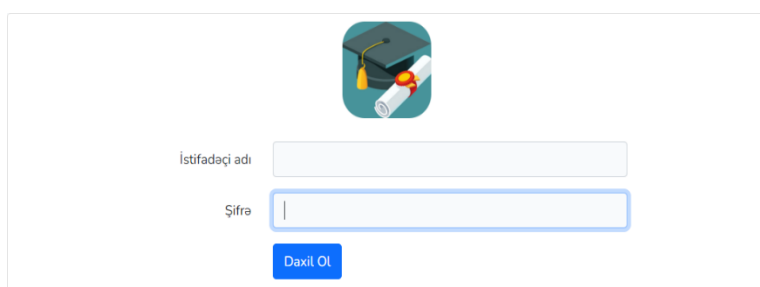
Şək. 2.3.1

Eyni zamanda veb saytın kodlaşdırılması zaman təhlükəsiz kodlaşdırma təcrübələrindən istifadə edilmişdir. Güclü autentifikasiya və giriş nəzarəti vasitəsilə yalnız səlahiyyətli istifadəçilərin veb sayta və onun resurslarına daxil ola bilməsini

təmin etmək üçün güclü autentifikasiya və giriş nəzarət mexanizmlərini tətbiq edilmişdir. Atılan bu addımlar veb saytın istənilən təhlükəsizlik hücumlarından qorumaq üçün kifayət qədər potensiala malikdir.

2.4. Yaradılmış intellektual sistemin nəzərdə tutulan auditoriyanın ehtiyaclarına nə dərəcədə cavab verir ?

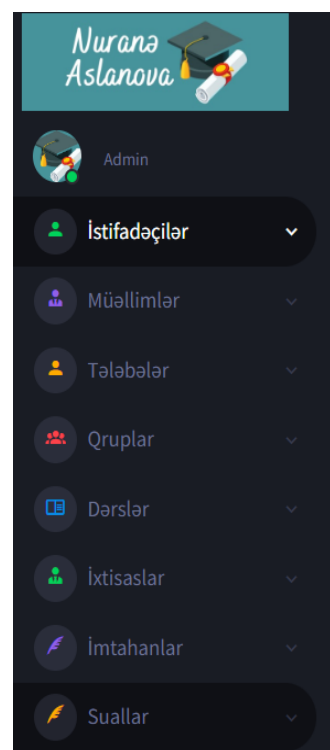
Yaradılmış veb sayt üç səviyyəli girişlə hazırlanmışdır: tələbə, müəllim və admin.



Şək. 2.4.2

istifadəçinin veb-sayta giriş səviyyəsini müəyyən edən (Şək. 2.4.2) unikal istifadəçi adı və parol təyin edilmişdir. Hansı ki, tələbə və müəllimin giriş məlumatlarında dəyişikliklər sadəcə admin tərəfindən edilə bilər. Buna giriş üçün nəzərdə tutulan istifadəçi adı və parolun redaktə edilməsi daxildir. Admin (Şək. 2.4.3) tərəfdə müəllimlər, tələbələr, qruplar, dərslər, ixtisaslar, imtahanlar bölməsi

Hər
bir



Şək. 2.4.3

görünür. Müəllim tərəfdə (Şək. 2.4.4) isə imtahanlar və imtahan suallarının əlavə edilməsi üçün nəzərdə tutulan bölmələr görünür.

Admin istər tələbəni, istərsə də müəllimi aktiv və passiv vəziyyətdə saxlaya bilər. Aktiv vəziyyətdə olan tələbə və ya müəllim istənilən zaman istifadəçi adı və uyğun parolu qeyd edərək, şəxsi hesabına giriş imkanı əldə edə bilər. Lakin admin tərəfindən passiv vəziyyətə salınmış müəllim və ya tələbə istifadəçi adı və parolunu düzgün yazdığı halda belə

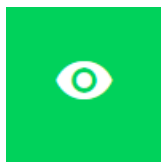


Şək. 2.4.4

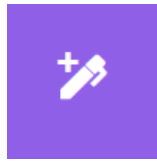
öz hesabına giriş edə bilməz. Admin sistemə yeni

bir tələbə əlavə edərkən tələbə nömrəsi, istifadəçi adı, ixtisasın adı, qrup nömrəsi, ad, soyad, ata adı və doğum tarixi daxil etməlidir. Bu zaman əlavə edilən ixtisas adı, menyu hissəsində yerləşən İxtisaslar bölməsində olmalıdır. Eyni zamanda daxil edilən qrup nömrəsi menyu hissəsində yerləşən Qruplar bölməsində olmalıdır. Admin sistemə müəllim əlavə edərkən istifadəçi adı, ad, soyad, ata adı və doğum tarixi daxil etməlidir. Eyni zamanda tələbələr admin tərəfindən sadəcə öz qruplarına əlavə edilə bilər. Hər hansı digər istifadəçi tərəfindən bu dəyişdirilə və ya redaktə edilə bilməz. Admin (Şək. 2.4.3) ən yüksək səviyyədə giriş əldə edəcək və istifadəçi hesablarını, kurs materiallarını və digər inzibati tapşırıqları idarə edə biləcəklər. Eyni adlı tələbə bir neçə qrupda iştirak edə bilməz. Menyu hissəsində yerləşən İstifadəçilər bölməsində həm müəllimlərin həm də tələbənin məlumatları görünür. Menyu hissəsindəki Müəllimlər bölməsində sadəcə müəllimlərin məlumatları, Tələbələr bölməsində isə sadəcə tələbələrin məlumatları görünür. Tələbələr öz kurs

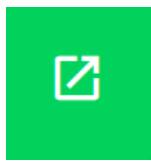
materiallarına, tapşırıqlarına və qiymətlərinə, müəllimlər isə tapşırıqları idarə edə və qiymət verə, tələbələrlə əlaqə saxlaya və kurs materiallarına daxil ola biləcəklər.



