**MUNDARIJA.**

**[KIRISH 2](#_Toc9116)**

**[I.Tizimli tahlil 4](#_Toc742)**

[1.1 Web sahifalarni yaratish usul va texnologiyalari 4](#_Toc11523)

[1.2 Masalaning qo‘yilishi 7](#_Toc16748)

**[II.Asosiy qism. 9](#_Toc19428)**

[2.1 Web ilova yaratish uchun dasturlash muhitini sozlash 9](#_Toc5525)

[2.2 PHP haqida ma’lumot. 10](#_Toc31909)

[2.3 Foydalanuvchi uchun yo‘riqnoma 14](#_Toc15543)

**[Xulosa 17](#_Toc6060)**

**[Men quyidagilarni o’rgandim: 17](#_Toc1152)**

**[Foydalanilgan adabiyotlar 18](#_Toc5983)**

**[Ilova 18](#_Toc7011)**

KIRISH

Hozirda Kompyuter va axborot texnologiyalari jadal sur’atlar bilan yangilanib, rivojlanishi bilan birga kundalik turmushimizning asosiga aylanib bormoqda.

Hozirgi kunda zamоnaviy talablarni hisоbga оlgan hоlda ma`lumоtlar tarmоg’ini tashkil qilishni takоmillashtirish, o`sib kelayotgan yosh avlоdning intellektual ehtiyojlarini qоndirishga, madaniy, ma`naviy-axlоqiy qadriyatlarni saqlab qоlishga yo`naltirilgan printsipial yangi axbоrоt markazlarini barpо etish, shuningdek, ahоlini yanada kengrоq va tizimli axbоrоt bilam ta`minlash uchun zarur shart-sharоitlar yaratish maqsadida O`zbekistоn Respublikasida ta`lim sоhasida quyidagilarni amalga оshirish ko`zda tutilgan:

* ta`lim muassasalari o`quvchilari hamda axоlining zamоnaviy axbоrоt texnоlоgiyalaridan fоydalangan hоlda muntazam ta`lim оlishi va mustaqil ravishda ta`lim оlishiga ko`maklashish;
* yangi axbоrоt texnоlоgiyalari
* asоsida ahоliga axbоrоt xizmati ko`rsatish;
* madaniy, ta`lim, axbоrоt hamda bоshqa dastur va lоyihalarni birgalikda amalga оshirish uchun ta`lim muassasalari, mahalliy o`zini o`zi bоshqarish о`rganlari, milliy madaniyat markazlari bilan xamkоrlikni rivоjlantirish.
* Respublikamizda ta`lim tizimida tub islоhatlar оlib bоrilmоqda. Shulardan yana biri masоfadan turib o`qitish sistemasining jоriy qilinishi. Bunda Respublikamiz оliygоhlarida malakali prоfessоrlarning ma`ruzalarini tinglash, bevоsita muloqat o`rnatish mumkin.
* Masоfaviy ta`lim sistemasining ishlab chiqilishi keyingi yillarda ta`lim xizmatiga bo`lgan talabning keskin o`sishi tufayli hamda uning ijtimоiy ahamiyatiga asоslangan hоlda sоdir bo`ladi. O`zbekistоnda yagоna masоfaviy ta`lim sistemasining tashkil etilishi va faоliyat ko`rsatishi Respublikamizda zarur bo`lgan kadrlar, pedagоgik, ilmiy texnik va ilmiy metоdik pоtentsiallarning mavjudligi tufayli bo`lsa kerak. Bundan tashqari masоfaviy ta`limning yangi texnоlоgiyalari haqida ko`plab tajriba yig`ilgan va ko`plab ishlar amalga оshirilgan.
* Ta`lim sоhasida elektrоn darslik va qo`llanmalarni yaratish uchun quyidagi muxim masalalarni hal qilish kerak bo`ladi.
* Internet texnоlоgiyasi bilan tanishib chiqish ;
* prоgramma vоsitalarini o`rganish, masalan Web -saytlarni yaratish va qayta ishlash bo`yicha;
* Web -saxifalarda har-xil infоrmatsiyalarni tasvirlash uchun usullar va imkоniyatlarni o`rganish va qo`llash;
* Web -saytlarni yaratish va qayta ishlashga dоir tavsiyalar va asоsiy qоidalar bilan tanishib chiqish;
* Web -saxifani strukturasini aniqlash;
* Web -saytni qayta ishlash va yaratish uchun maqsadni tanlash;
* WEB saxifalarini o`qish vоsitalari .

HTML va bоshqa prоgramma vоsitalari yordamida tayyorlangan Web saxifalarida fоydalanuvchiga tushunarli ko`rinishda tasvirlash uchun maxsus prоgrammalar ishlab chiqilgan bo`lib, bunday prоgrammalar brauzer prоgrammalar deb ataladi. Hоzirda bir necha shunday prоgrammalar ishlab chiqilgan bo`lib, ular tabiiy ravishda hujjatlarni ko`rishni turlicha tahrir qiladilar. Bular оrasida keng tarqalgan Firefox , Chrome va Safari prоgrammalaridir. Shunday qilib, brauzerning asоsiy vazifasi URL adreslarda jоylashgan Web saxifalarini kоmpyuterga yuklash va uni fоydalanuvchiga tushunarli ko`rinishda mоnitоr ekranida ko`rsatib berishdir.

# I.Tizimli tahlil

## 1.1 Web sahifalarni yaratish usul va texnologiyalari

Hozirgi axborot texnologiyalar rivojlanib borayotgan davrda butun jahon tarmog'idagi veb-sahifalar kundan kunga ko'payib bormoqda. Kimdir o'z shaxsiy sahifasini yaratsa, kompaniya va firmalar korporativ saytlarini yaratmoqda. Yana kimdir o'z kundaliklarini, ya'ni bloglarini olib boradilar.

Shu bilan birga, yirik portallar faoliyati ham rivojlanmoqda. Bularning hammasi esa biror dasturiy vosita yordamida yaratiladi. Kimdir HTML va CSS tilida, kimdir PHP da, yana kimlardir CMS deb ataluvchi tizimlar yordamida turli xil veb-saytlarni yaratmoqda.

1989-yilda Tim Berners-Li tomonidan World Wide Web, ya'ni Bununjahon o'rgimchak to'ri ixtiro qilinganidan boshlab uning har bir foydalanuvchisi oddiy bo'lsa-da, o'z veb-sahifasini yarata olish imkoniyatiga ega bo'ldi. Lekin, buning uchun HTML (Hyper Text Markup Language) tilini bilish shart edi. Turli xil IT-kompaniyalari tomonidan bu ishni osonlashtiruvchi bir qancha dasturiy paketlar, jumladan Adobe Dreamweaver va Microsoft FrontPage kabilar yaratildi. Bunday dasturlardan ba'zilari sayt yaratishning barcha bosqichlarini boshqarishni o'z zimmasiga oladi. Shu bilan birga, ko'pgina hollarda foydalanuvchi bu kodni bilishi shart bo'lmagan. Sababi bu dasturlarda veb-sahifalarni yaratishning vizual usullaridan foydalanilgan.

Biroq, HTML-redaktorlarining barcha afzalliklariga qaramay, saytni boshqarish vaqt o'tgani sayin va yangi ma'lumotlar qo'shilishi bilan qiyinlashib boraverdi. Chunki axborot texnologiyalari rivojlanishi bilan veb-saytlar ham rivojlanib bordi. Sayt xaritasini yaratishdek oddiy funksiyani qo'shish uchun ham veb-yaratuvchilar tomonidan ancha-muncha mehnat talab etilar edi. Axir buning uchun barcha ilovalarni ko'rib chiqish, sahifalar orasidagi bog'liqliklarni tekshirish, menyuni yaratish, forumni qo'shish kabi ishlarni bajarish lozim edi. Bu esa doimiy diqqatni va juda ko'p vaqtni talab etardi.

Sayt yangilanishi muammosini, ya'ni yangilanishning avtomatlashtirishini hal qilish maqsadida yirik axborot kompaniyalari, jumladan, Newsweek va Time yangi dasturiy ta'minot yaratish ishlarini boshlashga majbur bo'ladi. Bu dasturiy ta'minotlar kontentni (ma'lumotlarni) boshqarish tizimlari (CMS – Content Management System) deb ataldi.

