**MUNDАRIJА**

**KIRISH**............................................................................................................... 2

1. **TIZIMLI TAHLIL………………………………………………………...**  3
   1. PHP da 2 faktorli avtorizatsiya web ilova tahlili..…………….…………. 3
   2. Web sahifalarni yaratish usul va texnologiyalari………………………… 5
   3. Masalani qo’yilishi …………………………………...……..................... 7
2. **BOB. PHP da 2 faktorli avtorizatsiya web ilovasini yaratish.……….. 8**
   1. Web ilova yaratish uchun dasturlash muhitini sozlash……………........... 8
   2. PHP da 2 faktorli avtorizatsiya web ilovasini yaratish............................... 14
   3. Foydalanuvchi uchun yo‘riqnoma...............................…………………... 18

**XULOSA………………………………………………………………....** 22

**FOYDALANILGAN ADABIYOTLAR……………………………….. 23**

**ILOVA........................................................................................................** 24

**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 Microsoft Internet Explorer va Netscape Navigator 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 PHP da 2 faktorli avtorizatsiya web ilova tahlili.**

2FA(2 faktorli avtorizatsiya) - bu onlayn akkauntga kirishga harakat qilayotgan odamlar o'zlarining kim ekanliklariga ishonch hosil qilish uchun foydalaniladigan qo'shimcha xavfsizlik qatlami. Birinchidan, foydalanuvchi o'z foydalanuvchi nomi va parolini kiritadi. Keyin, darhol kirish huquqiga ega bo'lish o'rniga, ular boshqa ma'lumotni taqdim etishlari talab qilinadi.

 Agar siz foydalanadigan saytga kirish uchun faqat parol talab qilinsa va 2FA taklif qilmasa, u oxir-oqibat buzib kirishi ehtimoli katta. Bu barcha 2FA bir xil degani emas. Bugungi kunda ikki faktorli autentifikatsiyaning bir necha turlari qo'llanilmoqda; ba'zilari boshqalarga qaraganda kuchliroq yoki murakkabroq bo'lishi mumkin, ammo barchasi faqat parollardan ko'ra yaxshiroq himoyani ta'minlaydi. 2FA ning eng keng tarqalgan shakllarini ko'rib chiqamiz.

2FA ning eng qadimgi shakli,  apparat tokenlari  kalit kabi kichik va har 30 soniyada yangi raqamli kod ishlab chiqaradi. Foydalanuvchi hisob qaydnomasiga kirishga harakat qilganda, ular qurilmaga qarashadi va ko‘rsatilgan 2FA kodini yana sayt yoki ilovaga kiritadilar. Uskuna tokenlarining boshqa versiyalari kompyuterning USB portiga ulanganda 2FA kodini avtomatik ravishda uzatadi.

Biroq, ularning bir nechta salbiy tomonlari bor. Korxonalar uchun bu birliklarni tarqatish qimmatga tushadi. Va foydalanuvchilar o'zlarining o'lchamlari ularni yo'qotish yoki noto'g'ri joylashtirishni osonlashtiradi. Eng muhimi, ular xakerlik hujumidan butunlay xavfsiz emas.

SMS-ga asoslangan 2FA  to'g'ridan-to'g'ri foydalanuvchining telefoni bilan o'zaro ishlaydi. Foydalanuvchi nomi va parolni olgandan so'ng, sayt foydalanuvchiga matnli xabar orqali noyob bir martalik parolni (OTP) yuboradi. Uskuna tokeni jarayoni singari, foydalanuvchi ruxsat olishdan oldin dasturga qayta OTPni kiritishi kerak. Xuddi shunday, ovozga asoslangan 2FA avtomatik ravishda foydalanuvchini teradi va og'zaki 2FA kodini etkazib beradi. Oddiy bo'lmasa-da, u hali ham smartfonlar qimmat bo'lgan yoki uyali aloqa xizmati yomon bo'lgan mamlakatlarda qo'llaniladi.

Xavfi past bo'lgan onlayn faoliyat uchun matn yoki ovoz orqali autentifikatsiya qilish kerak bo'lishi mumkin. Ammo shaxsiy ma'lumotlaringizni saqlaydigan veb-saytlar, masalan, kommunal xizmatlar, banklar yoki elektron pochta hisoblari uchun bu 2FA darajasi etarlicha xavfsiz bo'lmasligi mumkin . Aslida,  SMS foydalanuvchilarni autentifikatsiya qilishning eng xavfsiz usuli hisoblanadi. Shu sababli, ko'plab kompaniyalar SMS-ga asoslangan 2FA dan tashqariga chiqib , o'z xavfsizligini oshirmoqda .

**1.2 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.

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. 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

**2. 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.

**3. 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 nancha kerakdir.

**1.3 Masalaning qo’yilishi**

Hozir internet rivojlanib bormoqda. Dunyodagi hamma insonlar internet orqali ma’lumotlar almashishishadi, internet tarmog’idan juda ko’p maqsadlardan foydalaniladi. Hatto internet saytlari orqali xohlagan narsasiga buyurtma berib savdo sotiq ishlarini ham amalga oshirmoqdalar. Hozirda shunday turli saytlar ham mavjud. Intrnet tarmog’idagi veb-sahifalar kundan kunga ko’payib bormoqda. Bu holat O’zbekistonda ham jadallik bilan amalga oshmoqda. Kimdir o’z shaxsiy sahifasini yaratsa, o’quv muassasalari, 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 veb-saytlarni yaratmoqda.

**Ushbu kurs ishi esa “PHP da 2 faktorli avtorizatsiya” web ilovasini yaratishga mo'ljallangan bo'lib, saytni tuzish jarayonida quyidagi ishlar amalga oshirilishi lozim:**

1. PHP da SMS integratsiya qilishni o’rganish;

2.PHP da 2 faktorli avtorizatsiya web ilova arxitekturasi;

3. PHP da 2 faktorli avtorizatsiya web ilovasini ishlab chiqish;

**Dasturda quydagi ishlar amalga oshirilishi lozim:**

- Web ilovani tushunarli qilish;

- Web ilovada 2 faktorli avtorizatsiyani qo‘shish;

- Web ilovaning tuzilishi sayt ko’rinishini ishlab chiqish;