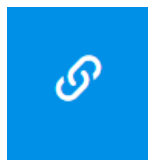
Şək. 2.4.6



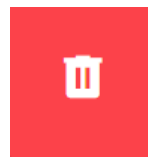
Şək. 2.4.7



Şək. 2.4.8


















Şək. 2.4.9



Şək. 2.4.10

Admin
girişi
zamanda
menyu

hissəsində görünən İmtahanlar bölməsinə yeni imtahan daxil edilərkən, imtahan üçün nəzərdə tutulan mövzu Başlıq olaraq qeyd edilməli, əlavə olaraq imtahanı keçirən müəllimin adı, müəllimin istifadəçi adı, imtahan fənninin adı, sual sayı və yaradılma tarixi qeyd edilməlidir. Bundan əlavə imtahanın aktiv və ya passiv vəziyyəti yaradılmışdır. Hansı ki, imtahan admin tərəfindən passiv vəziyyətə keçirildiyi halda (Şək. 2.4.5) bu imtahanda tələbələrin iştirakı mümkün olmayacaq. İmtahanlar bölməsində Statistika buttonuna (Şək. 2.4.6) klik etməklə yaradılmış imtahanda iştirak edən və etməyən tələbələri asanlıqla görmək mümkündür. Eyni

Qrup Nömrəsi	Status	Sual Sayı	Yaradılma Tarixi	
1317a	Aktivdir	4	2023-04-24 16:01:42	    
1317a	Aktivdir	1	2023-04-24 16:48:33	    
1317a	Aktivdir	2	2023-04-26 12:50:30	    

Şək. 2.4.5

zamanda iştirak etmiş tələbənin düzgün və səhv cavablarının sayını və imtahan suallarına verdiyi cavabları görmək mümkündür. İmtahanlar bölməsində Yenilə butonuna (Şək. 2.4.7) klik etməklə

admin tərəfindən imtahan məlumatları redaktə edilə bilər. İmtahanlar bölməsində Göstər butonuna (Şək. 2.4.8) butonuna klik edən zaman brauzerdə yeni səhifə açılır və bu zaman imtahan suallarını ekranda görmüş oluruq. Nüsxələ butonu (Şək. 2.4.9)

vasitəsi ilə admin imtahan linkini asanlıqla nüsxələyə bilər və imtahanda iştirakı nəzərdə tutulmuş tələbələr link vasitəsilə imtahana daxil ola bilər. Sil butonu (Şək. 2.4.10) vasitəsilə admin tərəfindən imtahan asanlıqla silinə bilər.

Admin menyusunda (Şək. 2.4.3) yerləşən Suallar bölməsində sualın hansı imtahana aid olması ilə bağlı imtahan başlığı və sualın şərti qeyd olunur. Sual seçimləri bölməsində isə hər sual başlığı üçün beş seçim yerləşdirilir. Daha sonra seçimlər Yanlış və Doğru seçimlər olaraq qeyd edilir. Tələbənin seçdiyi cavab admin və ya müəllim tərəfindən Yanlış olaraq qeydiyyata alındığı zaman tələbənin cavabı yanlış hesab edilir. Doğru cavab seçildiyi halda bu proses eyni ilə davam edir.

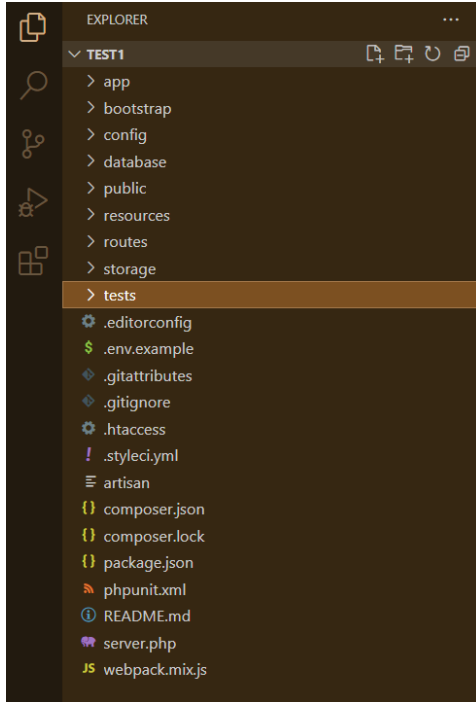
Bu çoxsəviyyəli giriş sistemi hər bir istifadəçinin ehtiyac duyduğu məlumat və alətlərə çıxışını təmin edir, eyni zamanda veb sayt məlumatlarının təhlükəsizliyini və bütövlüyünü qoruyur.

FƏSİL 3

YARADILMIŞ İNTELLEKTUAL SİSTEMİN KOD NÜMUNƏLƏRİ

3.1 Proyektin ümumi strukturu

Proyektin ümumi strukturu (Şək. 3.1.1) doqquz kataloqdan ibarət olub, bu bölmələrə "app", "bootstrap", "config", "database", "public", "resources", "routes",



Şək. 3.1.1

"storage", "tests" daxildir. "App" fayl içərisində əsas kodlar yerləşir, hansı ki, bu kodlar məntiqi kodlar və ya main kodlar adlanır. Veb layihəsinin ümumi strukturunda "app" faylı adətən veb-sayt daxilində xüsusi proqram və ya modulun kodunu ehtiva edən fayl və ya qovluğa istinad edir. PHP-dən istifadə edən veb layihəsində "app" adətən veb-sayt daxilində xüsusi modul və ya proqramın kodunu ehtiva edən qovluğa və ya kataloqa istinad edir. Bu qovluq verilənlər bazası əlaqələri, marşrutlaşdırma və kontrollerlər kimi backend funksionallığı və ya istifadəçi interfeysi

komponentləri və JavaScript davranışı kimi frontend funksionallığı üçün PHP fayllarını ehtiva edə bilər. PHP-dən istifadə edən veb layihəsində "bootstrap" adətən veb tətbiqini işə salan və ya "bootstrap" edən kodu ehtiva edən fayl və ya qovluğa istinad edir. Bu kod sabitlərin müəyyən edilməsi, avtomatik yükləyicinin konfigurasiyası, xidmət təminatçılarının qeydiyyatı və verilənlər bazası əlaqələrinin qurulması daxil olmaqla, mühitin qurulmasına cavabdehdir.

"Bootstrap" faylının və ya qovluğunun xüsusi strukturu və məzmunu PHP veb inkişafında istifadə olunan çərçivədən asılı olaraq dəyişə bilər. Tərtib olunan veb layihəsində Laravel PHP çərçivəsində "bootstrap" qovluğunda "app.php" və "cache" daxil olmaqla proqramın yüklənməsi üçün bir neçə fayl var. Bu fayllar xidmət təminatçıları, fəsadlar və konfigurasiya kimi Laravel tətbiqinin müxtəlif

aspektlərinin konfigurasiyasına cavabdehdir.