**1. Dizayn yaratish.**

Hozirda dizayn yo'nalishi ham juda ko'p qismlarga bo'lingan. Veb dizayner, Banner dizaynerlari, Logotip dizayneri, naqsh va emblema dizaynerlari va hokazo.... Bularni hammasini chuqur bilish shart emas. Asosiysi, Adobe Photoshop dasturini bilish kerak, chuqur bilmasa ham oddiy dizyanlar yoki interfeyslar chiza olishingiz kerak, Bundan tashqari, Adobe Fireworks dasturi ham interfeyslar chizish uchun juda qulay

**2. HTML CSS**

Sayt yoki web dasturning dizayni tayyor bo'lgach uni browserda ko'rinishi uchun html va css kodlari orqali ko'rsatish kerak

**3. Dasturlash tillari va malumotlar bazasi.**

Hozirda dasturlash tillari ham ko'p, Eng mashxurlari, PHP, ASP, Python, Lekin juda ko'p saytlar PHP tilida qilingan. php haqida juda ko'p darsliklar bor. Dasturlash tillari bilan birgalikda ma’lumotlar bazasi bilan ishlanadi,asosan ularga MySQL va PostgreSQLlar kiradi.Ularda saytrning asosiy matnlar bazasi saqlanadi

**4. Browser dasturlash tili – JavaScript**

JavaScript tilini o'rganish ham muhim, u yaratayotgan dasturimizning dizayni bilan ish ko‘radi , JavaScript uchun tayyor frameworklar bor. Framework - bu oddiy qilib aytganda avvaldan qilingan tayyor mayda dasturlar yoki effektlardir. Ularga JQuery, Mootools va boshqalar kiradi va u bizning ishimizni ancha osonlashtiradi.

**5. CMS –Tayyor boshqaruv tizimlari.**

CMS (Content Management System) bu tayyor qilingan sayt, Dasturlash va dizaynni bilmaydigan odam ham CMS yordamida chiroyli saytlar qilishi mumkin. Kod va dizaynlari ham tayyor qilingan bo'ladi. Siz uni faqat internetga joylash va ichini to'ldirishingiz kerak xolos. Hozirda mashxur CMS dasturlari bu Joomla, Wordpress, Drupal, hisoblanadi. Bundan tashqari PHP Framework dasturlari ham bor. Framework bu tayyor yozilgan kodlar. Masalan malumotlarni bazadan olish va uni ko'rsatish uchun PHP dasturinida bir necha qator kod yozish kerak, bu kod yozishda ham ancha bosh qotirasiz. Framework esa hamma kodlarni tayyor yozib qo'ygan, siz ularni bir biriga ulaysiz xolos.

**6. Hosting va Domen**

Hosting bu saytingizni internetda saqlaydigan server, Domen esa sayt nomi (masalan: www.saytnomi.uz) . Saytni internetga joylash, va hostingni moslash (htaccess fayllari yoki cron bilan ishlash) kabi bilimlar ham dastur ishlashimiz uchun ancha kerakdir.

**7. Saytni e’lon qilish, qidiruv tizimlari bilan hamkorlik**

Bu eng oxirgi qadam. Buni SEO (Search engine Optimization) deb ham aytiladi. Siz qilgan saytingiz yoki proektingizni odamlar bilishlari uchun uni elon qilish kerak. Qidiruv saytlari: google, yandex, mail.ru orqali qidirganda sizni saytingiz ham ko'rinishi kerak. Buni bajarish uchun saytni turli kataloglarga qo'shish kerak, Google yoki yandex qidiruv tizimiga kiritish kerak. Xullas, yuqoridagi bilimlar bu Sayt mutaxassislari (Web-master) uchun kerak bo'lgan umumiy bilimlardir. Agar siz dastulash yoki saytlar tuzish bilan shug'ullanadigan biror korxona rahbari bo'lsangiz, u holda yuqorida sanalgan bilimlarni bilishingiz lozim. Lekin hammasini professional darajada bilish shart emas. Hech bo'lmaganda, qaysi dastur qachon kerak bo'lishi va qanday ishlashi haqida umumiy tushunchaga ega bo'lish kerak. Hozirda, zamon talabi shundayki: bir sohani chuqur o'rganish uchun boshqa sohalardan voz kechishingiz kerak, Masalan. Dasturchi bo'lishga qaror qildingiz. Buning uchun dasturlash tillaridan 3 - 4 tasini o'rganasiz. Keyin ularni chuqur o'rganishga vaqtingiz ham yetmay qoladi. Keyin 1 ta yo'nalishdagi dasturlashni tanlashga majbur bo'lasiz. Aytaylik Web dasturlashni tanlaysiz. Avvalda, vaqtingizni PHP va ASP tillarini o'rganishga bag'ishlaysiz. Keyinchalik yanada chuqurroq o'rganishni davom ettirsangiz faqat bittasini tanlash kerak bo'ladi. Chunki hozirda ilmlar chuqurlashib bormoqda. Bir vaqtning o'zida ko'p bilimlarni o'rganaman degan odamning bilimlari yuzaki qolib ketadi. Yuqorida sanalgan bilimlardan faqat bittasini chuqur o'rganaman desangiz, albatta boshqalaridan voz kechishga majbursiz. Agar hammasiga qiziqsangiz, u holda siz boshqaruvchi bo'la olishingiz mumkin. Professional sayt ochishdan avval bepul yo'l bilan sayt ochib tajriba orttirgan yaxshi. Hozirda internet tarmog'ida ko'plab bepul hostinglar va bepul domen berish xizmatlari bor. Bunday holda saytingiz nomi – domen uchinchi darajali bo'ladi. Masalan:  www.saytnomi.ucoz.com yoki www.saytnomi.wix.com Xullas, saytingiz nomidan keyin qo'shimcha yana nom, keyin esa .com yoki .ru va hokazo turdagi domen bo'lishi mumkin.

## 1.2 Masalaning qo‘yilishi

Hozirda talabalar bitiruv malakaviy ishlarini olish uchun o’qituvchi bilan muloqot qiladi va berilgan mavzu yuzasidan ish boshlaydi. Bizning hozirdagi maqsadimiz shu jarayonlarni avtomatlashtirishdir. O’qutchi mavzularni online tarzda yaratishi va talabalar mavjud mavzulardan o’zi uchun ma’qulini tanlashi mumkin bo’ladi.

**Ushbu kurs ishi esa “BMI Kurs ishi” web ilovasini yaratishga mo'ljallangan bo'lib, saytni tuzish jarayonida quyidagi ishlar amalga oshirilishi lozim:**

- HEMIS tizimidan talaba ma’lumotlarini olish;

- Talaba, O’qituvchi, Mudir, Sifat bo’limi uchun rollar yaratish;

- Talabalar mavzu tanlash imkoniyati;

- O’qituvchilar mavzu yaratish va tanlangan mavzuni foizini belgilash;

- Mudir shu kafedradagi o’qtuvchilar satistikasini ko’rish;

- Sifat bo’limi barcha kafedralar statistikasini ko’rish;

- Super Admin tizimda yangi mudirlar va sifat bo’limi xodimlarini yaratish

# II.Asosiy qism.

## 2.1 Web ilova yaratish uchun dasturlash muhitini sozlash

Web-ilova - mijoz (jumladan foydalanuvchi) veb-brauzerida ishlaydigan mijoz-server dasturidir. Umumiy veb-ilovalar veb-pochtani, onlayn chakana savdoni, onlayn-auksionlarni, vikislarni, tezkor xabar xizmatlarini va boshqa ko'pgina funktsiyalarni o'z ichiga oladi.

Har qanday dinamik veb-sahifa va "veb-ilovalar" o'rtasidagi umumiy farq noaniq. Ko'proq "veb-ilovalar" deb ataladigan veb-saytlar ish stoli dasturiga yoki mobil ilovaga o'xshash funksiyalarga ega bo'lganlardir. HTML5 veb-sahifalar sifatida yuklangan ilovalarni yaratish uchun aniq tilni qo'llab-quvvatladi, lekin ma'lumotlarni mahalliy sifatida saqlashi va oflayn rejimda ishlashga davom etishi mumkin.

**Web sahifa** va **web saytlarga** xizmat ko`rsatish uchun mezbon kompyuterlar va serverlarda ishlatish uchun server dasturiy ta'minoti kerak bo`ladi. Web saytlar va bu saytlarni yaratish, boshqarish uchun mo`ljallangan dasturiy ta'minot mezbon kompyuterda joylashgan bo`ladi, mezbon kompyuterlar bog`lanish protokollarini boshqaradi. Mezbon kompyuterlar Unix, Windows yoki Linux kabi turli operatsion tizim ostida ishlaydilar. Bu operatsion tizimlarning ammasi tarkibiga TCP/IP protokolini qo`llab-quvvatlovchi dasturiy vositalar kiradi. Server dasturiy ta'minoti serverlarda joylashgan bo`lib, klient kompyuteridagi web brauzer talablariga ko`ra so`ralgan web sahifalarga xizmat ko`rsatadi va boshqa amallarni bajaradi. Server dasturiy ta'minoti TCP/IP bog`lanishlar uchun javob bermaydi. Buni serverdagi operatsion tizim bajaradi. Lekin server dasturiy ta'minoti operatsion tizim yordamida HTTP bog`lanishlar va talablarini qo`lga kiritadi va ularni qayta ishlaydi. Server dasturiy ta'minotining har xil turlari, masalan ma'lumotlar ombori serveri, FTP serveri, tarmoq serverlari mavjud. Xususan, web server HTTP serverining bir turi bo`lib, uning vazifasiga klient kompyuterlariga HTTP protokoli yordamida turli ma'lumotlarni jo`natish kiradi. Odatda, klient kompyuteridagi web brauzer serverga HTML xujjatni jo`natish kabi talablarni qo`yadi. Server bunday talablarni olgach, unga javob beradi. Bu javobning boshida uzatish haqida ma'lumot, qolgan qismi HTML fayldan iborat bo`ladi.  
Albatta, web serverlar sahifalarni web brauzerga uzatishdan boshqa ishlarni ham qiladi. U web sahifadagi CGI ssenariylarini mos CGI ilovalariga uzatadi. Bu ssenariylar tashqi mini-dasturlarni ishga tushiradi. Zarurat bo`lganda, bu dasturlar ishining natijalarini web server orqaga web brauzerga qaytaradi. Bulardan tashqari, web serverlar tarkibiga konfiguratsiya fayllari va utilitlar kiradi, ular web saytlarni turli usullarda himoya qilish va boshqarish imkonini beradilar.

