**MUNDARIJA.**

**Kirish**………………...……………………………………………....……………2

**I.Tizimli tahlil……...**……………………………………………………………..4

1.1 API ishlab chiqish imkonini beruvchi web ilova ishlab chiqish tahlili….........4

1.2 Web ilovani yaratish usuli va texnologiyalari……………………………...6

1.3 Masalaning qo’yilishi…………………………………..…………………….11

**II.Asosiy qism………..**……………………………………….............................12

2.1 Web ilova yaratish uchun dasturlash muhitini sozlash……………………….12

2.2 PHP dasturlash tili yordamida “API ishlab chiqish imkonini beruvchi

web ilovasini yaratish”……………….…………………………………………17

2.3 Foydalanuvchi uchun yo`riqnoma……………………………………………21

**Xulosa**.................................................................................................. . ...............24

**Foydalanilgan adabiyotlar**................................................................. . ...............25

**Ilova**...................................................................................................... . ..............26

**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.

**I.Tizimli tahlil**

**1.1 API ishlab chiqish imkonini beruvchi web ilova ishlab chiqish tahlili.**

API "Application Programming Interface" degan ma'noni anglatadi. API interfeysi - bu ikkita dasturning o'zaro ta'sirini talab qiladigan qoidalar to'plami. Ushbu qoidalar to'plamidan foydalangan holda, bu ikki dastur ma'lumotlarni almashadi, API so'rovlarini yuboradi va bir-biridan javob oladi.

Agar veb-ilovalar haqida gapiradigan bo'lsak, bu ikkita dastur mijoz va serverdir va bu o'zaro ta'sir aslida amalga oshiriladi. Mijoz odatda API serveridan kerakli ma'lumotlarni olish uchun so'rovlar yuboradigan dastur bo'lib, API server bu so'rovlarni qabul qiladi, ularni qayta ishlaydi va mijozga javoblarni shakllantiradi.

## Umumiy API turlari

Turli xil API turlari nafaqat arxitektura uslubi, balki veb-yechimlardagi ilovalari bilan ham farqlanadi.

### Open API

Ochiq API-lar odatda mavjud bo'lib, ularni talab qiladigan har bir kishi ulardan foydalanishi mumkin. Ushbu turdagi xizmatlar foydalanuvchi avtorizatsiyasi va autentifikatsiyasiga ega bo'lishi mumkin va ulardan foydalanish uchun ma'lum to'lov talab qilinishi mumkin. Ushbu turdagi API misollariga kelsak, biz umumiy foydalanish uchun Google Maps API va Google Places API kabi xizmatlarni taqdim etadigan Google API-ni eslatib o'tamiz.

Bundan tashqari, tayyor API yechimini topish, uni dasturiy mahsulotlaringizda sinab ko'rish va ushbu API uchun to'lovni amalga oshirishdan oldin baholash imkonini beruvchi ochiq API ro'yxati bilan maxsus tayyorlangan kataloglar mavjud. Masalan, ushbu API'lar Public API, Rapid API va boshqalar bo'lishi mumkin.

### Private API

Ushbu turdagi API turli korxonalar tomonidan ichki tizimlar va ma'lumotlar bazalarini mazmunli qilish uchun ishlatiladi. Ular boshqa foydalanuvchilar uchun mavjud emas, shuning uchun ushbu interfeyslar va ma'lumotlarga uchinchi tomon dasturiy ta'minot etkazib beruvchilari kirishlari mumkin.

### Partner API

Ushbu API nomi o'z-o'zidan gapiradi, ya'ni bu API faqat vakolatli uchinchi tomon ishlab chiquvchilari uchun mavjud. Bunday sherik hamkorlik kelishuviga erishishi yoki ushbu pullik API-lardan foydalanishga ruxsat beruvchi hamkorlik litsenziyasi uchun to'lashi kerak.