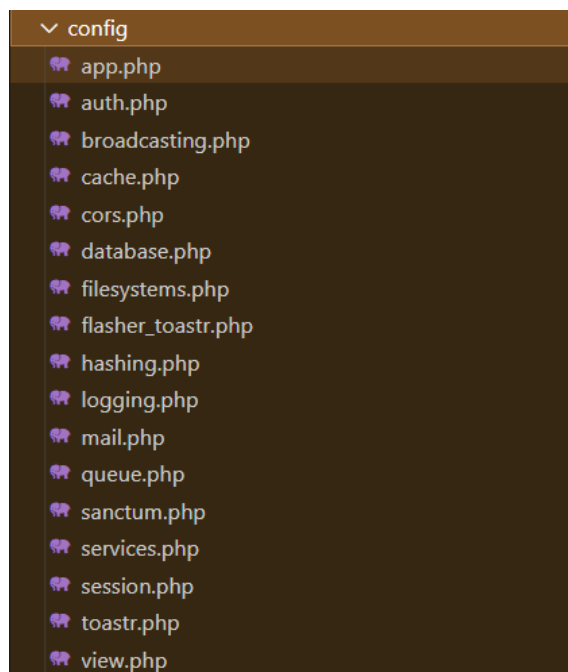
Ümumiyyətlə, "bootstrap" qovluğu və ya faylı PHP veb layihəsinin ümumi strukturunun vacib komponentidir. Onun kodu tətbiqi işə salır, mühiti qurur və proqramın müştərilərdən daxil olan sorğuları idarə etməyə hazır olmasını təmin edir.

PHP istifadə edən veb layihəsində "config" adətən veb tətbiqi üçün konfigurasiya faylları olan qovluğa istinad edir. Bu fayllar verilənlər bazası parametrləri, təhlükəsizlik parametrləri və giriş parametrləri kimi proqramın davranışını müəyyən edən parametrləri və seçimləri saxlayır.

"config" qovluğunun xüsusi strukturu və məzmunu PHP veb inkişafında istifadə olunan çərçivədən asılı olaraq dəyişə bilər. Məsələn, Laravel PHP çərçivəsində "config" kataloqunda proqramın müxtəlif aspektləri üçün konfigurasiya parametrlərini müəyyən edən müxtəlif PHP faylları var. "Config" qovluğunda aşağıdakı PHP faylları(Şək. 3.1.2) əlavə edilmişdir:

- 1) "app.php" : Bu fayl proqram adı, mühit və saat qurşağı kimi proqram üçün ümumi konfigurasiya parametrlərini özündə birləşdirir.
- 2) "database.php" : Bu fayl verilənlər bazası bağlantısı üçün verilənlər bazası növü, host, port, istifadəçi adı və parol kimi konfigurasiya parametrlərini ehtiva edir.
- 3) "cache.php" : Bu fayl proqramda keşləmə üçün konfigurasiya parametrlərini ehtiva edir, məsələn, keş sürücüsü, keş prefiksi və keşin bitmə vaxtları.
- 4) "session.php" : Bu fayl proqramda sessiyanın idarə edilməsi üçün konfigurasiya parametrlərini ehtiva edir, məsələn, sessiya ömrü və sessiya şifrələmə parametrləri.
- 5) "mail.php" : Bu fayl e-poçt sürücüsü, SMTP(Simple Mail Transfer Protocol)

server təfərrüatları və e-poçt şifrələmə parametrləri kimi proqramda e-poçt göndərmək üçün konfigurasiya parametrlərini ehtiva edir.



Şək. 3.1.2

6) "filesystem.php" : Bu fayl fayl uzantısı , standart fayl sistemi və fayl sistemi diskləri kimi proqramda fayl saxlanması üçün konfigurasiya parametrlərini ehtiva edir.

7) "auth.php" : Bu fayl tətbiqdə istifadəçinin autentifikasiyası və avtorizasiyası üçün konfigurasiya parametrlərini ehtiva edir, məsələn, standart autentifikasiya qoruyucusu və istifadəçi provayderi.

8)"broadcasting.php" : Bu fayl standart yayımçı, yayım əlaqələri və hadisə dinləyiciləri kimi proqramda hadisələrin

yayımlanması üçün konfigurasiya parametrlərini ehtiva edir.

9) "logging.php" : Bu fayl proqrama daxil olmaq üçün konfigurasiya parametrlərini ehtiva edir, məsələn, login səviyyələri və login kanalları.

10) "queue.php" : Bu fayl proqramdakı növbələr üçün standart növbə sürücüsü, əlaqə parametrləri və işə yenidən cəhdlər kimi konfigurasiya parametrlərini ehtiva edir.

11) "services.php" : Bu fayl sosial media API-ləri və e-poçt xidmətləri kimi proqramda istifadə olunan üçüncü tərəf xidmətləri üçün konfigurasiya parametrlərini ehtiva edir.

12) "view.php" : Bu fayl tətbiqin görünüş qatı üçün konfigurasiya parametrlərini ehtiva edir, məsələn, standart görünüş mühərriki, görüntü saxlama yolları

13) "cors.php" : Bu fayl müxtəlif domenlərdən olan veb səhifələrin bir-birinə sorğu göndərməsinə imkan verən Çarpaz Mənşəli Resurs Paylaşımı (CORS) üçün

konfigurasiya parametrlərini ehtiva edir. Bu fayldakı parametrlər hansı mənşələrin veb proqrama daxil olmasına icazə verildiyini və hansı növ sorğulara icazə verildiyini müəyyən edir.

14) "flasher_toastr.php" : Bu faylda Toastr kitabxanasından istifadə edərək veb proqramında flaş mesajların göstərilməsi üçün konfigurasiya parametrləri var. Bu fayldakı parametrlər mesaj növü, mövqeyi və animasiyası kimi flaş mesajların görünüşünü və davranışını idarə edir.

15) "hashing.php" : Bu fayl istifadəçi parollarını təhlükəsiz saxlamaq üçün istifadə edilən veb proqramda parolun heşinqi üçün konfigurasiya parametrlərini ehtiva edir. Bu fayldakı parametrlər istifadə ediləcək hashing alqoritmini, həmçinin təkrarların sayını və uzunluğunu müəyyən edir.