## 2.2 PHP haqida ma’lumot.

PHP (Hypertext Preprocessor) — eng ko‘p tarqalgan dasturlash tillaridan biri bo‘lib, Web — dasturlar yaratishda ishlatiladi. Hozirda PHP juda ko‘p hosting- provayderlar tomonidan foydalaniladi, bu esa uni harqanday internet- proyektlarni (oddiygina saytdan tortib, to yirik portallargacha) yaratishda ishlatilatiladigan, sal kam, eng asosiy tilga aylantirdi. PHP da juda ko‘p miqdorda alohida skriptlar, shuningdek, forum, kontent tizimi boshqaruvi kabi tugallangan proyektlar yozilgan. PHP tarixi ... 1994 yili php tilinig yaratuvchisi Rasmus Lerdorf o ’zinig saytiga mehmonlar kirishini hisoblash uchun Perl/ CGI (skriptlar to‘plami) da maxsus qobiq yozib, amalda qo‘lladi va uni «Personal Home Page»(PHP nomi shundan kelib chiqqan) deya nomladi.

**PHP tilida dasturlash asoslari** PHP dasturlash tili tilida tuzilgan har qanday dastur **php** kengaytmali fayllarda saqlanadi.

Misol: uchun **functions.php**, **index.php**, **admin.php** kabi. Birnchi misolni ko'ramaiz. **test.php** nomli fayl tashkil etamiz va unga quyidagi kodlarni yozamiz:

|  |
| --- |
| <html>  <head>  <title>HTML hujjat</title>  <head>  <body>  Hujjat tanasi  </body>  </html> |

Bu matn HTML asosida yozilgan. 1-misoldan ko'rinib turibdiki php hujjat HTML hujjat kabi shakllantirilmoqda. Lekin php hujjatda biz qo'shimcha imkoniyatlarga ega bo'lamiz.**test.php** faylga quyidagi o'zgarishlarni kiritamiz:

|  |
| --- |
| <?php  Echo "<html>";  Echo "<head>";  Echo "<title>HTML hujjat</title>";  Echo "<head>";  Echo "<body>";  Echo "Bu matn HTML asosida yozilgan";  Echo "</body>";  Echo "</html>";  ?> |

 2-misol ham 1-si kabi hujjat hosil qiladi. Farqi shundaki ikkinchi holda HTML teglarni PHP dasturi yordamida hosil qilinmoqda. Lekin ko'p hollarda HTML teglari butunicha PHP yordamida hosil qilinmay ora-oralarda PHP kodlari yoziladi:

|  |
| --- |
| <?php  $title = “HTML hujjat”;  $text = “Bu matn PHP asosida yaratilgan”;  ? >  <html>  <head>  <title><?php echo $title; ?></title>  </head>  <body>  <?php echo $text; ?>  </body>  </html> |

Bu yerda PHP kod HTML teglarni generatsiya qilyapti. Demak PHP dasturining vazifalaridan biri HTML teglarini generatsiya qilish ekanini bilib oldik. PHP ning bundan tashqari yana ko'plab imkoniyatlari mavjud bo'lib unga ma'lumotlar bazasiga murojaat qilish, turli hisob-kitoblarni amalga oshirish kabilar ham kiradi. Lekin PHPdan ko'pchilik aynan klent kompyuterida emas, balki serverda bajarilishi uchun foydalanadi. PHP dasuri tuzish haqida shu vaqtgacha bilib olgnlarimizni umumlashtiramiz:

1) \*.php kengaytmali fayl yaratiladi;

2) faylda istalgancha HTML teglarini ishlatish mumkin;

3) PHP kodlarini ochish va yopish belgilari orasida yoziladi:

a) <?php –ochish belgizi va ?> -yopish belgisi;

b) <? –ochish belgisi va ?> - yopish belgisi;

c) <script language = "php"> -ochish belgisi va </script> - yopish belgisi;

d) <?= -ochish belgisi ba ?> yopish belgisi.

PHP kodlarini yozish bo'yicha yana bitta misol ko'ramiz:

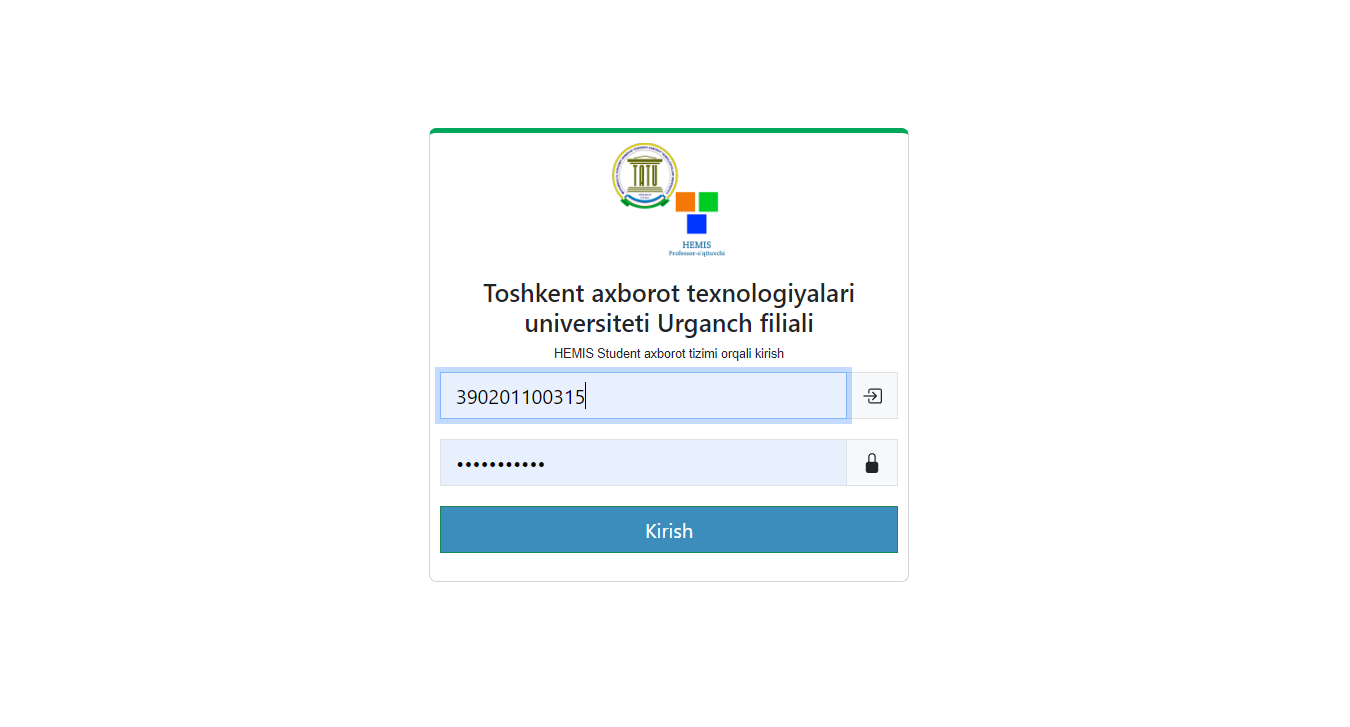
|  |
| --- |
| <?php // birinchi usuldan foydalanilmoqda  $title = “HTML hujjat”;  $ text = “Bu matn PHP asosida yaratilgan”;  ? >  <html>  <head>  < title>  <? // ikkinchi usuldan foydalanilmoqda  eecho $title;  ?>  </title>  </head>  <body>  <script language = “php”> // uchunchi usuldan foydalanilmoqda  Echo $text;  </script>  </body>  </html> |

Universal usul sifatida birinchi usulni aytish mumkin. Serverdagi PHP ning sozlamasiga ko'ra ikkinchi va to'rtinchi usullarni ham qo'llash PHP kodlarini yozishda yengilliklar yaratadi. Uchunchi usulni dasturchilar kamdan-kam foydalanishdi. Bu kurs ishi ya’ni “Murabbiylar kengashi faoliyati” ma`lumotlar monotoringini yuritish web ilovasini yaratishda Laravel framework foydalangan holda yaratiladi. bu tarmoqli web ilovalar qurish uchun yuqori sifatli kompanentalar asosidagi PHP fremework strukturasidi. U katta kodlardan qayta va qayta foydalanishda moslashuvchanlikni va web ilovani tezkorligini oshirishga imkon beradi.Qisqa qilib aytganda oddiy, samarali va keng ko‘lamli PHP framework.