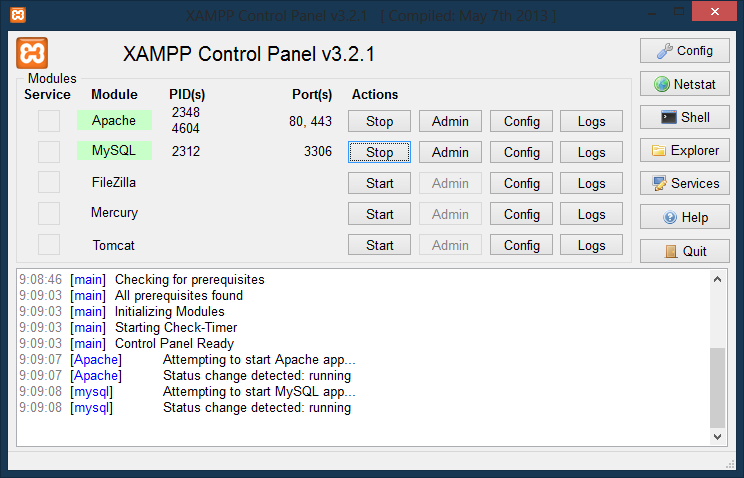
**II.Asosiy qism.**

**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.

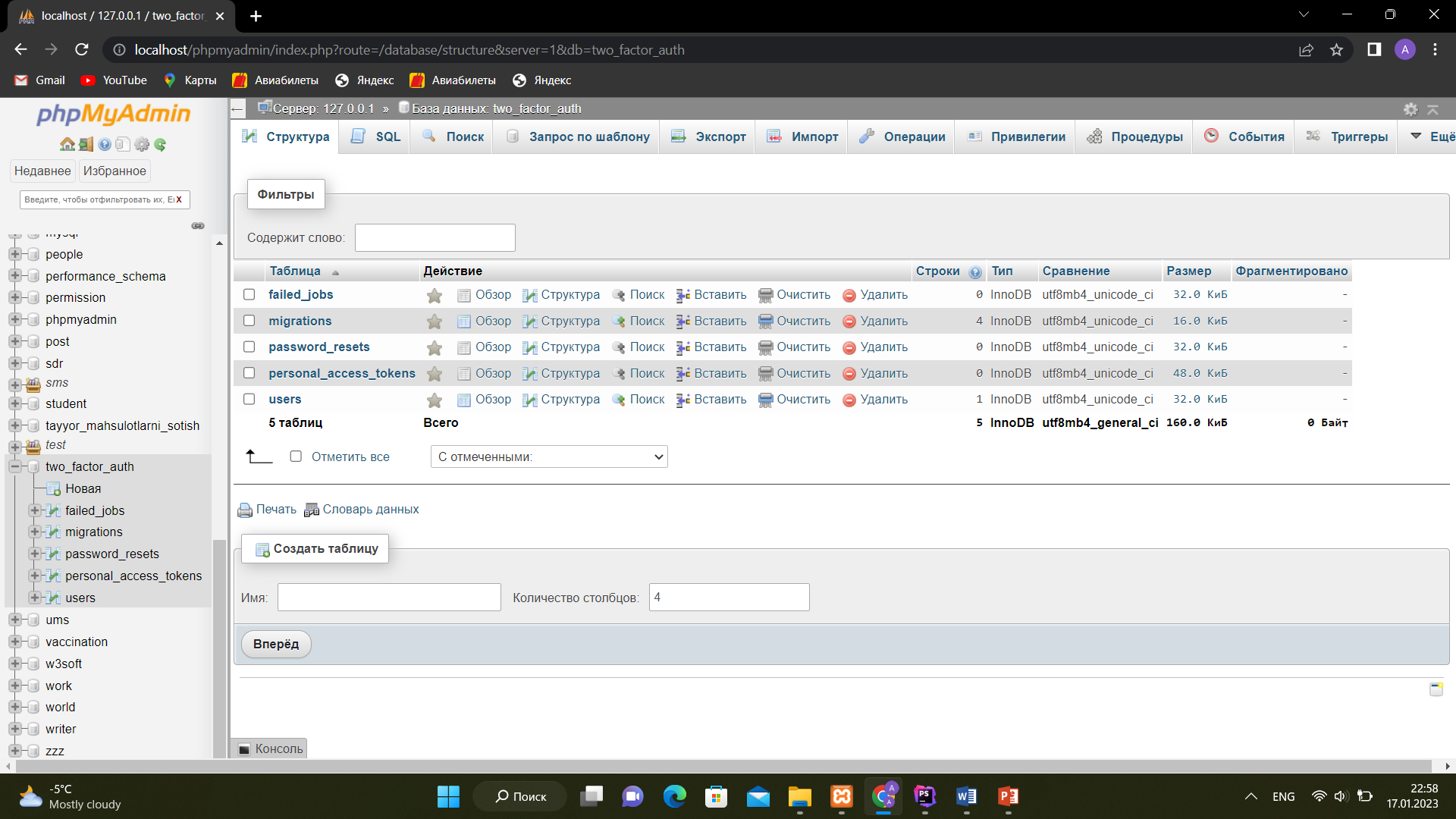
**Web sahifa** va **web saytlarga** xizmat ko`rsatish uchun mezbon kompyuterlar va serverlarda ishlatish uchun server dasturiy ta'minoti kerak bo`ladi. Web saytlar va bu saytlarni yaratish, boshqarish uchun mo`ljallangan dasturiy ta'minot mezbon kompyuterda joylashgan bo`ladi, mezbon kompyuterlar bog`lanish protokollarini boshqaradi. Mezbon kompyuterlar Unix, Windows yoki Macintosh 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.