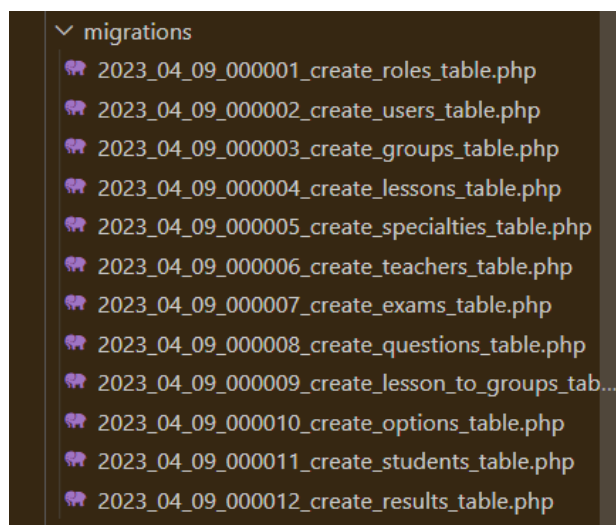
16) "sanctum.php" : Bu fayl veb proqramda API-lərin autentifikasiyasının sadə yolunu təmin edən Laravel Sanctum paketi üçün konfigurasiya parametrlərini ehtiva edir. Bu fayldakı parametrlər istifadə ediləcək autentifikasiya sürücüsünü, həmçinin istifadəçilərin autentifikasiyası üçün qoruyucuları və provayderləri müəyyən edir.

17) "toastr.php" : Bu faylda Toastr kitabxanasından istifadə edərək veb proqramda bildirişlərin göstərilməsi üçün konfigurasiya parametrləri var. Bu fayldakı parametrlər mesaj növü, mövqeyi və animasiyası kimi bildirişlərin görünüşünü və davranışını idarə edir.

Ümumiyyətlə, "config" qovluğu və ya kataloqu PHP veb layihəsinin ümumi strukturunun mühüm komponentidir. Onun faylları bütün proqramda istifadə olunan parametrləri ehtiva edir və o, tərtibatçılara proqramın əsas kodunu dəyişdirmədən proqramın davranışını asanlıqla dəyişdirməyə və fərdiləşdirməyə imkan verir.

PHP veb layihəsində "database" qovluğunda verilənlər bazası konfigurasiyası və

idarə olunması ilə bağlı PHP fayllar yerləşir. Bu qovluqda yerləşən ümumi fayllar və onların məqsədləri bunlardır:



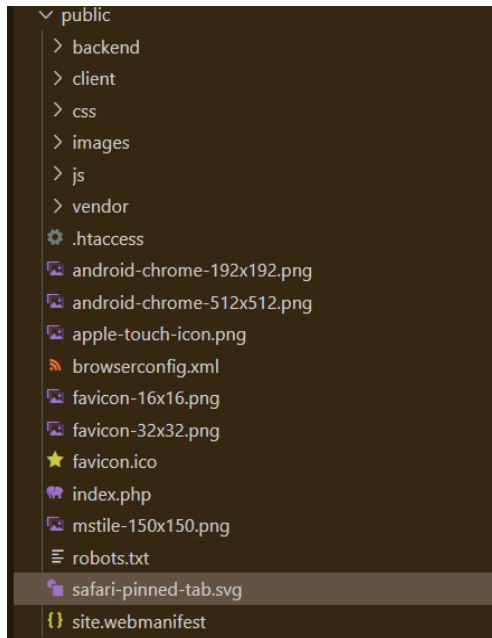
Şək. 3.1.3

1) "migrations" : Bu qovluqda (Şək. 3.1.3) verilənlər bazası miqrasiyaları üçün PHP faylları var ki, bunlar zamanla verilənlər bazası sxemini dəyişdirən skriptlərdir. Hər bir miqrasiya faylı adətən verilənlər bazası cədvəllərini, indeksləri və ya digər obyektləri yaradan və ya dəyişdirən bir sıra SQL əmrlərini özündə birləşdirir. Miqrasiya verilənlər bazası sxeminə versiya nəzarət etmək və

zamanla dəyişiklikləri idarə etməyi asanlaşdırmaq üçün istifadə olunur.

2) "seeds" : Bu kataloq verilənlər bazası ilkin hissə üçün PHP fayllarını ehtiva edir, bunlar verilənlər bazasına nümunə məlumatları daxil edən skriptlərdir. Seeders sınaq və ya inkişaf məqsədləri üçün verilənlər bazasını ilkin məlumatlarla doldurmaq üçün istifadə edilə bilər.

3) "factories" : Bu kataloq sınaq məqsədləri üçün saxta məlumatlar yaratmaq üçün istifadə edilən model fabriklər üçün PHP fayllarını ehtiva edir. "Factories" avtomatlaşdırılmış sınaq üçün nümunə məlumatları yaratmaq və ya verilənlər bazasını böyük həcmdə məlumatlarla doldurmaq üçün istifadə edilə bilər.



Şək. 3.1.4

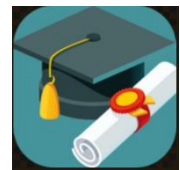
PHP veb layihəsində “public” kataloq(Şək. 3.1.4) tətbiqin ictimai əsas kataloquna istinad edir. Bu kataloq veb server vasitəsilə ictimaiyyətə açıq olan faylları ehtiva edir. "Public" kataloqda tapılan bəzi ümumi fayl və qovluqlar və onların məqsədləri bunlardır:

1) "index.php" : Bu fayl proqram üçün giriş nöqtəsidir. İstifadəçi veb sayta daxil olduqda, veb server bu faylı yükləyir və istifadəçinin brauzerinə geri göndərilən HTML yaratmaq üçün PHP kodunu icra edir.

2) ".htaccess" : Bu fayl URL yenidən yazma qaydaları və keşləmə direktivləri kimi Apache server konfigurasiya parametrlərini ehtiva edir. O, veb serverin proqram üçün sorğuları necə idarə etdiyini konfigurasiya etmək üçün istifadə edilə bilər.

3) "vendor/": Bu kataloq üçüncü tərəf kitabxanalarını və tətbiqin tələb etdiyi asılılıqları ehtiva edir. Bu fayllar ictimaiyyət üçün əlçatan deyil, lakin proqram tərəfindən pərdə arxasında istifadə olunur.

4) "favicon.ico" : Bu fayl istifadəçinin brauzer nişanında və ya əlfəcin çubuğunda göstərilən ikonadır. Bu, veb saytın markasına uyğunlaşdırıla bilən kiçik bir şəkil faylıdır. Yaradılmış veb layihəsində aşağıdakı favicon(Şək. 3.1.5) yerləşdirilmişdir.