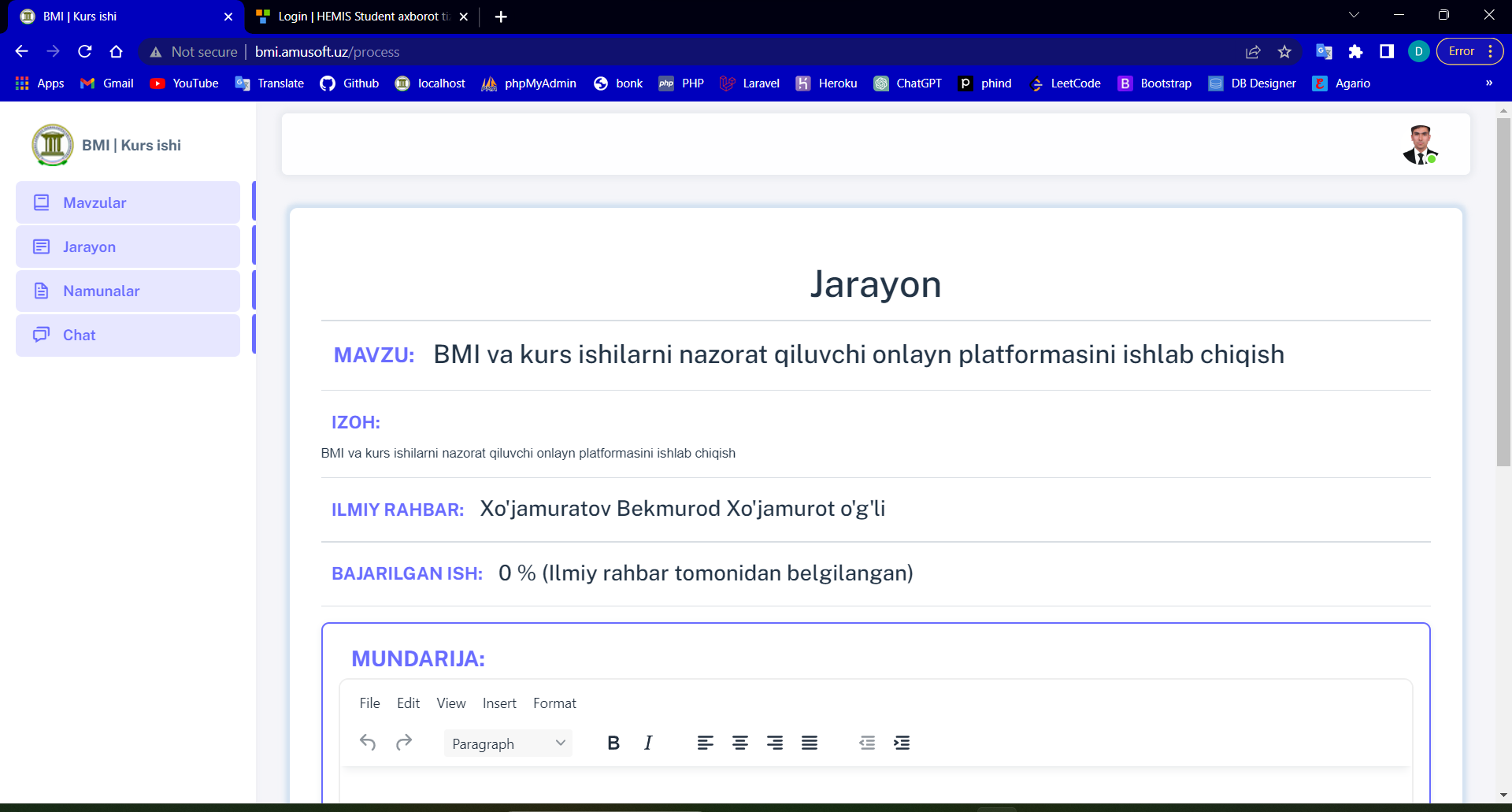
Laravel- universal framework va uni barcha turdagi web ilovalar uchun qo‘llash mumkin. Uning kompanentalar strukturasi va keshlanishi zo‘r qo‘llab quvvatlanishi evaziga framework asosan portlar, forumlar, CMS, magazinlar yoki REST ful ilovalar kabi proektlar uchun katta qo‘l keladi. Bu kurs ishida asosiy algoritmlar Laravelda Controller deb ataluvchi class dagi methodlarda joylashgan bo‘‘ladi