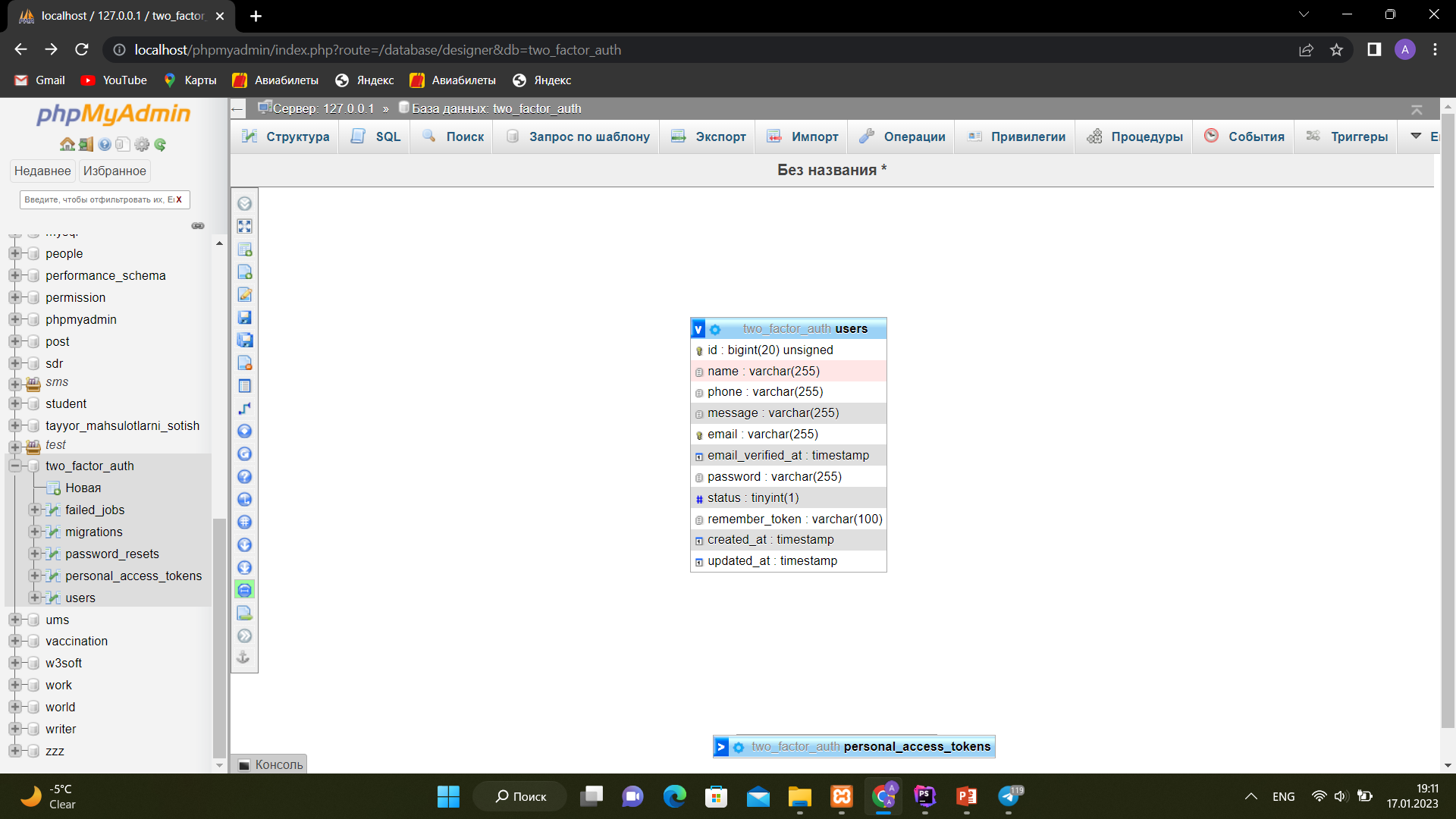
1.1 –rasm. ''Xampp Control Panel''

Bizning ma`lumotlar bazamizni jadvallarini PhpMyadmin dasturi orqali “two\_factor\_auth” deb nomlangan ma’lumotlar bazasini yaratib olamiz.

Ma’lumotlar bazamizni quyidagicha yaratiladi:

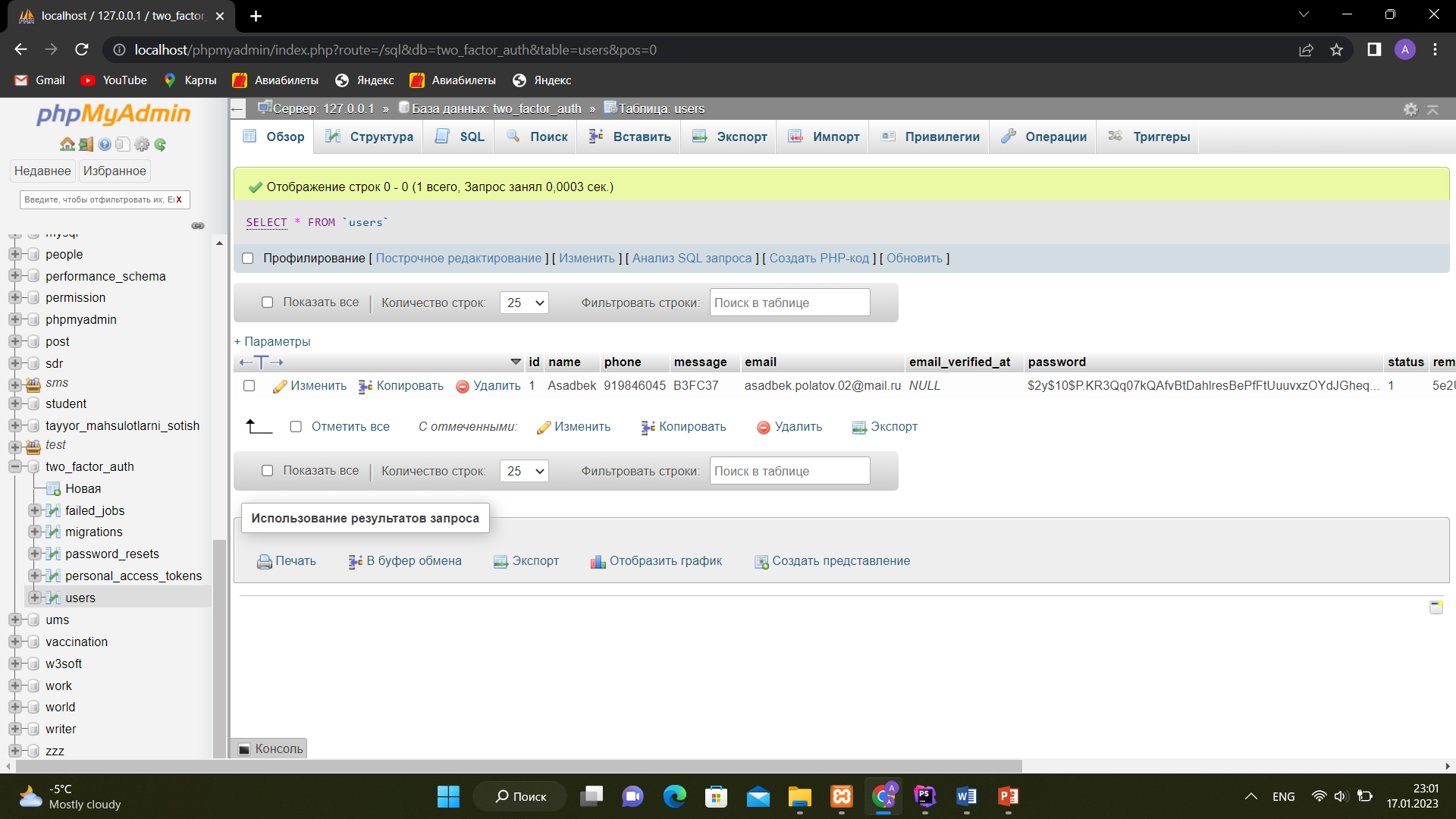


1.2 - rasm Ma’lumotlar bazasi



1.3 – rasm Ma’lumotlar bazasi Strukturasi

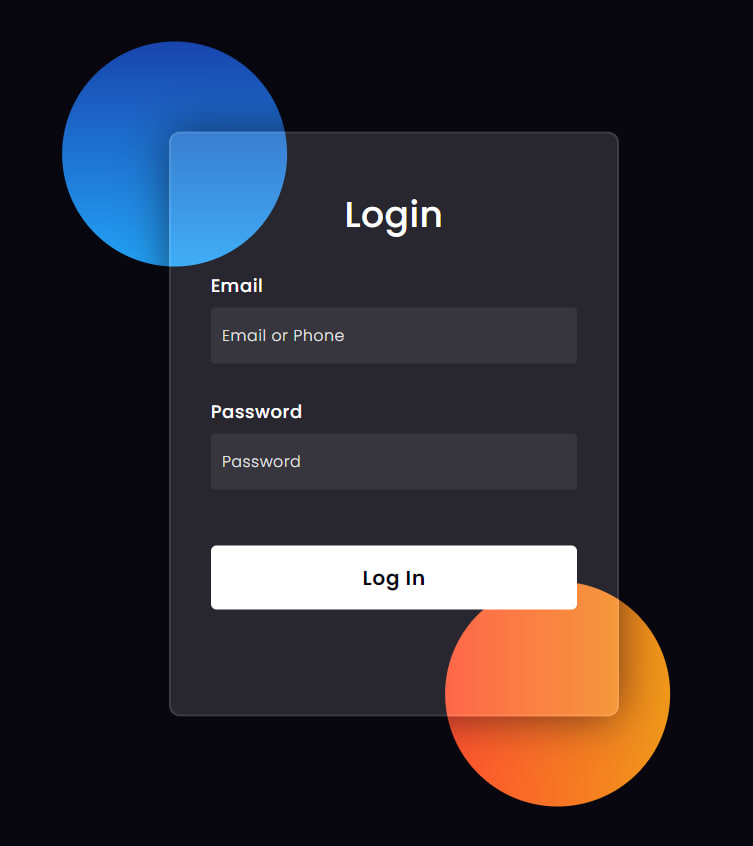
Brauzerga kirib phpmyadminni chaqirib ma’lumotlar bazasini ko’rishimiz mumkin.



1.4 – rasm users jadvalining ma’lumotlar bazasi

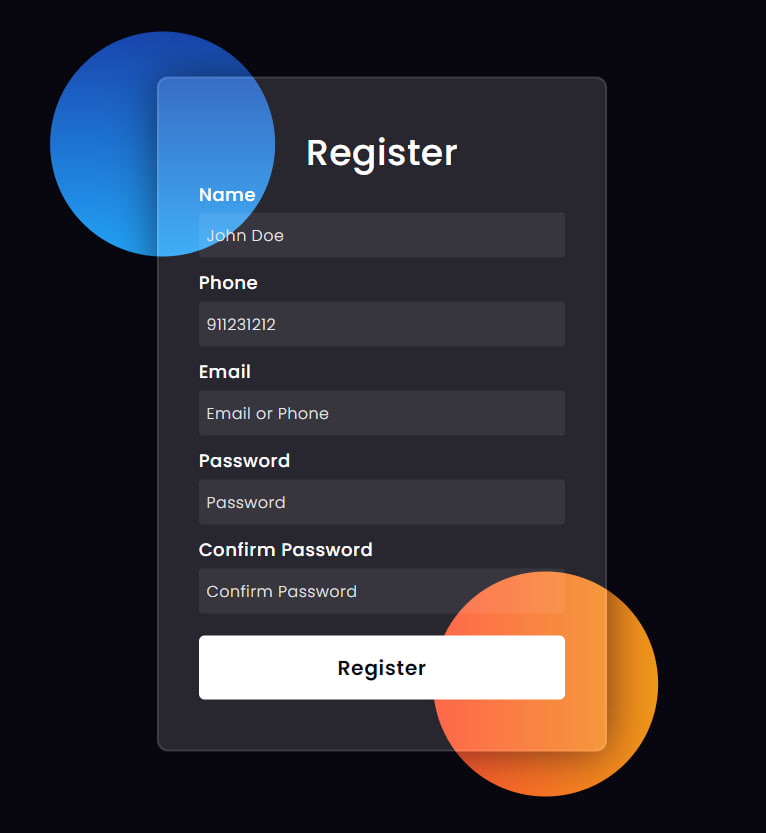
Ma’lumot bazamizda quyidagi jadvallarni yaratib olganimizdan keyin php dasturlash tilida dizaynini yaratishni boshlaymiz.

Web ilovamizni kirish oynasi ko’rinishi quyida ko`rsatilgan:



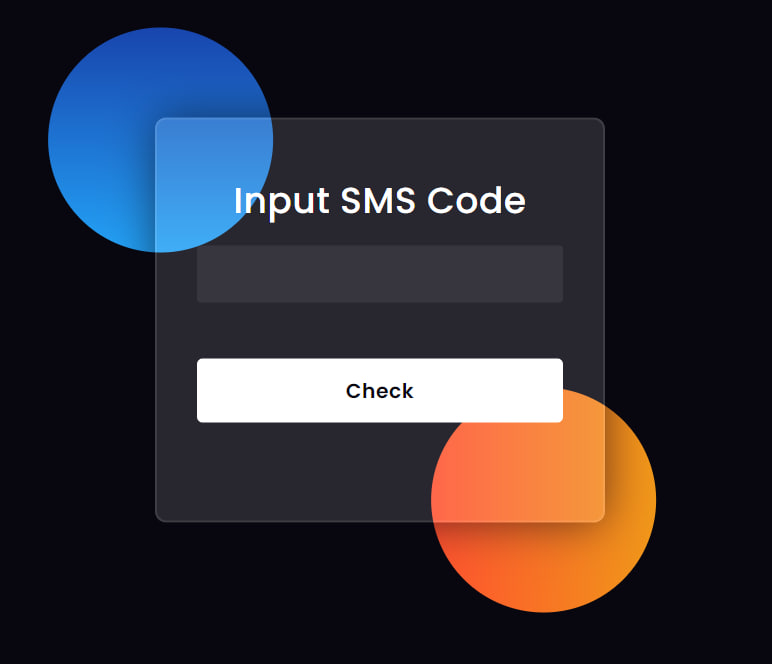
1.5 - rasm “Web ilovamizni kirish oynasi ko’rinishi”