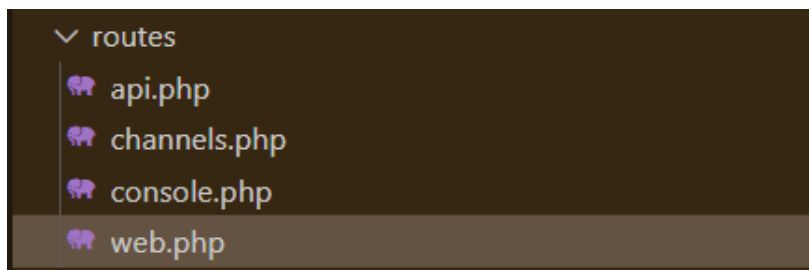
Şək. 3.1.5

PHP veb layihəsində "resources" kataloqu görünüşlər, dil faylları kimi tətbiq tərəfindən istifadə edilən ictimai olmayan resursları ehtiva edən qovluğa istinad edir. "Resources" kataloqunda tapılan ümumi fayl və qovluqlara və onların məqsədlərinə aşağıdakılar daxildir:

views/: Bu kataloq tətbiqin məzmununu yaratmaq üçün istifadə olunan HTML şablonlarını ehtiva edir. Bu şablonlar dinamik HTML yazmağı asanlaşdırmaq üçün adətən Blade və ya Twig kimi şablon dilini istifadə edir.

lang/: Bu kataloq proqramın müxtəlif dillər və yerli dillər üçün lokallaşdırılması üçün istifadə edilən dil fayllarını ehtiva edir. Bu fayllar adətən tərcümə edilmiş sətirlərə uyğunlaşan açar-dəyər cütlərini ehtiva edir.

PHP veb layihəsində "routes" kataloqu (Şək. 3.1.6) proqram üçün marşrut təriflərini



Şək. 3.1.6

ehtiva edən qovluğa istinad edir. Bu marşrutlar tətbiqin istifadəçinin brauzerindən və ya digər müştəridən daxil olan sorğuları necə idarə etməli olduğunu müəyyən –

ləşdirir.

"Routes" kataloqundakı bəzi ümumi fayl və qovluqlara və onların məqsədlərinə aşağıdakılar daxildir:

1) "web.php" : Bu fayl HTTP GET və POST sorğuları kimi veb sorğular üçün

marşrutları özündə birləşdirir. Bu marşrutlar adətən tətbiqin istifadəçi interfeysinə uyğundur və səhifələrin göstərilməsi, formaların təqdim edilməsi və istifadəçinin autentifikasiyasının idarə edilməsi kimi işləri idarə edir.

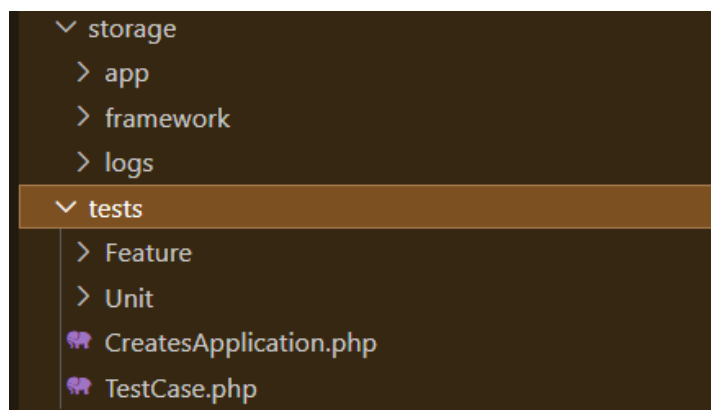
2) "api.php" : Bu fayl HTTP GET, POST, PUT və DELETE sorğuları kimi API sorğuları üçün marşrutları özündə birləşdirir. Bu marşrutlar adətən API son nöqtələrinə uyğundur və məlumatların əldə edilməsi, resursların yaradılması, resursların yenilənməsi və silinməsi kimi işləri idarə edir.

3) "channels.php" : Bu fayl WebSocket əlaqələri kimi real vaxt rabitə kanalları üçün marşrutları ehtiva edir. Bu marşrutlar real vaxt söhbəti, bildirişlər və digər interaktiv funksiyalar kimi funksiyaları həyata keçirmək üçün istifadə edilir.

4) "console.php" : Bu faylda konsol əmrləri üçün marşrutlar var və bunlar komanda xəttində tapşırıqları yerinə yetirmək üçün istifadə olunur. Bu marşrutlar adətən verilənlər bazası köçürmələri, planlaşdırılmış tapşırıqlar və digər sistemə texniki xidmət tapşırıqları kimi tapşırıqları yerinə yetirir.

“Routes” kataloqundakı xüsusi fayllar və qovluqlar istifadə olunan çərçivədən və layihənin tələblərindən asılı olaraq dəyişə bilər.

PHP veb layihəsində “storage” kataloqu (Şək. 3.1.7) proqram tərəfindən yaradılan və ya yüklənmiş faylları, həmçinin proqram tərəfindən istifadə olunan keş faylları, sessiya faylları və digər müvəqqəti faylları ehtiva edən qovluğa istinad edir.



Şək. 3.1.7

"Storage" kataloqunda tapılan bəzi ümumi fayl və qovluqlar və onların məqsədləri bunlardır:

- 1) "app/" : Bu qovluqda qeydlər, keşlənmiş görünüşlər və digər proqrama aid məlumatlar kimi proqram tərəfindən yaradılan fayllar var.
- 2) "framework/" : Bu kataloq keş faylları, sessiya faylları və framework tərəfindən istifadə edilən digər müvəqqəti fayllar kimi framework tərəfindən yaradılan faylları ehtiva edir.
- 3) "logs/" : Bu kataloq proqram və ya framework tərəfindən yaradılan, sazlama və ya audit məqsədləri üçün istifadə edilə bilən jurnal fayllarını ehtiva edir.
- 4) "public/" : Bu kataloq yüklənmiş şəkillər, sənədlər və digər media faylları kimi internetdən ictimaiyyət üçün açıq olan fayllardan ibarətdir.

PHP veb layihəsində "tests" kataloqu adətən proqram üçün vahid testləri və inteqrasiya testlərini ehtiva edən qovluğa istinad edir. Bu testlər tətbiqin düzgün

işləməsini və onun tələblərinə cavab verməsini təmin etmək üçün istifadə olunur.