**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.

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. 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 ishida qo’yilgan asosiy maqsad, quyidagi vazifalarni hal qilish:**
* - Dasturchilar ishini osonlashtirish;
* - tez va oson boshqa dasturlar uchun ham qo’llash;
* **Dasturda quydagi ishlar amalga oshirilishi lozim:**
* - Web ilovani dasturchilar uchun tushunatrli bo’lishi;
* - Web ilovadan foydalanishni osonlashtirish;

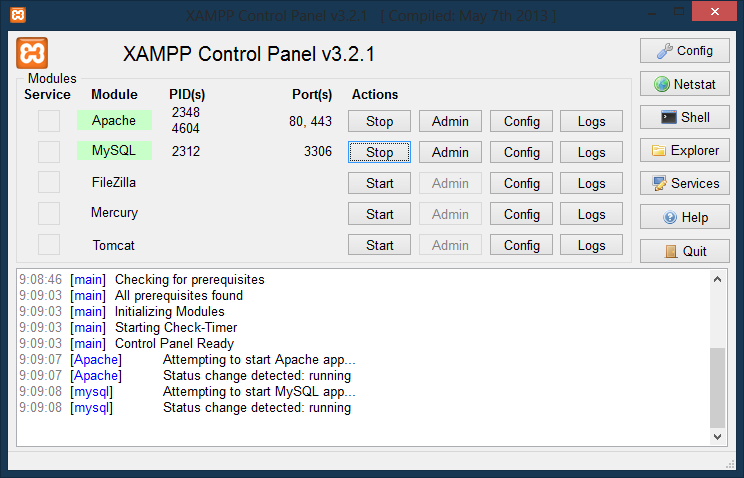
**II.Asosiy qism.**

**2.11 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.

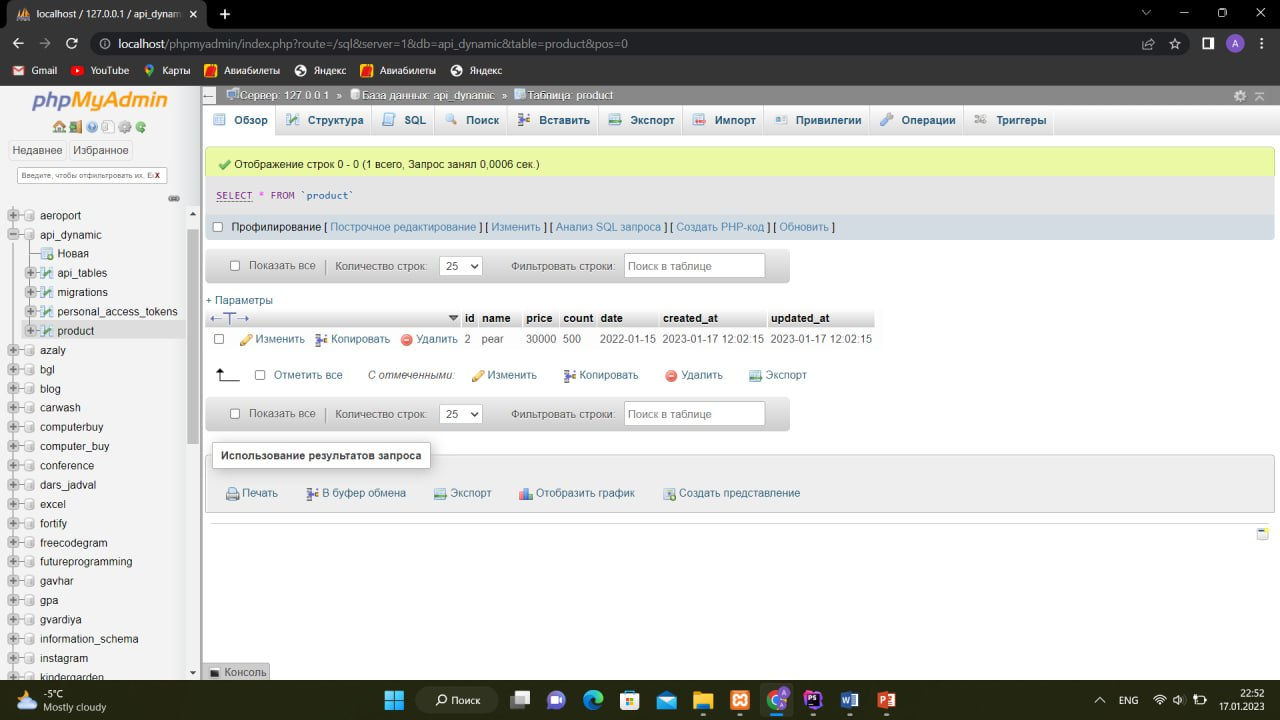
**Web sahifa** va **web saytlarga** xizmat ko`rsatish uchun mezbon kompyuterlar va serverlarda ishlatish uchun server dasturiy ta'minoti kerak bo`ladi. Web saytlar va bu saytlarni yaratish, boshqarish uchun mo`ljallangan dasturiy ta'minot mezbon kompyuterda joylashgan bo`ladi, mezbon kompyuterlar bog`lanish 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 ma’lumotlar bazasini yaratib olamiz.