Web ilovamizni ro‘yxatdan o‘tish oynasi ko’rinishi quyida ko`rsatilgan:



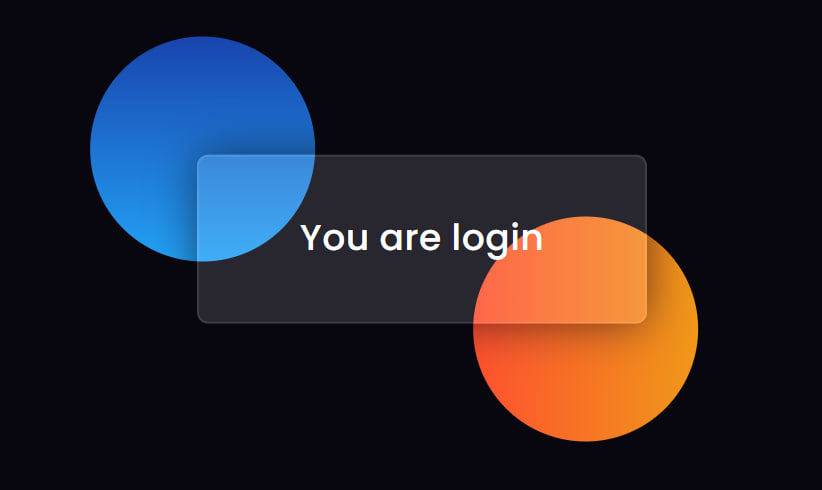
1.6 - rasm “Web ilovamizni ro‘yxatdan o‘tish oynasi ko’rinishi”

Web ilovamizni tasdiqlash oynasi ko’rinishi quyida ko`rsatilgan:



1.7 - rasm “Web ilovamizni tasdiqlash oynasi ko’rinishi”

Ro‘yxatdan o‘tganingizni tasdiqlaganingizdan keyin tizimga kirasiz



1.8 - rasm “Web ilovamizni admin oynasi ko’rinishi”

**2.2 Ikki faktorli avtorizatsiya web ilovasini yaratish. 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:

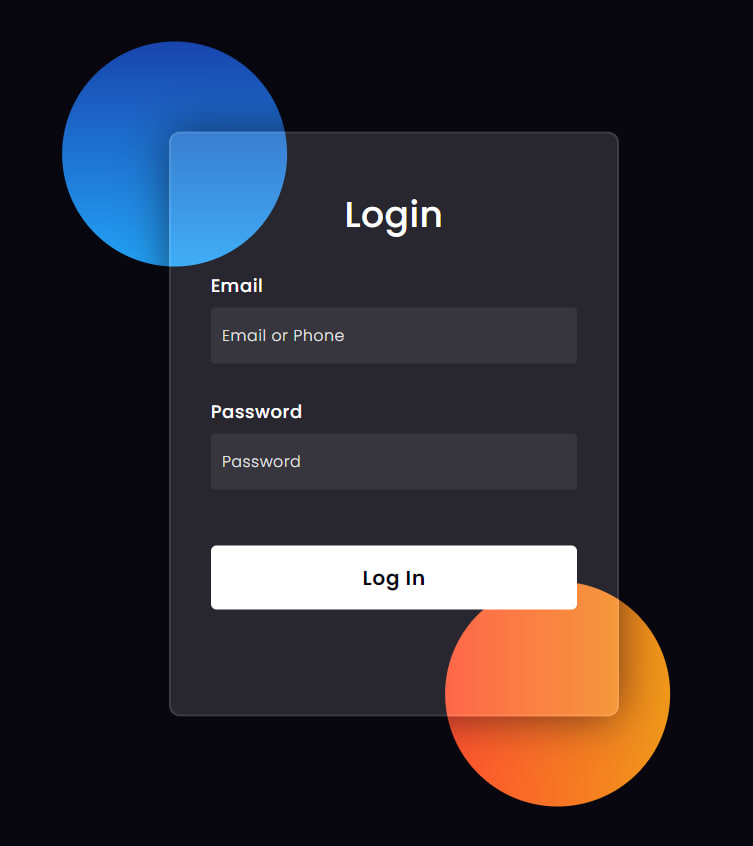
|  |
| --- |
| <?php // birinchi usuldan foydalanilmoqda  $title = “HTML hujjat”;  $ text = “Bu matn PHP asosida yaratilgan”;  ? >  <html>  <head>  < title>  <? // ikkinchi usuldan foydalanilmoqda  eecho $title;  ?>  </title>  </head>  <body>  <script language = “php”> // uchunchi usuldan foydalanilmoqda  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 “2 faktorli avtorizatsiya” web ilovasini yaratishda Laravel framework foydalangan holda yaratiladi.

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 ishini bazasini yaratganimizdan keyin, biz bazani PHP ga bog’lashda Laravel frameworkidagi CRUD dan foydalanamiz.

**2.3 Foydalanuvchi uchun yo’riqnoma**

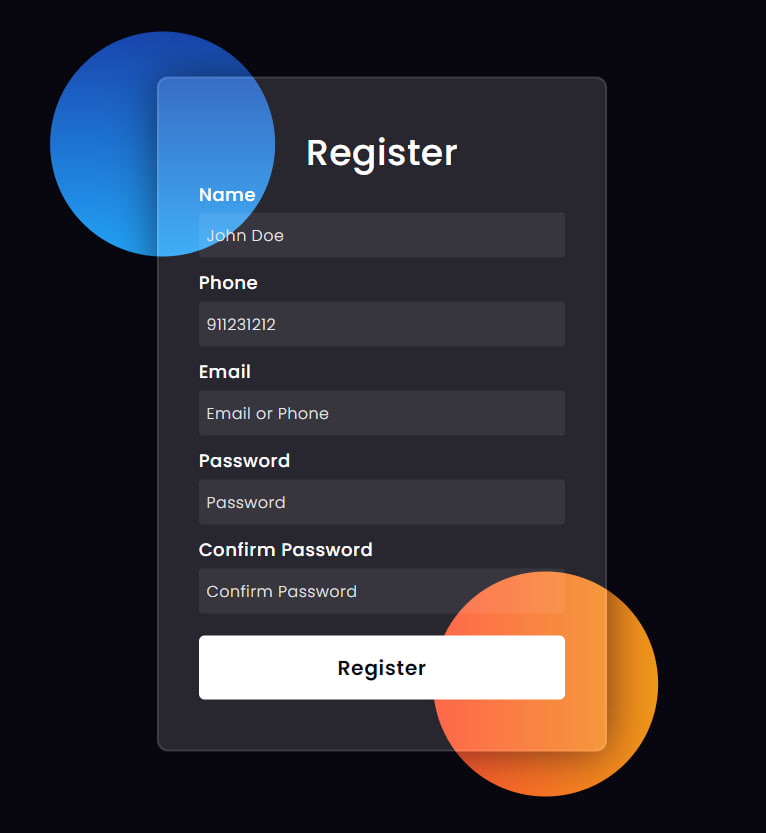
Web ilovamizni kirish oynasi ko’rinishi quyida ko`rsatilgan:



1.5 - rasm “Web ilovamizni kirish oynasi ko’rinishi”

Dasturning kirish qismida foydalanuvchi o‘zining email va parolidan foydalangan holda tizimga kirishi mumkin. Keyin foydalanuvchining telefon raqamiga sms kod keladi. Shu kodni tasdiqlaganidan keyin tizimga kirishi mumkin.