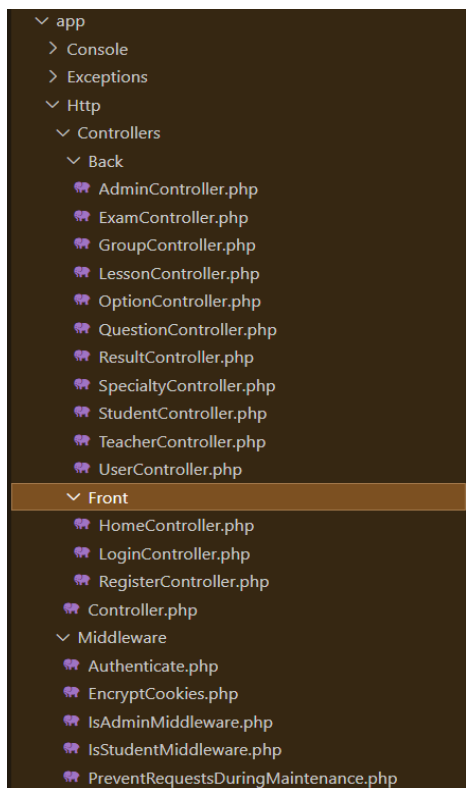
"Tests" kataloqunda (Şək. 3.1.7) tapılan bəzi ümumi fayl və qovluqlara və onların məqsədlərinə aşağıdakılar daxildir:

- 1) "Feature/" : Bu kataloqda tətbiqin davranışını istifadəçi nöqtəyi-nəzərindən yoxlayan xüsusiyyət testləri var. Bu testlər adətən formanı doldurmaq və ya düyməyə basmaq kimi istifadəçi hərəkətlərini simulyasiya edir və tətbiqin düzgün davrandığını yoxlayır.
- 2) "Unit/" : Bu kataloq proqramın ayrı-ayrı komponentlərini təcrid olunmuş şəkildə sınaqdan keçirən vahid testlərini ehtiva edir. Bu testlər adətən tətbiqin funksiyalarını, siniflərini və ya digər modullarını sınaqdan keçirir və onların müxtəlif şərtlər altında düzgün davrandığını yoxlayır.
- 3) "TestCase.php" : Bu fayl bütün digər testlər üçün başlanğıc nöqtəsi kimi istifadə edilən əsas test nümunəsi sinfini ehtiva edir. Bu sinif müxtəlif testlərdə istifadə olunan ümumi funksionallıq və yardım proqramlarını təmin edir.
- 4) "CreatesApplication.php" : Bu fayl test üçün tətbiq nümunəsini yaratmaq üçün istifadə olunan xüsusiyyəti ehtiva edir. Bu xüsusiyyət tətbiqi yükləmək və ümumi quraşdırma tapşırıqlarını yerinə yetirmək üçün sadə bir yol təqdim edir.

3.2. Veb layihəsinin ümumi strukturunda "app" kataloq faylları

PHP veb layihəsində "app" kataloqu (Şək. 3.2.1) adətən tətbiqi təşkil edən əsas faylları və qovluqları ehtiva edən qovluğa istinad edir. Bu fayllar və qovluqlar proqramın məntiqini idarə etmək, sorğu və cavabları idarə etmək və proqram

məlumatlarını idarə etmək üçün məsuliyyət daşıyır. "App" kataloqunda tapılan bəzi ümumi fayl və qovluqlar və onların məqsədləri bunlardır:



1) "Http/" : Bu qovluq HTTP sorğuları və cavablarını idarə etmək üçün məsul olan fayllardan ibarətdir. Buraya nəzarətçilər, ara proqram və proqramın biznes məntiqini idarə edən digər siniflər daxildir. Bu qovluq daxilində "Controllers" və "Middleware" kataloq faylları yerləşdirilmişdir. "Middleware" kataloq faylında yerləşən php uzantılı fayllar Laravel framework yükləyən zaman proyektə daxil olunur. Lakin "IsAdminMiddleware.php" və "IsStudentMiddleware.php" adlı php faylları tərtibatçı tərəfindən yazılmışdır.

Şək. 3.2.1

"IsAdminMiddleware.php" php faylında kod strukturu (Şək. 3.2.2) aşağıdakı kimi işləyir. if şərt funksiyası daxilində auth check olunur. Yəni ilk öncə sayta giriş olub-

```

17     public function handle(Request $request, Closure $next)
18     {
19         if (!auth()->check()) {
20             return redirect('/login');
21         }
22         if (auth()->user()->role_id==3) {
23             return redirect('/');
24         }
25         return $next($request);
26     }

```

Şək. 3.2.2

olmadığı yoxlanılır. Əgər sayta giriş olmayıbsa bu zaman sayt istifadəçini avtomatik

olaraq login səhifəsinə yönləndirir. Əks halda, yəni sayta giriş olunubsa və giriş üçün `role_id = 3` olarsa istifadəçini admin üçün tərtib olunmuş istifadəçi interfeysinə yönləndirilir. `Role_id=3` qeyd olunması isə belə şərh edilir: Database-də 3 istifadəçi yaddaşa yazılıb - admin, müəllim, tələbə. Hər dəfə müəyyən istifadəçini çağırmaq üçün istifadəçinin `role_id` qiymətindən istifadə olunur. Hansı ki, biz database-də admin üçün `role_id` olaraq 3 daxil etmişik.

"IsAdminMiddleware.php" adlı php faylının yaradılma səbəbi isə admin adlı istifadəçi sayta giriş etməyibsə admin interfeysinə daxil olmağı məhdudlaşdırılsın. Yəni admin interfeysinə daxil olmaq üçün qeydiyyatdan keçməmiş – istifadəçi adı və parol yazmamış daxil olma ehtimalını aradan qaldırmaq üçün bu kod strukturundan istifadə olunur. Hər hansı link vasitəsilə sayta daxil olmağa çalışan dələduzların qarşısının alınması səbəbi ilə tərtib olunmuş kod strukturudur.

```

18 public function handle(Request $request, Closure $next)
19 {
20     if (!auth()->check()) {
21         return redirect('/login');
22     }
23     if (auth()->user()->role_id!=3) {
24         return redirect('/admin');
25     }
26     if(auth()->user()->role_id==3 && auth()->user()->getStudent==null){
27         auth()->logout();
28         return redirect('/login')->with('error',"Sizin hesabınızda tələbə mövcud deyil");
29     }
30     if(auth()->user()->role_id==3 && !auth()->user()->status){
31         auth()->logout();
32         return redirect('/login')->with('error',"sizin girişiniz deaktiv edilib.Admin ilə əlaqə saxlayın");
33     }
34     return $next($request);
35 }
36 }
37

```

Şək. 3.2.3

"Controllers" kataloq faylının içərisində "UserController.php", "AdminController.php", "ExamController.php", "StudentController.php", "TeacherController.php", "ResultController.php", "QuestionController.php", "OptionController.php", "LessonController.php" və "GroupController.php" adlı php faylları yazılmışdır. Hansı ki, hər bir php uzantılı fayl adından məlum olduğu kimi müəyyən qrupa aid əməliyyatları özündə birləşdirir. Məsələn, "UserController.php" faylında sadəcə istifadəçiyə aid ola bilən əməliyyatlar – istifadəçi əlavə edilməsi, silinməsi, aktiv və ya passiv statusa keçirilməsi və sair əməliyyatlar bura daxildir.