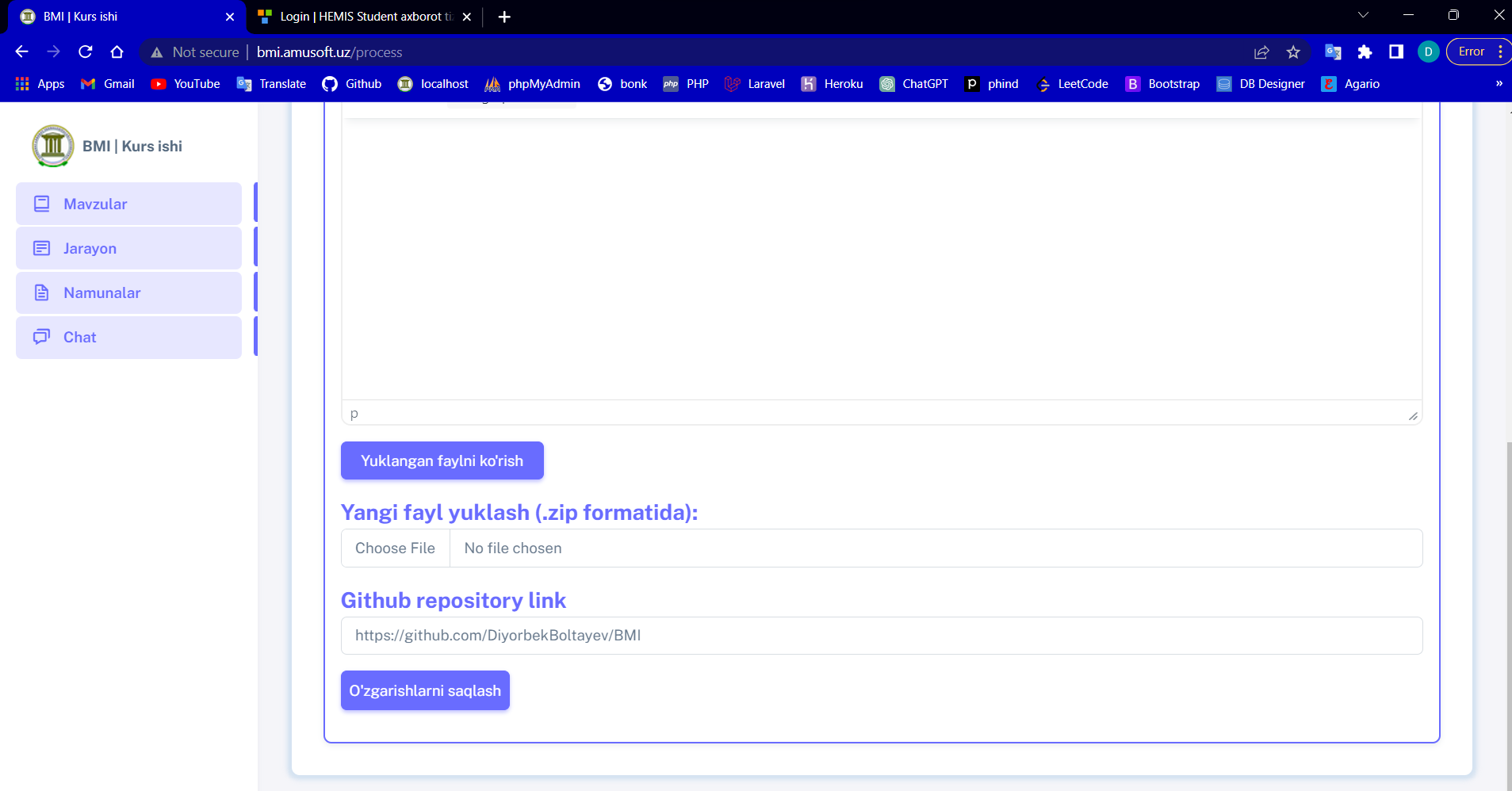
## 2.3 Foydalanuvchi uchun yo‘riqnoma

Dasturdan talaba sifatida foydalanish uchun tizimga kirish lozim

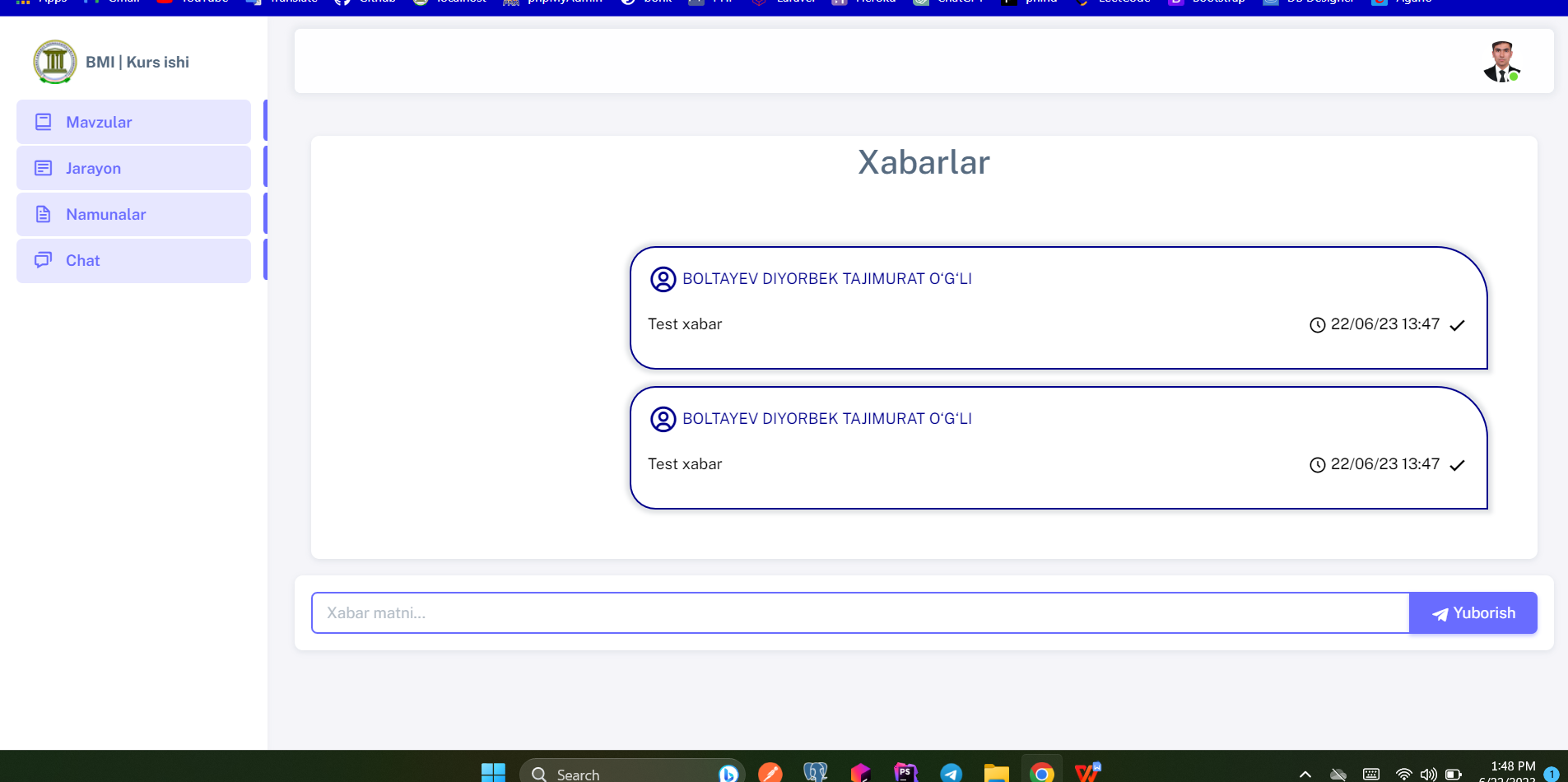


**Unda jarayon bo’limi orqali quyidagilarni bajarish mumkin**

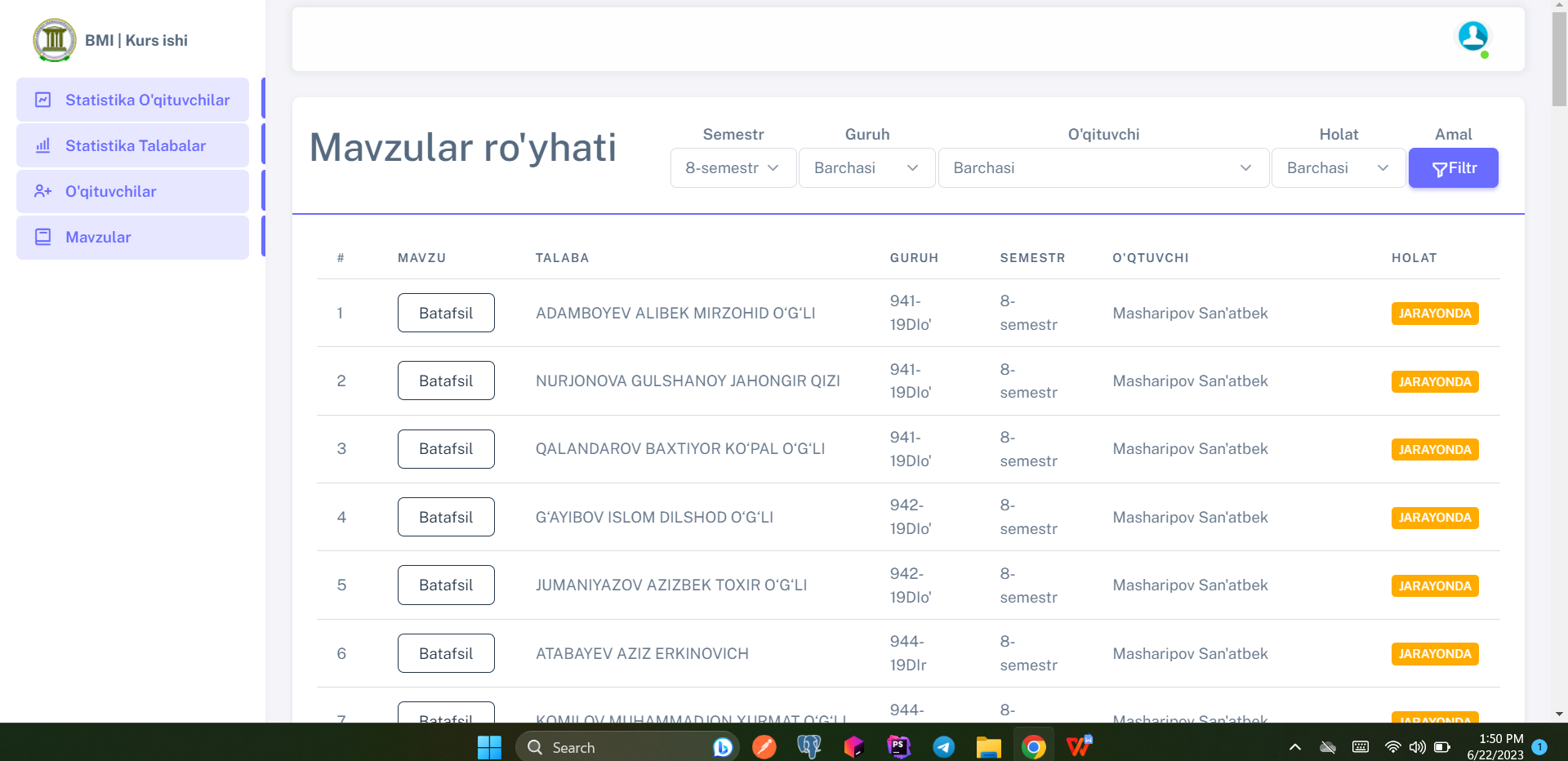




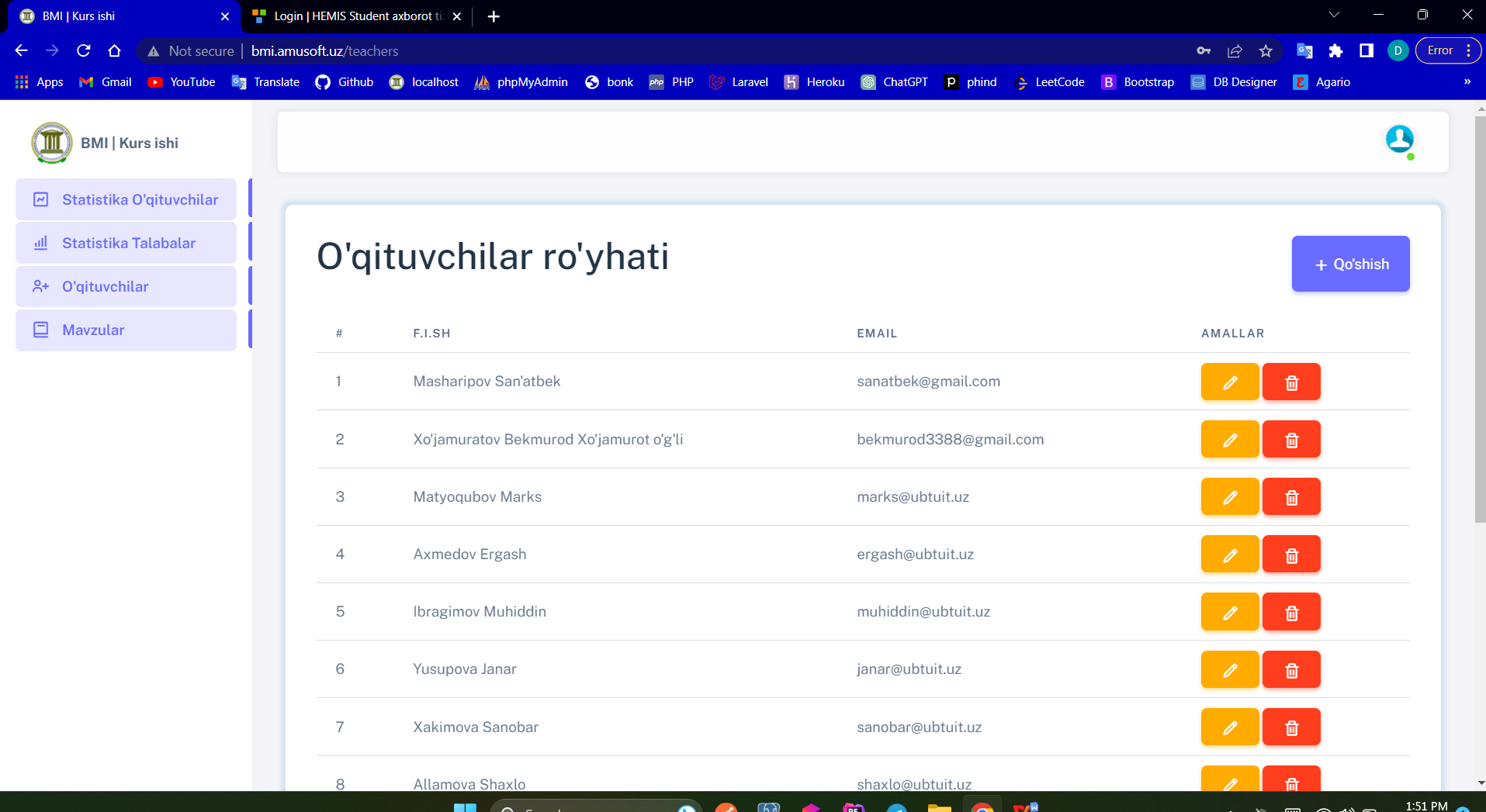
Chat bo’limida ilmiy rahbar bilan xabr almashish mumkin



Namunalar bo’limida mavjud namunalar bilan tanishish mumkin.



Mudir profilida quyidagi amallarni bajarish mumkin



**O’qituvchilar yaratish oynasi**

# Xulosa

Men quyidagilarni o’rgandim:

1. Ma'lumotlar bazasi strukturasi: HEMIS tizimining ma'lumotlar bazasini yaxshilash uchun ma'lumotlar modellari, migratsiya fayllari va bilan bog'liq bo'lgan seederlar yaratish kerak bo'ladi. Laravel freymvorki yordamida ma'lumotlar bazasini migratsiya qilish va qo'shimcha ma'lumotlarni kiritishning bir qancha usullari mavjud.

2. Backend API integratsiyasi: HEMIS tizimining Backend API-si orqali guruhlar ro'yhatini olish uchun, API-ni chaqirish uchun Laravelning HTTP klientlaridan, `Guzzle`dan foydalandim . API-dan ma'lumotlarni olish va ularni Laravel tizimida yoqilgan modellarga yuklash uchun to'g'ri so'rovlar va ma'lumotlar strukturasi kerak bo'ladi.

3. Vaqtincha session da saqlash: Laravel freymvorki sizga sessionlarni yoqish va ma'lumotlarni saqlash imkoniyatini beradi. Guruhlar bilan ishlash davomida zarur bo'lgan ma'lumotlarni sessionlarda saqlashingiz mumkin. Siz ushbu ma'lumotlarga kerakli ko'rinishga to'g'ri keladigan Laravelning sessiyalarini va sessiya ma'lumotlarini ishlatish mumkin.

4. Statistika PDF fayl shaklida generatsiyasi: Laravel freymvorki PDF fayllarni generatsiya qilishni osonlashtirish uchun qulay kutubxonalarga ega. Masalan, `Dompdf` kutubxonasi, matnlarni PDF formatida generatsiya qilish imkonini beradi. Statistika ma'lumotlarni PHP-dan olib, bu kutubxonalarga o'tkazib, PDF fayllarini generatsiya qilishingiz mumkin.

# Foydalanilgan adabiyotlar

Internet resurslari

<https://www.php.net/>

<https://laravel.com/>

<https://sweetalert2.github.io/>

<https://stackoverflow.com/>

<https://getbootstrap.com/>

<https://chat.openai.com/>

# Ilova

**Web.php fayli**

*<?php  
  
use* Illuminate\Support\Facades\Route;  
*use* App\Http\Controllers\{  
 HemisController,  
 TeacherController,  
 ThemeController,  
 ProcessController,  
 MudirController,  
 StatisticController,  
 ChatController,  
 MixController,  
 SifatController,  
 SuperUserController,  
};  
  
  
Route::middleware('web')->group(*function* (){  
 Route::get('/', [MixController::*class*, 'firstPage'])->name('first-page');  
 Route::get('login', [HemisController::*class*, 'login'])->name('login-student');  
 Route::post('login-student-user', [HemisController::*class*, 'loginUser'])->name('login-student-user');  
});  
  
  
Route::middleware('hemis')->group(*function* () {  
  