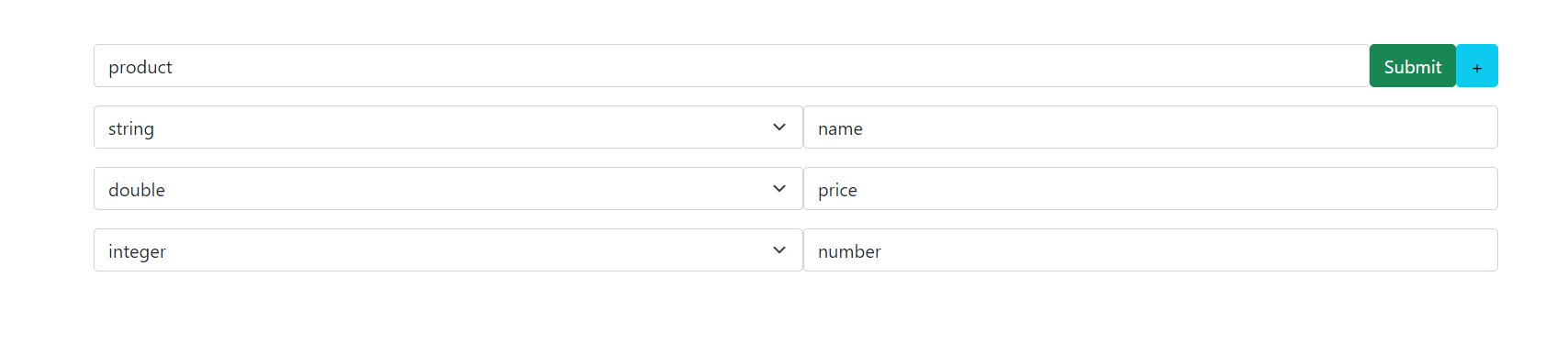
Ma’lumotlar bazamizni quyidagicha yaratiladi:



1.2 - rasm Ma’lumotlar bazasi

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

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



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

**2.2 Api yaratish imkonini beruvchi 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> |

**Xulosa**

Hozirgi davrda O’zbekistonda Internet tizimining jadal kirib borishi unga bog’liq bo’lgan sohalarga ham o’zining katta ta’sirini o’tkazmoqda. Shunday soxalardan biri bu elektron biznes soxasi hisoblanadi. Bu soxada moddiy qiymatdagi va katta xajmdagi mahsulotlarni sotish xali qiyinchilik tug’dirsada informatsion mahsulotlarni sotish internetda elektron pochtaga ega bo’lgan har bir shaxs bilan qiyinchiliksiz amalga oshirilishi mumkin. Buning uchun mijozlarga veb-pullar va internet - magazin haqida tasavvur hosil qilish kerak.

***Web ilovani yaratish davomida quydagi Bilim va ko’nikmalar o’zlashtirildi:***

1. MYSQL ma’lumotlar bazasi bilan ishlash va undagi so’rovlar Update.Select,Insert,Delete so’rovlari o’rganildi;

2.Bu kurs ishini tayyorlashda PHP, HTML,CSS va MySQL dan foydalanildi va ushbu kurs ishini bajarish mobaynida web dasturlash bo’yicha avvalgi bilim va ko’nikmalarni yanada yanada mustahkamladik.

**Foydalanilgan adabiyotlar**

1. Дэвид Скляр. **“Изучаем PHP 7”.** 2017 г.

2**.** Робин Никсон. “ PHP, MySQL, JavaScript, CSS и HTML5 “. 2015 г.

3. Крис Минник, Ева Холланд. “**Javascript для чайников**” 2017 г.