2) "Provayderlər/" : Bu kataloq verilənlər bazası əlaqələri, hadisə dinləyiciləri və keşləmə kimi müxtəlif proqram xidmətlərinin qeydiyyatı və yüklənməsinə cavabdeh olan xidmət təminatçılarına ehtiva edir.

3) "Console/" : Bu qovluqda əmr xətti sorğularının idarə edilməsi üçün məsul olan fayllar var. Buraya verilənlər bazası köçürmələri və ya kod yaratmaq kimi xüsusi tapşırıqları yerinə yetirən siniflər olan əmrlər daxildir.

4) "Exception/" : Bu kataloq proqramda baş verən istisnaların idarə edilməsinə cavabdeh olan fayllardan ibarətdir. Buraya xüsusi istisna növlərini idarə edən və müvafiq cavablar yaradan siniflər olan istisna işləyiciləri daxildir.

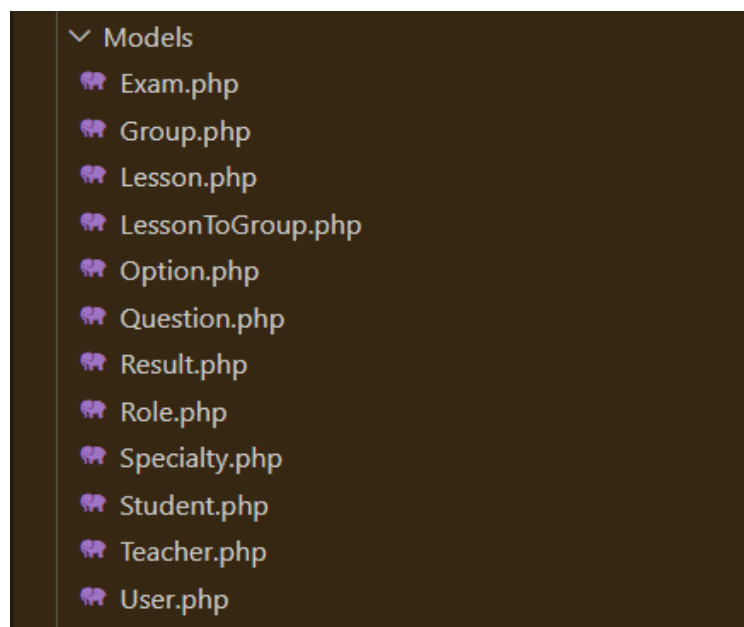
5) "Models/" : Bu kataloq proqramın məlumat modellərini təyin edən fayllardan ibarətdir. O, verilənlər bazası cədvəllərini və ya digər məlumat mənbələrini təmsil edən və məlumatların əldə edilməsi, yaradılması və yenilənməsi üsullarını təmin edən sinifləri əhatə edir.

"Tətbiq" kataloqundakı xüsusi fayllar və qovluqlar istifadə olunan çərçivədən və layihənin tələblərindən asılı olaraq dəyişə bilər. Bununla belə, bunlar PHP veb layihəsində bu kataloqda tapıla bilən ümumi fayl və qovluqların bəzi nümunələridir.

3.3. Veb layihəsinin ümumi strukturunda "models" kataloq faylları

Veb saytın layihə strukturunda olan "models" kataloqu (Şək. 3.3.1) məlumat modelləri və verilənlər bazası əməliyyatları ilə əlaqəli faylları və ya modulları ehtiva edir. Bu, Model-View-Controller (MVC) arxitekturasını izləyən veb inkişaf çərçivələrində tapılır.

"Models" kataloqunda yerləşən hissələr bunlardır:



Şək. 3.3.1

Məlumat Modelləri: Bu kataloq veb saytda istifadə edilən məlumat obyektlərinin və ya obyektlərinin strukturunu və davranışını müəyyən edən fayllardan ibarət ola bilər. Bu modellər məlumat sxemini təmsil edir və məlumatların manipulyasiyası və təsdiqlənməsi ilə bağlı atributları, əlaqələri və metodları ehtiva edə

bilər.

Verilənlər Bazasının Konfigurasiyası: "Models" kataloquna verilənlər bazası ilə əlaqənin konfigurasiyasına cavabdeh olan fayllar daxildir. Bu fayllar verilənlər bazası etimadnaməsini, əlaqə parametrlərini və seçilmiş verilənlər bazası sistemi ilə əlaqə yaratmaq üçün lazım olan digər parametrləri müəyyənləşdirir.

Verilənlər Bazası Əməliyyatları: Bu kataloq verilənlər bazası əməliyyatlarını idarə edən faylları və ya modulları ehtiva edir, məsələn, məlumat qeydlərinin yaradılması, oxunması, yenilənməsi və silinməsi (CRUD). Bu fayllar adətən verilənlər bazası ilə qarşılıqlı əlaqədə olan, SQL sorgularını yerinə yetirən və ya verilənlər bazası əməliyyatlarını yerinə yetirmək üçün Obyekt-Relational Xəritəçəkmə (ORM) çərçivəsindən istifadə edən funksiyaları və ya metodları ehtiva edir.

Model Əlaqələri: Veb saytın məlumat modellərinin əlaqələri varsa (məsələn, bir-bir, bir-çox, çox-çox), "modellər" kataloquna bu əlaqələri müəyyən edən və idarə edən fayl və ya modullar daxil ola bilər. Bu, müxtəlif məlumat obyektləri arasında assosiasiyalar yaratmağa kömək edir və əlaqəli məlumatların səmərəli sorğulanmasına və manipulyasiyasına imkan verir.

Validatorlar və Məlumatların Manipulyasiyası: "Models" kataloqu verilənlər bazasından saxlanmazdan və ya götürülməzdən əvvəl istifadəçi daxiletməsinin təsdiqlənməsi və ya verilənlərin manipulyasiyası üçün cavabdeh olan validatorları və ya funksiyaları müəyyən edən fayllardan ibarətdir. Bu fayllar verilənlərin bütövlüyünü təmin edir, məhdudiyyətləri tətbiq edir və biznes məntiqinin tələb etdiyi kimi məlumat transformasiyalarını idarə edir.