 Route::middleware('without\_mudir')->group(*function* (){  
 Route::get('logout-student', [HemisController::*class*, 'logout'])->name('logout-student');  
 Route::get('student-profile', [HemisController::*class*, 'profile'])->name('student-profile');  
 Route::get('student-themes', [ThemeController::*class*, 'themes'])->name('student-themes');  
 Route::get('filtered-student-themes', [ThemeController::*class*, 'themesFilter'])->name('filtered-student-themes');  
 Route::get('get-theme/{id}', [ThemeController::*class*, 'getTheme'])->name('get-theme');  
 Route::get('cancel-theme/{id}', [ThemeController::*class*, 'cancelTheme'])->name('cancel-theme');  
 Route::get('cancel-confirm/{id}', [ThemeController::*class*, 'cancelConfirmTheme'])->name('cancel-confirm');  
 Route::get('process', [ProcessController::*class*, 'student\_index'])->name('process');  
 Route::post('update-process', [ProcessController::*class*, 'update'])->name('update-process');  
 Route::get('show-process/{id}', [ProcessController::*class*, 'showProcess'])->name('show-process');  
 Route::post('send-message', [ChatController::*class*, 'create'])->name('send-message');  
 Route::get('chat-student',[ChatController::*class*, 'showChatForStudent'])->name('chat-student');  
 Route::get('examples',[MixController::*class*, 'examples'])->name('examples');  
 });  
  
  
 Route::middleware('mudir')->group(*function*(){  
 Route::resource('teachers',TeacherController::*class*);  
 Route::get('mudir-themes', [MudirController::*class*, 'themes'])->name('mudir-themes');  
 Route::get('filtered-themes', [MudirController::*class*, 'filteredThemes'])->name('filtered-themes');  
 Route::get('statistics-teacher', [StatisticController::*class*, 'teachers'])->name('statistics-teacher');  
 Route::get('statistics-student', [StatisticController::*class*, 'students'])->name('statistics-student');  
  
 });  
  
 Route::middleware('teacher')->group(*function*(){  
 Route::get('themes', [ThemeController::*class*, 'index'])->name('themes');  
 Route::get('filtered-teacher-themes', [ThemeController::*class*, 'filter'])->name('filtered-teacher-themes');  
 Route::post('store-theme', [ThemeController::*class*, 'store'])->name('store-theme');  
 Route::post('update-theme', [ThemeController::*class*, 'update'])->name('update-theme');  
 Route::post('delete-theme', [ThemeController::*class*, 'delete'])->name('delete-theme');  
 Route::get('chat/{id}', [ChatController::*class*, 'show'])->name('chat');  
  
 });  
  
 Route::middleware('sifat\_bolimi')->group(*function* (){  
 Route::get('sifat-bolimi/statistika',[SifatController::*class*,'statisticsAll'])->name('sifat-bolimi-statistika');  
 Route::get('sifat-bolimi/statistika/print',[SifatController::*class*,'generateFile'])->name('sifat-bolimi-print');  
 });  
 Route::middleware('super')->group(*function* (){  
  