**Internet resurslari**

1. <https://www.w3schools.com/html/>
2. <https://www.php.net/>
3. http://www.google.com

**Ilova**

**ApiController.php**

<?php

namespace App\Http\Controllers;

use App\Models\ApiTable;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Config;

class ApiController extends Controller

{

    public function create(){

        return view('create\_api');

    }

    public function exist($table){

        $count = ApiTable::where('name', $table)->count();

        if ($count > 0){

            return 'Table already exists';

        }else{

            $api = new ApiTable();

            $api->name = $table;

            $api->save();

            return 'Table created';

        }

    }

    public function create\_table($table, $col, $type){

        $sql = "CREATE TABLE $table(";

        $sql .= "id INT(6) AUTO\_INCREMENT PRIMARY KEY, ";

        if (isset($col) && isset($type)){

            foreach ($col as $key => $value) {

                if ($key == count($col)-1)

                    $sql .= $value." ".$type[$key]." NOT NULL";

                else

                    $sql .= $value." ".$type[$key]." NOT NULL, ";

            }

        }

        $sql .= ", created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, ";

        $sql .= "updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP)";

        return $sql;

    }

    public function make\_model($table, $col){

        $file = file\_exists(base\_path("app/Models/{$table}.php"));

        if ($file){

            return 'Model already exists';

        }else{

            $model = fopen(base\_path("app/Models/{$table}.php"), "w");

            $fillable = "protected \$fillable = [";

            foreach ($col as $key => $value) {

                if ($key == count($col)-1)

                    $fillable .= "'".$value."'";

                else

                    $fillable .= "'".$value."', ";

            }

            $fillable .= "];";

            $txt =

                "<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class {$table} extends Model

{

    use HasFactory;

    protected \$table = '{$table}';

    {$fillable}

}

";

            fwrite($model, $txt);

            fclose($model);

            return 'Model created';

        }

    }

    public function make\_controller($table){

        $file = file\_exists(base\_path("app/Http/Controllers/Api/{$table}Controller.php"));

        if ($file){

            return 'Controller already exists';

        }else {

            $controller = fopen(base\_path("app/Http/Controllers/Api/{$table}Controller.php"), "w");

            $txt =

"<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;

use App\Models\\$table;

use Illuminate\Http\Request;

class {$table}Controller extends Controller

{

    public function index()

    {

        \$test = {$table}::all();

        return response()->json(\$test);

    }

    public function store(Request \$request)

    {

        {$table}::create(\$request->all());

        return response()->json(['message' => 'Test created successfully']);

    }

    public function show(\$id)

    {

        \$test = {$table}::find(\$id);

        return response()->json(\$test);

    }

    public function update(Request \$request, \$id)

    {

        \$test = {$table}::find(\$id);

        \$test->update(\$request->all());

        return response()->json(['message' => 'Test updated successfully']);

    }

    public function destroy(\$id)

    {

        \$test = {$table}::find(\$id);

        \$test->delete();

        return response()->json(['message' => 'Test deleted successfully']);

    }

}

";

            fwrite($controller, $txt);

            fclose($controller);

            return 'Controller created';

        }

    }

    public function make\_route($table){

        $file = file\_exists(base\_path("routes/api.php"));

        if ($file){

            $route = fopen(base\_path("routes/api.php"), "a");

            $txt ="Route::apiResource('{$table}',\App\Http\Controllers\Api\\{$table}Controller::class);\n";

            fwrite($route, $txt);

            fclose($route);

            return 'Route created';

        }else{

            return 'Route file not found';

        }

    }

    public function store(Request $request){

        $table = $request->name;

        $col = $request->col;

        $type = $request->type;

        if (!isset($col)){

            return redirect()->back()->with('error', "Please add at least one column");

        }

        $model = $this->make\_model($table, $col);

        if ($model == 'Model already exists'){

            return redirect()->back()->with('error', "Model already exists");

        } else {

            $controller = $this->make\_controller($table);

            $route = $this->make\_route($table);

        }

        $message = $this->exist($table);

        if ($message == 'Table already exists'){

            return redirect()->back()->with('error', $message);

        }

        $sql = $this->create\_table($table, $col, $type);

        $host = Config::get('database.connections.mysql.host');

        $database = Config::get('database.connections.mysql.database');

        $username = Config::get('database.connections.mysql.username');

        $password = Config::get('database.connections.mysql.password');

        $conn = mysqli\_connect($host, $username, $password, $database);

        if (mysqli\_query($conn, $sql)) {

            $message = "index ->".route($table.'.index')."\n";