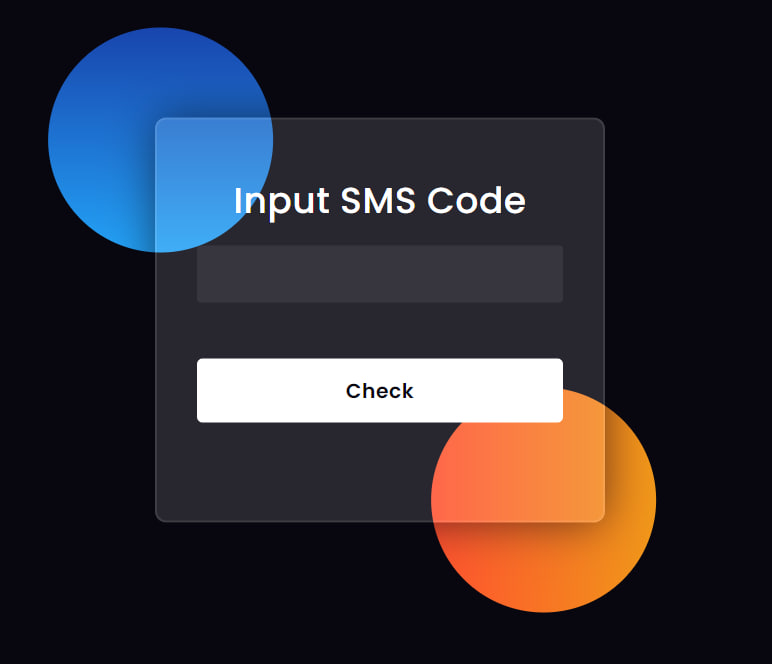
Web ilovamizni ro‘yxatdan o‘tish oynasi ko’rinishi quyida ko`rsatilgan:



1.6 - rasm “Web ilovamizni ro‘yxatdan o‘tish oynasi ko’rinishi”

Dasturning ro‘yxatdan o‘tish qismida foydalanuvchi o‘zining ism familiyasini, telefon raqamini, email va parolini kiritib ro‘yxatdan o‘tishi mumkin. Keyin foydalanuvchining telefon raqamiga sms kod keladi. Shu kodni tasdiqlaganidan keyin tizimga kirishi mumkin.

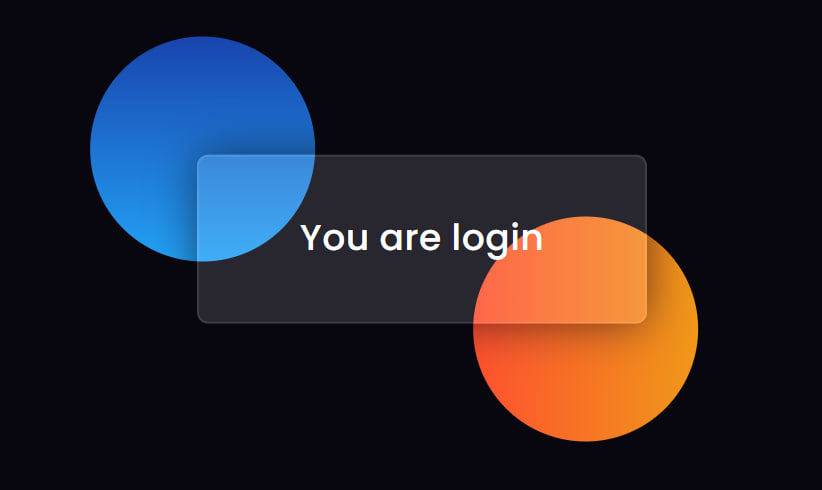
Web ilovamizni tasdiqlash oynasi ko’rinishi quyida ko`rsatilgan:



1.7 - rasm “Web ilovamizni tasdiqlash oynasi ko’rinishi”

Dasturning tasdiqlash qismida telefon raqamiga kelgan sms kodni kiritishi kerak. Shundan keyin foydalanuvchi tizimdan foydalanishi mumkin bo‘ladi.

Ro‘yxatdan o‘tganingizni tasdiqlaganingizdan keyin tizimga kirasiz



1.8 - rasm “Web ilovamizni admin oynasi ko’rinishi”

**Integratsiya qilish**

1. [**https://github.com/AsadbekPulatov/two\_factor\_auth**](https://github.com/AsadbekPulatov/two_factor_auth) githubdan ushbu fayllarni yuklab olasiz.
2. Ilovangiz ichiga kirib terminalga ushbu buyruqni yozasiz: **composer require napaeduhub/eskiz-sms-client**
3. **app/Http/Controllers/AuthController.php** faylni olib o‘zingizning ilovangizning xuddi shu joyiga joylaysiz.
4. **database/migrations/2014\_10\_12\_000000\_create\_users\_table.php** faylni olib o‘zingizning ilovangizning xuddi shu joyiga joylaysiz.
5. **routes/web.php** faylni olib o‘zingizning ilovangizning xuddi shu joyiga joylaysiz.
6. **resources/views/auth** faylni olib o‘zingizning ilovangizning xuddi shu joyiga joylaysiz.

**Xulosa**

Hozirda tizim xavfsizligi juda muhim. Ilovadan hamma foydalanaversa, bu tizim xavfsiz emas. Tizimni xavfsiz qilish uchun 2 faktorli avtorizatsiyani qo‘llashni bilishimiz lozim. Biz tizimni xavfsiz qilsak, ilovada qalbaki foydalanuvchilar bo‘lmasligi, foydalanuvchi uchun tizim xavfsizligi erishamiz. Bundan tashqari foydalanuvchi uchun qulay interfeys qilishimiz kerak. Yaratgan ilovamiz har qanday dasturiy ta’minot uchun ishlay olishi kerak. Web ilovani yaratish davomida PHP dasturlash tilida 2 faktorli avtorizatsiya qilish o‘rganildi.