Qeyd etmək vacibdir ki, "models" kataloqunda yerləşən kodların əsas mahiyyəti

SQL kodları ilə uzun-uzun yazmaq əvəzinə, cədvəllər arasındakı əlaqəni daha sadə kodlarla yazmaqdır. Bir sözlə "models" database-də yerləşən cədvəllərin class-ları olaraq başa düşülür. Yaradılmış olan proyektə (Şək. 3.3.1) yerləşən cədvəllər isə "Exam", "Group", "Lesson", "LessonToGroup", "Option", "Question", "Result", "Role", "Specialty", "Student", "Teacher", "User" adlandırılmışdır.

```
class Exam extends Model
{
    use HasFactory;
    protected $guarded=[];
    public function user()
    {
        return $this->hasOne(User::class,'id','user_id')->withDefault();
    }
    public function teacher()
    {
        return $this->hasOne(Teacher::class,'id','teacher_id')->withDefault();
    }
    public function lesson()
    {
        return $this->hasOne(Lesson::class,'id','lesson_id')->withDefault();
    }
    public function group()
    {
        return $this->hasOne(Group::class,'id','group_id')->withDefault();
    }
    public function questions()
    {
        return $this->hasMany(Question::class,'exam_id','id')->withDefault();
    }
}
```

Şək. 3.3.2

Models kataloq faylında yerləşən Exam adlı php faylından kod nümunəsi (Şək. 3.3.2) verilmişdir. Müəyyən funksiyalar qeyd olunur, hansı ki, bu funksiyalar vasitəsilə əlaqələr göstərilir. hasOne və ya hasMany olaraq qeyd olunan əlaqələr aşağıdakı kod nümunələrində daha aydın qeyd olunmuşdur:

```
public function user()
{
    return $this->hasOne(User::class, 'id', 'user_id')->withDefault();
}
```

Bir istifadəçinin sadəcə bir imtahanı ola bilər. Və ya bir müəllimin, bir dərsin, bir qrupun bir imtahanı ola bilər.

```
public function questions()
{
    return $this->hasMany(Question::class, 'exam_id', 'id')->withDefault();
}
```

Lakin bu kod nümunəsində hasMany əlaqəsi qeyd olunduğu üçün deyə bilərik ki, bir imtahanın çoxlu sualları ola bilər.

NƏTİCƏ

Bu tədqiqat kollec tələbələrinin praktiki bacarıqlarını qiymətləndirmək üçün hazırlanmış onlayn sınaq platformasının hazırlanması və tətbiqini təqdim edir. Platforma üç giriş rolu təklif edir: Admin, tələbə və müəllim. Müəllimlər testlər yarada və dərc edə, tələbələr isə qeydiyyatdan keçib bu testlərdən keçə bilirlər. Bu tədqiqatın məqsədi tələbələrin praktiki bacarıqlarının qiymətləndirilməsində onlayn sınaq platformasının effektivliyini və onların təlim nəticələrinə təsirini qiymətləndirməkdən ibarət olub.

- Testin hazırlanması və idarə edilməsi:
- Tələbə İşi və İstifadəçi Təcrübəsi:
- Qiymətləndirmənin Dəqiqliyi və Etibarlılığı:
- Təlim Nəticələri və Fəaliyyət Təhlili:

Bu tədqiqatın nəticələri kollec tələbələrinin praktiki bacarıqlarının qiymətləndirilməsində onlayn test platformasının effektivliyini nümayiş etdirir. Platforma istifadəçi dostu təcrübə təklif edir, tələbələrin iştirakını təşviq edir, qiymətləndirmənin dəqiqliyini və etibarlılığını təmin edir və dərin performans təhlilini asanlaşdırır. Praktiki bacarıqların qiymətləndirilməsi üçün texnologiyadan istifadə etməklə, təhsil müəssisələri öz qiymətləndirmə üsullarını təkmilləşdirə və tələbələrə dəyərli rəylər təqdim edə, nəticədə onların ümumi təlim nəticələrinə və bacarıqlarının inkişafına töhfə verə bilirlər.

İSTİFADƏ EDİLMİŞ ƏDƏBİYYAT

1. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins (Publication Year: 2018)
2. "HTML and CSS: Design and Build Websites" by Jon Duckett (Publication Year: 2011)
3. "Responsive Web Design with HTML5 and CSS3" by Ben Frain (Publication Year: 2019)
4. "CSS: The Missing Manual" by David Sawyer McFarland (Publication Year: 2015)
5. "JavaScript and JQuery: Interactive Front-End Web Development" by Jon Duckett (Publication Year: 2014)
6. "HTML and CSS: Design and Build Websites" by Jon Duckett (Publication Year: 2011)
7. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics" by Jennifer Niederst Robbins (Publication Year: 2018)
8. "HTML5 and CSS3 All-in-One For Dummies" by Andy Harris (Publication Year: 2014)
9. "HTML, CSS, and JavaScript Mobile Development For Dummies" by William Harrel (Publication Year: 2011)
10. "PHP and MySQL Web Development" by Luke Welling and Laura Thomson (Publication Year: 2016)
11. "Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5" by Robin Nixon (Publication Year: 2014)
12. "PHP Objects, Patterns, and Practice" by Matt Zandstra (Publication Year: 2013)

13. "Modern PHP: New Features and Good Practices" by Josh Lockhart (Publication Year: 2015)
14. "PHP Cookbook" by David Sklar and Adam Trachtenberg (Publication Year: 2014)
15. "Laravel: Up and Running" by Matt Stauffer (Publication Year: 2016)
16. "Laravel: Code Bright" by Dayle Rees (Publication Year: 2013)
17. "Learning Laravel 5: Building Practical Applications" by Antonio Ribeiro (Publication Year: 2015)
18. "CSS: The Definitive Guide" by Eric Meyer and Estelle Weyl (Publication Year: 2017)
19. "CSS Secrets: Better Solutions to Everyday Web Design Problems" by Lea Verou (Publication Year: 2015)
20. "CSS Pocket Reference: Visual Presentation for the Web" by Eric A. Meyer (Publication Year: 2018)
21. "HTML and CSS: Design and Build Websites" by Jon Duckett (Publication Year: 2011)
22. "SQL in 10 Minutes a Day" by Ben Forta (Publication Year: 2012)
23. "SQL Pocket Guide" by Jonathan Gennick (Publication Year: 2010)
24. "SQL Queries for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL" by John L. Viescas and Michael J. Hernandez (Publication Year: 2018)
25. "Assessing Student Learning: A Common Sense Guide" by Linda Suskie (Publication Year: 2018)