            $message .= "store ->".route($table.'.store')."\n";

            $message .= "show ->".route($table.'.show', 1)."\n";

            $message .= "update ->".route($table.'.update', 1)."\n";

            $message .= "destroy ->".route($table.'.destroy', 1)."\n";

            return redirect()->back()->with('success', $message);

        } else {

            return redirect()->back()->with('error', 'Error creating table: ' . mysqli\_error($conn));

        }

        $message = array();

        array\_push($message, [

            'function' => 'index',

            'url' => route($table.'.index'),

            'method' => 'GET',

        ]);

        array\_push($message, [

            'function' => 'store',

            'url' => route($table.'.store'),

            'method' => 'POST',

        ]);

        array\_push($message, [

            'function' => 'show',

            'url' => route($table.'.show', 1),

            'method' => 'GET',

        ]);

        array\_push($message, [

            'function' => 'update',

            'url' => route($table.'.update', 1),

            'method' => 'PUT',

        ]);

        array\_push($message, [

            'function' => 'destroy',

            'url' => route($table.'.destroy', 1),

            'method' => 'DELETE',

        ]);

        return redirect()->back()->with('message', $message);

    }

}

**create.api.blade.php**

<!doctype html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport"

          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"

          integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"

            integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"

            crossorigin="anonymous"></script>

    <title>Create Api</title>

</head>

<body>

<div class="container p-5">

    @if( Session::has('error') )

        <div class="alert alert-danger">

            {{ Session::get('error') }}

        </div>

    @endif

    @if( Session::has('success') )

        <div class="alert alert-success">

            {{ Session::get('success') }}

        </div>

    @endif

    <form action="{{ route('api.store') }}" method="post" id="form">

        @csrf

        <div class="d-flex">

            <input type="text" name="name" class="form-control" placeholder="database" required>

            <button type="submit" class="btn btn-success">Submit</button>

            <button type="button" class="btn btn-info" onclick="add()">+</button>

        </div>

    </form>

    <div class="mt-3">

        @if(Session::has('message'))

            <table class="table table-bordered table-info">

                <tr>

                    <th>Function</th>

                    <th>Url</th>

                    <th>Method</th>

                </tr>

                @foreach(Session::get('message') as $msg)

                    <tr>

                        <td>{{ $msg['function'] }}</td>

                        <td>{{ $msg['url'] }}</td>

                        <td>{{ $msg['method'] }}</td>

                    </tr>

                @endforeach

            </table>

        @endif

    </div>

</div>

</body>

</html>

<script>

    function add() {

        let form = document.getElementById('form');

        let add = document.createElement('div');

        add.className = 'd-flex';

        const select = document.createElement('select');

        select.setAttribute('name', 'type[]');

        select.setAttribute('class', 'form-control form-select mt-3');

        select.setAttribute('required', 'required');

        const option1 = document.createElement('option');

        option1.setAttribute('value', 'varchar(255)');

        option1.innerHTML = 'string';

        const option2 = document.createElement('option');

        option2.setAttribute('value', 'int(11)');

        option2.innerHTML = 'integer';

        const option3 = document.createElement('option');

        option3.setAttribute('value', 'double(8, 2)');

        option3.innerHTML = 'double';

        const option4 = document.createElement('option');

        option4.setAttribute('value', 'date');

        option4.innerHTML = 'date';

        const option5 = document.createElement('option');

        option5.setAttribute('value', 'boolean');

        option5.innerHTML = 'boolean';

        select.appendChild(option1);

        select.appendChild(option2);

        select.appendChild(option3);

        select.appendChild(option4);

        select.appendChild(option5);

        add.appendChild(select);

        const col = document.createElement("input");

        col.setAttribute('class', 'form-control mt-3');

        col.setAttribute('type', 'text');

        col.setAttribute('name', 'col[]');

        col.setAttribute('required', 'true');

        add.appendChild(col);

        form.appendChild(add);

    }

</script>

**Migratsiya fayli**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

*/\*\**

*\* Run the migrations.*

*\**

*\* @return void*

*\*/*

    public function up()

    {

        Schema::create('api\_tables', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->timestamps();

        });

    }

*/\*\**

*\* Reverse the migrations.*

*\**

*\* @return void*

*\*/*

    public function down()

    {

        Schema::dropIfExists('api\_tables');

    }

};