**Foydalanilgan internet resurslari**1. https:://www.php.net/  
2. https:://www.laravel.com/  
3. https://packagist.org/packages/napaeduhub/eskiz-sms-client**Ilova**

**AuthController.php**

<?php

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Nakroma\Cereal;

use Napa\R19\Sms;

class AuthController extends Controller

{

public function register(Request $request){

$message = Cereal::generate(['length'=>6, 'delimiter'=>'']);

$user\_phone = User::where('phone', $request['phone'])->count();

$user\_email = User::where('email', $request['email'])->count();

if ($user\_email > 0 && $user\_phone>0)

return redirect()->back()->with('error','Registered with phone number and email');

else if($user\_email > 0) return redirect()->back()->with('error','Registered with email');

else if($user\_phone>0) return redirect()->back()->with('error','Registered with phone number');

else if ($request['password'] == $request['confirm\_password']){

$user = new User();

$user['name'] = $request['name'];

$user['phone'] = $request['phone'];

$user['email'] = $request['email'];

$user['password'] = Hash::make($request['password']);

$user['message'] = $message;

$user->save();

Sms::send('998'.$request['phone'], $message);

return view('auth.check',[

'id' => $user->id,

'code' => $user->message,

]);

}

}

public function check(Request $request){

$id = $request['id'];

$user = User::find($id);

if ($request['code'] == $user['message']){

$user->status = 1;

$user->save();

Auth::login($user, 1);

return view('auth.dashboard');

}

else return view('auth.check',[

'id' => $user->id,

"error" => "Invalid code"

]);

}

public function login(Request $request){

$message = Cereal::generate(['length'=>6, 'delimiter'=>'']);

$user = User::where('email', $request['email'])->first();

$user->message = $message;

$user->status = 0;

$user->save();

Sms::send('998'.$user['phone'], $message);

return view('auth.check',[

'id' => $user->id,

'code' => $message,

]);

}

}

**create\_users\_table.php**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

public function up()

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->string('name');

$table->string('phone');

$table->string('message');

$table->string('email')->unique();

$table->timestamp('email\_verified\_at')->nullable();

$table->string('password');

$table->boolean('status')->default(0);

$table->rememberToken();

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('users');

}

};

**web.php**

<?php

use Illuminate\Support\Facades\Route;

Route::get('/', function () {

return view('welcome');

});

Route::get('register', function (){

return view('auth.register');

});

Route::get('login', function (){

return view('auth.login');

});

Route::post('register', [\App\Http\Controllers\AuthController::class,'register'])->name('register');

Route::post('login', [\App\Http\Controllers\AuthController::class,'login'])->name('login');

Route::post('check',[\App\Http\Controllers\AuthController::class,'check'])->name('check');

**login.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Design by foolishdeveloper.com -->

<title>Login</title>

<link rel="preconnect" href="https://fonts.gstatic.com">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap" rel="stylesheet">

<!--Stylesheet-->

<style media="screen">

\*,

\*:before,

\*:after {

padding: 0;

margin: 0;

box-sizing: border-box;

}

body {

background-color: #080710;

}

.background {

width: 430px;

height: 520px;

position: absolute;

transform: translate(-50%, -50%);

left: 50%;

top: 50%;

}

.background .shape {

height: 200px;

width: 200px;

position: absolute;

border-radius: 50%;

}

.shape:first-child {

background: linear-gradient(

#1845ad,

#23a2f6

);

left: -80px;

top: -80px;

}

.shape:last-child {

background: linear-gradient(

to right,

#ff512f,

#f09819

);

right: -30px;

bottom: -80px;

}

form {

height: 520px;

width: 400px;

background-color: rgba(255, 255, 255, 0.13);

position: absolute;

transform: translate(-50%, -50%);

top: 50%;

left: 50%;

border-radius: 10px;

/\*backdrop-filter: blur(10px);\*/

border: 2px solid rgba(255, 255, 255, 0.1);

box-shadow: 0 0 40px rgba(8, 7, 16, 0.6);

padding: 50px 35px;

}

form \* {

font-family: 'Poppins', sans-serif;

color: #ffffff;

letter-spacing: 0.5px;

outline: none;

border: none;

}

form h3 {

font-size: 32px;

font-weight: 500;

line-height: 42px;

text-align: center;

}

label {

display: block;

margin-top: 30px;

font-size: 16px;

font-weight: 500;

}

input {

display: block;

height: 50px;

width: 100%;

background-color: rgba(255, 255, 255, 0.07);

border-radius: 3px;

padding: 0 10px;

margin-top: 8px;

font-size: 14px;

font-weight: 300;

}

::placeholder {

color: #e5e5e5;

}

button {

margin-top: 50px;

width: 100%;

background-color: #ffffff;

color: #080710;

padding: 15px 0;

font-size: 18px;

font-weight: 600;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<div class="background">

<div class="shape"></div>

<div class="shape"></div>

</div>

<form action="{{ route('login') }}" method="post">

<h3>Login</h3>

@csrf

<label for="email">Email</label>

<input type="text" placeholder="Email or Phone" id="email" name="email">

<label for="password">Password</label>

<input type="password" placeholder="Password" id="password" name="password">

<button type="submit">Log In</button>

</form>

</body>

</html>

**register.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Design by foolishdeveloper.com -->

<title>Register</title>

<link rel="preconnect" href="https://fonts.gstatic.com">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap" rel="stylesheet">

<!--Stylesheet-->

<style media="screen">

\*,

\*:before,

\*:after {

padding: 0;

margin: 0;

box-sizing: border-box;

}

body {

background-color: #080710;

}

.background {

width: 430px;

height: 520px;

position: absolute;

transform: translate(-50%, -50%);

left: 50%;

top: 50%;

}

.background .shape {

height: 200px;

width: 200px;

position: absolute;

border-radius: 50%;

}

.shape:first-child {

background: linear-gradient(

#1845ad,

#23a2f6

);

left: -80px;

top: -80px;

}

.shape:last-child {

background: linear-gradient(

to right,

#ff512f,

#f09819

);

right: -30px;

bottom: -80px;

}

form {

height: 600px;

width: 400px;

background-color: rgba(255, 255, 255, 0.13);

position: absolute;

transform: translate(-50%, -50%);

top: 50%;

left: 50%;

border-radius: 10px;

/\*backdrop-filter: blur(10px);\*/

border: 2px solid rgba(255, 255, 255, 0.1);

box-shadow: 0 0 40px rgba(8, 7, 16, 0.6);

padding: 50px 35px;