 Route::get('mudirlar',[SuperUserController::*class*,'mudirlar'])->name('mudirlar');  
 Route::post('mudirlar/create',[SuperUserController::*class*,'mudirCreate'])->name('create-mudir');  
 Route::put('mudirlar/update/{id}',[SuperUserController::*class*,'mudirUpdate'])->name('update-mudir');  
 Route::delete('mudirlar/delete/{id}',[SuperUserController::*class*,'mudirDelete'])->name('delete-mudir');  
  
 });  
 Route::middleware('auth')->group(*function* (){  
 Route::get('profile',[MixController::*class*,'profile'])->name('profile');  
 Route::post('update-profile/{user}',[MixController::*class*,'updateProfile'])->name('update-profile');  
 Route::post('update-password/{user}',[MixController::*class*,'updatePassword'])->name('update-password');  
 });  
  
  
  
  
  
  
});  
Route::get('oauth',[\App\Http\Controllers\OAuthController::*class*,'index'])->name('oauth');  
Route::get('oauth/callback',[\App\Http\Controllers\OAuthController::*class*,'callback'])->name('oauth.callback');  
  
*require\_once* \_\_DIR\_\_ . '/auth.php';

**ThemeController.php**

*<?php  
  
namespace* App\Http\Controllers;  
  
*use* App\Models\Chat;  
*use* App\Models\Message;  
*use* App\Models\Process;  
*use* App\Models\Theme;  
*use* App\Services\ThemeService;  
*use* Illuminate\Http\Request;  
*use* Illuminate\Support\Facades\Schema;  
  
*class* ThemeController *extends* Controller  
{  
 *public function* index()  
 {  
 $themes = Theme::*where*('semester', '8-semestr')  
 ->where('teacher\_id', auth()->id())  
 ->where('specialty', 5330600)  
 ->select(array\_diff(  
 Schema::getColumnListing('themes'),  
 ['created\_at', 'updated\_at']  
 ))  
 ->get();  
  
  
  
  
 $options = (*object*)[  
 'specialty' => 5330600,  
 'status' => 0,  
 'semester' => "8-semestr",  
 ];  
 *return* view('admin.themes.index', compact('themes', 'options',));  
 }  
  
 *public function* store(Request $request)  
 {  
 $request->validate([  
 'name' => 'required',  
 'description' => 'required',  
 'specialty' => 'required',  
 'level' => 'required',  
 'semester' => 'required',  
 ]);  
  
 *try* {  
 ThemeService::*create*($request->name, $request->description, $request->specialty, $request->level, $request->semester, auth()->user()->id);  
 *return* redirect()->route('themes')->with('msg', 'Mavzu muvaffaqiyatli yaratildi');  
  
 } *catch* (\Exception $e) {  
 *return* redirect()->back()->withErrors('Xatolik yuz berdi');  
 }  
 }  
  
 *public function* update(Request $request)  
 {  
 $request->validate([  
 'name' => 'required',  
 'description' => 'required',  
 'specialty' => 'required',  
 'level' => 'required',  
 'semester' => 'required',  
 ]);  
 *try* {  
 ThemeService::*update*($request->id, $request->name, $request->description, $request->specialty, $request->level, $request->semester);  
 *return* redirect()->route('themes')->with('msg', 'Mavzu muvaffaqiyatli yangilandi');  
  
 } *catch* (\Exception $e) {  
 *return* redirect()->back()->withErrors($e->getMessage());  
 }  
 }  
  
 *public function* delete(Request $request)  
 {  
 $request->validate([  
 'id' => 'required',  
 ]);  
 *try* {  
 ThemeService::*delete*($request->id);  
 *return* redirect()->route('themes')->with('msg', 'Mavzu muvaffaqiyatli o`chirildi');  
  
 } *catch* (\Exception $e) {  
 *return* redirect()->back()->withErrors($e->getMessage());  
 }  
 }  
  
 *public function* getTheme($id)  
 {  
 $theme = Theme::*find*($id);  
 *if* ($theme->student\_id==0 ){  
 *if* (Theme::*all*()->where('student\_id', '=', session('hemisaboutme')->student\_id\_number)->count() ==0){  
 $theme->group\_name = session('hemisaboutme')->group->name;  
 $theme->student\_name = session('hemisaboutme')->second\_name . ' ' . session('hemisaboutme')->first\_name . ' ' . session('hemisaboutme')->third\_name;  
 $theme->student\_id = session('hemisaboutme')->student\_id\_number;  
 $theme->status = 'process';  
 $theme->save();  
 $process = *new* Process();  
 $process->theme\_id = $id;  
 $process->save();  
 $chat= *new* Chat();  
 $chat->theme\_id=$id;  
 $chat->user\_id=$theme->teacher\_id;  
 $chat->save();  
 *return* redirect()->route('process')->with('msg', 'Mavzu tanlandi');  
 }*else*{  
 *return* redirect()->route('student-themes')->withErrors("Siz boshqa mavzuni tanlab bo'lgansiz");  
 }  
 }*else*{  
 *return* redirect()->route('student-themes')->withErrors("Bu mavzu boshqa talaba tomonidan tanlangan");  
 }  
  
 }  
  
 *public function* cancelTheme($id){  
 $theme= Theme::*find*($id);  
 $theme->percentage=-1;  
 $theme->save();  
 *return* redirect()->route('student-themes')->with('msg', "So`rov yuborildi. O`qituvchi bekor qilishi kutilmoqda");  
 }  
 *public function* cancelConfirmTheme(Request $request,$id){  
 *if* ($request->confirm==0){  
 $theme= Theme::*find*($id);  
 $theme->percentage=0;  
 $theme->save();  
 *return* redirect()->back()->with('msg', "So`rov bekor qilindi");  
 }  
 *else*{  
 *if*($request->confirm==1){  
 $theme= Theme::*find*($id);  
 $theme->percentage=0;  
 $process=Process::*where*('theme\_id',$id)->first();  
 $process->delete();  
 $chat=Chat::*where*('theme\_id',$id)->first();  
 $messages=Message::*where*('chat\_id',$chat->id)->get();  
 *foreach* ($messages *as* $message){  
 $message->delete();  
 }  
 $theme->student\_id=0;  
 $theme->status='new';  
 $theme->student\_name=*null*;  
 $theme->group\_name=*null*;  
 $theme->save();  
 *return* redirect()->back()->with('msg', "So`rov tasdiqlandi");  
 }  
 }  
  
 }  
*//for teacher  
 public function* filter(Request $request)  
 {  
 $request->validate([  
 'specialty' => 'required',  
 'semester' => 'required',  
 'status' => 'required',  
 ]);  
 $themes=Theme::*all*()  
 ->where('teacher\_id',auth()->id())  
 ->when($request->specialty, *function* ($query) *use* ($request) {  
 *return* $query->where('specialty', $request->specialty);  
 })  
 ->when($request->semester, *function* ($query) *use* ($request) {  
 *return* $query->where('semester', $request->semester);  
 })  
 ->when($request->status != 0, *function* ($query) *use* ($request) {  
 *return* $query->where('status', $request->status);  
 });  
 $options = (*object*)[  
 'specialty' => $request->specialty,  
 'status' => $request->status,  
 'semester' => $request->semester,  
 ];  
  
  
  
 *return* view('admin.themes.index', compact('themes','options'));  
 }  
*//for student  
 public function* themes(){  
 $code=substr(session('hemisaboutme')->specialty->code,0,6);  
 *if* ($code==533050){  
 $themes = Theme::  
 *where*('specialty','LIKE', $code."%")  
 ->where('level', session('hemisaboutme')->level->name)  
 ->where('semester', session('hemisaboutme')->semester->name)  
 ->where(*function* ($query) {  
 $query->where('student\_id', 0)  
 ->orWhere('student\_id', session('hemisaboutme')->student\_id\_number);  
 })  
 ->get();  
 }*else*{  
  
 $themes = Theme::  
 *where*('specialty', session('hemisaboutme')->specialty->code)  
 ->where('level', session('hemisaboutme')->level->name)  
 ->where('semester', session('hemisaboutme')->semester->name)  
 ->where(*function* ($query) {  
 $query->where('student\_id', 0)  
 ->orWhere('student\_id', session('hemisaboutme')->student\_id\_number);  
 })  
 ->get();  
 }  
  
  
 $options = (*object*)[  
 'semester' => session('hemisaboutme')->semester->name,  
  
 ];  
 *return* view('admin.themes.student',compact('themes','options'));  
 }  
 *public function* themesFilter(Request $request){  
 $request->validate([  
 'semester'=>'required'  
 ]);  
 $themes = Theme::  
 *where*('specialty', session('hemisaboutme')->specialty->code)  
 ->where('level', session('hemisaboutme')->level->name)  
 ->where('semester', $request->semester)  
 ->where(*function* ($query) {  
 $query->where('student\_id', 0)  
 ->orWhere('student\_id', session('hemisaboutme')->student\_id\_number);  
 })  
 ->get();  
  