}

form \* {

font-family: 'Poppins', sans-serif;

color: #ffffff;

letter-spacing: 0.5px;

outline: none;

border: none;

}

form h3 {

font-size: 32px;

font-weight: 500;

line-height: 30px;

text-align: center;

}

label {

display: block;

margin-top: 10px;

font-size: 16px;

font-weight: 500;

}

input {

display: block;

height: 40px;

width: 100%;

background-color: rgba(255, 255, 255, 0.07);

border-radius: 3px;

padding: 0 7px;

margin-top: 5px;

font-size: 14px;

font-weight: 300;

}

::placeholder {

color: #e5e5e5;

}

button {

margin-top: 20px;

width: 100%;

background-color: #ffffff;

color: #080710;

padding: 15px 0;

font-size: 18px;

font-weight: 600;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<div class="background">

<div class="shape"></div>

<div class="shape"></div>

</div>

<form action="{{ route('register') }}" method="post">

<h3>Register</h3>

@csrf

<label for="name" class="form-label">Name</label>

<input type="text" placeholder="John Doe" name="name" id="name">

<label for="phone" class="form-label">Phone</label>

<input type="text" placeholder="911231212" name="phone" id="phone">

<label for="email">Email</label>

<input type="text" placeholder="Email or Phone" id="email" name="email">

<label for="password">Password</label>

<input type="password" placeholder="Password" id="password" name="password">

<label for="confirm\_password" class="form-label">Confirm Password</label>

<input type="password" placeholder="Confirm Password" name="confirm\_password" id="confirm\_password">

<button type="submit">Register</button>

</form>

</body>

</html>

**check.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Design by foolishdeveloper.com -->

<title>Check</title>

<link rel="preconnect" href="https://fonts.gstatic.com">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap" rel="stylesheet">

<!--Stylesheet-->

<style media="screen">

\*,

\*:before,

\*:after {

padding: 0;

margin: 0;

box-sizing: border-box;

}

body {

background-color: #080710;

}

.background {

width: 430px;

height: 360px;

position: absolute;

transform: translate(-50%, -50%);

left: 50%;

top: 50%;

}

.background .shape {

height: 200px;

width: 200px;

position: absolute;

border-radius: 50%;

}

.shape:first-child {

background: linear-gradient(

#1845ad,

#23a2f6

);

left: -80px;

top: -80px;

}

.shape:last-child {

background: linear-gradient(

to right,

#ff512f,

#f09819

);

right: -30px;

bottom: -80px;

}

form {

height: 360px;

width: 400px;

background-color: rgba(255, 255, 255, 0.13);

position: absolute;

transform: translate(-50%, -50%);

top: 50%;

left: 50%;

border-radius: 10px;

/\*backdrop-filter: blur(10px);\*/

border: 2px solid rgba(255, 255, 255, 0.1);

box-shadow: 0 0 40px rgba(8, 7, 16, 0.6);

padding: 50px 35px;

}

form \* {

font-family: 'Poppins', sans-serif;

color: #ffffff;

letter-spacing: 0.5px;

outline: none;

border: none;

}

form h3 {

font-size: 32px;

font-weight: 500;

line-height: 42px;

text-align: center;

}

label {

display: block;

margin-top: 30px;

font-size: 16px;

font-weight: 500;

}

input {

display: block;

height: 50px;

width: 100%;

background-color: rgba(255, 255, 255, 0.07);

border-radius: 3px;

padding: 0 10px;

margin-top: 20px;

font-size: 14px;

font-weight: 300;

}

::placeholder {

color: #e5e5e5;

}

button {

margin-top: 50px;

width: 100%;

background-color: #ffffff;

color: #080710;

padding: 15px 0;

font-size: 18px;

font-weight: 600;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<div class="background">

<div class="shape"></div>

<div class="shape"></div>

</div>

<form action="{{ route('check') }}" method="post">

<h3>Input SMS Code</h3>

@csrf

<input type="text" name="code" maxlength="6">

<input type="hidden" name="id" value="{{ $id }}">

@if(isset($error))

<p>{{ $error }}</p>

@endif

<button type="submit" class="btn btn-primary">Check</button>

</form>

</body>

</html>

**dashboard.blade.php**

<!DOCTYPE html>

<html lang="en">

<head>

<!-- Design by foolishdeveloper.com -->

<title>Check</title>

<link rel="preconnect" href="https://fonts.gstatic.com">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;600&display=swap" rel="stylesheet">

<!--Stylesheet-->

<style media="screen">

\*,

\*:before,

\*:after {

padding: 0;

margin: 0;

box-sizing: border-box;

}

body {

background-color: #080710;

}

.background {

width: 430px;

height: 200px;

position: absolute;

transform: translate(-50%, -50%);

left: 50%;

top: 50%;

}

.background .shape {

height: 200px;

width: 200px;

position: absolute;

border-radius: 50%;

}

.shape:first-child {

background: linear-gradient(

#1845ad,

#23a2f6

);

left: -80px;

top: -80px;

}

.shape:last-child {

background: linear-gradient(

to right,

#ff512f,

#f09819

);

right: -30px;

bottom: -80px;

}

form {

height: 150px;

width: 400px;

background-color: rgba(255, 255, 255, 0.13);

position: absolute;

transform: translate(-50%, -50%);

top: 50%;

left: 50%;

border-radius: 10px;

/\*backdrop-filter: blur(10px);\*/

border: 2px solid rgba(255, 255, 255, 0.1);

box-shadow: 0 0 40px rgba(8, 7, 16, 0.6);

padding: 50px 35px;

}

form \* {

font-family: 'Poppins', sans-serif;

color: #ffffff;

letter-spacing: 0.5px;

outline: none;

border: none;

}

form h3 {

font-size: 32px;

font-weight: 500;

line-height: 42px;

text-align: center;

}

label {

display: block;

margin-top: 30px;

font-size: 16px;

font-weight: 500;

}

input {

display: block;

height: 50px;

width: 100%;

background-color: rgba(255, 255, 255, 0.07);

border-radius: 3px;

padding: 0 10px;

margin-top: 20px;

font-size: 14px;

font-weight: 300;

}

::placeholder {

color: #e5e5e5;

}

button {

margin-top: 50px;

width: 100%;

background-color: #ffffff;

color: #080710;

padding: 15px 0;

font-size: 18px;

font-weight: 600;

border-radius: 5px;

cursor: pointer;

}

</style>

</head>

<body>

<div class="background">

<div class="shape"></div>

<div class="shape"></div>

</div>

<form action="{{ route('check') }}" method="post">

<h3>You are login</h3>

</form>

</body>

</html>