 $options = (*object*)[  
 'semester' => $request->semester,  
 ];  
 *return* view('admin.themes.student',compact('themes','options'));  
 }  
}

**ThemeService.php**

*<?php  
  
namespace* App\Services;  
  
*use* App\Models\Theme;  
  
*class* ThemeService  
{  
 *public static function* create($name, $description, $specialty, $level,$semester, $teacher\_id)  
 {  
 $theme = *new* Theme();  
 $theme->name = $name;  
 $theme->description = $description;  
 $theme->teacher\_id = $teacher\_id;  
 $theme->specialty = $specialty;  
 $theme->level = $level;  
 $theme->semester = $semester;  
 $theme->save();  
 *return* $theme;  
 }  
  
 */\*\*  
 \** ***@throws*** *\Exception  
 \*/  
 public static function* update($id, $name, $description, $specialty, $level,$semester)  
 {  
 $theme = Theme::*find*($id);  
 *if* ($theme->student\_id != 0) {  
 *throw new* \Exception('Mavzu talaba tomonidan tanlanganligi uchun o`zgaritirsh mumkin emas');  
 }  
 $theme->name = $name;  
 $theme->description = $description;  
 $theme->specialty = $specialty;  
 $theme->level = $level;  
 $theme->semester = $semester;  
 $theme->save();  
 *return* $theme;  
 }  
  
 *public static function* delete($id)  
 {  
 $theme = Theme::*find*($id);  
 *if* ($theme->student\_id != 0) {  
 *throw new* \Exception('Mavzu talaba tomonidan tanlanganligi uchun o`chirish mumkin emas');  
 }  
 $theme->delete();  
 *return* $theme;  
 }  
  
 *public static function* studentChatMessagesCount(){  
 $a=Theme::*all*()->where('student\_id',session('hemisaboutme')->student\_id\_number)  
 ->where('semester',session('hemisaboutme')->semester->name)  
 ->first();  
 *if* ($a!=*null*)  
 *return* $a->chat->messages->where('is\_read',*false*)->where('type','1')->count();  
 *return* 0;  
 }  
  
  
  
  
  
  
}

**HemisService.php**

*<?php  
  
namespace* App\Services;  
  
*use* GuzzleHttp\Client;  
*use* GuzzleHttp\Psr7\Request;  
  
*class* HemisService  
{  
 *public static function* login($login,$password)  
 {  
 $client = *new* Client(['verify' => *false*]);  
 $options = [  
 'multipart' => [  
 [  
 'name' => 'login',  
 'contents' => "{$login}"  
 ],  
 [  
 'name' => 'password',  
 'contents' => "{$password}"  
 ]  
 ]];  
 $request = *new* Request('POST', 'https://student.ubtuit.uz/rest/v1/auth/login');  
 $res = $client->sendAsync($request, $options)->wait();  
 $a=json\_decode($res->getBody());  
 *if* ($a->success){  
 session()->put('hemistoken',$a->data->token);  
 session()->put('loggedin',*true*);  
 }  
 *self*::*getMe*();  
 *return* $a->success;  
  
 }  
 *public static function* getMe(){  
  
 $token=session()->get('hemistoken');  
 $client = *new* Client(['verify' => *false*]);  
 $headers = [  
 'Authorization' => 'Bearer '.$token,  
  
 ];  
 $request = *new* Request('GET', 'https://student.ubtuit.uz/rest/v1/account/me', $headers);  
 $res = $client->sendAsync($request)->wait();  
 $a=json\_decode($res->getBody());  
 *if* ($a->success){  
 session()->put('hemisaboutme',$a->data);  
 session()->put('hemisshortname',$a->data->short\_name);  
 session()->put('hemisimage',$a->data->image);  
 }  
 *return* $a->success;  
  
  
 }  
  
  
  
}

**Statistic.php**

*<?php  
  
namespace* App\Services;  
  
*use* App\Models\Kafedra;  
*use* App\Models\Theme;  
*use* App\Models\User;  
*use* Carbon\Carbon;  
*use* Illuminate\Support\Collection;  
*use* Illuminate\Support\Facades\DB;  
  
*class* Statistic  
{  
  
  
 *public static function* teachers($mudir\_id, $options=[]) {  
  
 *if* ($options->year!=0){  
 $startYear='01.09.'.(explode('-',$options->year)[0]);  
 $endYear='01.08.'.(explode('-',$options->year)[1]);  
 $startYear = Carbon::createFromFormat('d.m.Y', $startYear)->startOfDay();  
 $endYear = Carbon::createFromFormat('d.m.Y', $endYear)->endOfDay();  
  
 }*else*{  
 $startYear=0;  
 $endYear=0;  
 }  
  
  
  
 $teachers = User::*with*(['themes' => *function*($query) *use* ($options, $startYear, $endYear) {  
 *if* ($options->semester != 0) {  
 $query->where('semester', $options->semester);  
 }  
 *if* ($options->year != 0) {  
 $query->whereBetween('created\_at', [$startYear, $endYear]);  
 }  
 }])  
 ->where('role', 'teacher')  
 ->when($mudir\_id != 0, *function* ($query) *use* ($mudir\_id) {  
 *return* $query->where('mudir\_id', $mudir\_id);  
 })  
 ->get();  
  
  
  
  
 $data = $teachers->map(*function*($teacher) {  
 $themes=$teacher->themes;  
 $count = $themes->count();  
 $percentage = $teacher->themes->sum('percentage');  
 $new = $teacher->themes->where('status', 'new')->count();  
 $progress = $teacher->themes->where('status', 'process')->count();  
 $end = $teacher->themes->where('status', 'end')->count();  
  
  
  
 *return* [  
 'teacher' => $teacher,  
 'count' => $count,  
 'percentage' => $percentage,  
 'new' => $new,  
 'progress' => $progress,  
 'end' => $end,  
 'themes' => $themes  
 ];  
  
 });  
  
 *//sort by count  
 if* ($options->sort == 'ASC')  
 $data = $data->sortBy('count');  
 *else if* ($options->sort == 'DESC')  
 $data = $data->sortByDesc('count');  
  
 *return* (*object*)$data->toArray();  
 }  
  
 *public static function* students($mudir\_id,$options){  
  
  
 $themes = Theme::*whereHas*('teacher', *function* ($query) *use* ($mudir\_id) {  
 $query->where('mudir\_id', $mudir\_id);  
 })  
 ->where('semester', $options->semester)  
 ->when($options->group != 0, *function* ($query) *use* ($options) {  
 *return* $query->where('group\_name', $options->group);  
 })  
 ->orderBy('percentage',$options->sort)  
 ->get();  
  
  
 *return* $themes;  
  
 }  
  
 *public static function* themesGivenOptionsSelectedByStudents($options):*array*{  
 $startYear=explode('-',$options->year);  
 $endYear=strtotime('1.08.'.$startYear[1]);  
 $startYear=strtotime('1.09.'.$startYear[0]);  
  
 $themes = DB::table('themes')  
 ->join('users', 'themes.teacher\_id', '=', 'users.id')  
 ->when($options->semester != 0, *function* ($query) *use* ($options) {  
 *return* $query->where('themes.semester', $options->semester);  
 })  
 ->whereDate('themes.created\_at','>=',date('Y-m-d',$startYear))  
 ->whereDate('themes.created\_at','<=',date('Y-m-d',$endYear))  
 ->where('themes.student\_id','!=',0)  
 ->select('themes.student\_name','themes.name', 'themes.group\_name', 'themes.percentage','users.name as teacher\_name','users.mudir\_id')  
 ->orderBy('themes.group\_name',$options->sort)  
 ->get()  
 ->groupBy('mudir\_id')  
 ->toArray();  
 $keys=array\_keys($themes);  
 *if* (!in\_array($options->mudir\_id,$keys) *and* $options->mudir\_id!=0){  
 *return* [];  
 }  
 $data=[];  
 *foreach* ($keys *as* $key){  
 $a=[];  
 $a['kafedra']=Kafedra::*all*()->where('user\_id',$key)->first()->name;  
 $a['themes']=$themes[$key];  
 $data[$key]=$a;  
 }  
  
 *if* ($options->mudir\_id !=0){  
 $b=[];  
 $b[]=$data[$options->mudir\_id];  
 *return* $b;  
 }  
  
 *return* $data;  
 }  
 *public static function* getYears(): *array* {  
 *return* DB::table('themes')  
 ->select(DB::raw("DISTINCT CONCAT(  
 IF(MONTH(created\_at) >= 9, YEAR(created\_at), YEAR(created\_at) - 1),  
 '-',  
 IF(MONTH(created\_at) >= 9, YEAR(created\_at) + 1, YEAR(created\_at))  
 ) AS year"))  
 ->orderBy('year')  
 ->get()  
 ->pluck('year')  
 ->toArray();  
 }  
  
}