|  |
| --- |
| Unit 4: Programming |
| Abdulloh Halilov |

**O‘qituvchi:** Ilyoskhuja Ikromkhujaev

**Guruh identifikatori:** 23-412

**Talaba ID:** 231397

**Taqdim etilgan sana:** 20.01.2025 yil

**BTEC o'quvchilar topshiriqlarini baholash va deklaratsiya**

Baholash uchun ishlarni taqdim etganda, har bir o'quvchi ish o'ziniki ekanligini tasdiqlovchi deklaratsiyani imzolashi kerak.

|  |  |
| --- | --- |
| **O‘quvchi (talaba) identifikatori:** | 231397 |
| **Baholovchi nomi:** | Ilyoskhuja Ikromkhujaev |
| **BTEC dasturi nomi:** | Pearson BTEC Higher Nationals in Digital Technologies |
| **Birlik yoki komponent raqami va nomi:** | Unit 4: Programming |
| **Topshiriq nomi:** | Internal Assignment |
| **Topshiriq topshirilgan sana:** | 20.01.2025 yil |

Iltimos, har bir topshiriq uchun berilgan ishlarni sanab o'ting. Ishlarni topish mumkin bo'lgan sahifa raqamlarini ko'rsating yoki ishlarning mohiyatini tavsiflang (masalan, grafik, rasm).

|  |  |  |
| --- | --- | --- |
| **Topshiriq vazifasi ma'lumoti** | **Ishlar taqdim etildi** | **Sahifa** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **O'quvchi deklaratsiyasi**  Men ushbu topshiriq uchun taqdim etilgan ish meniki ekanligini tasdiqlayman. Ishda foydalanilgan manbalarga aniq havola qildim. Men noto'g'ri deklaratsiya noto'g'ri ishlashning bir shakli ekanligini tushunaman.  **O‘quvchi imzosi:**  **Sana:** 20.01.2025 yil |

Mundarija

[Kirish 3](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462352)

[1-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462353) 5

[2-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462354) ?

[3-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462355) ?

[4-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462356) ?

[Adabiyotlar ro’yxati](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462358) ?

KIRISH

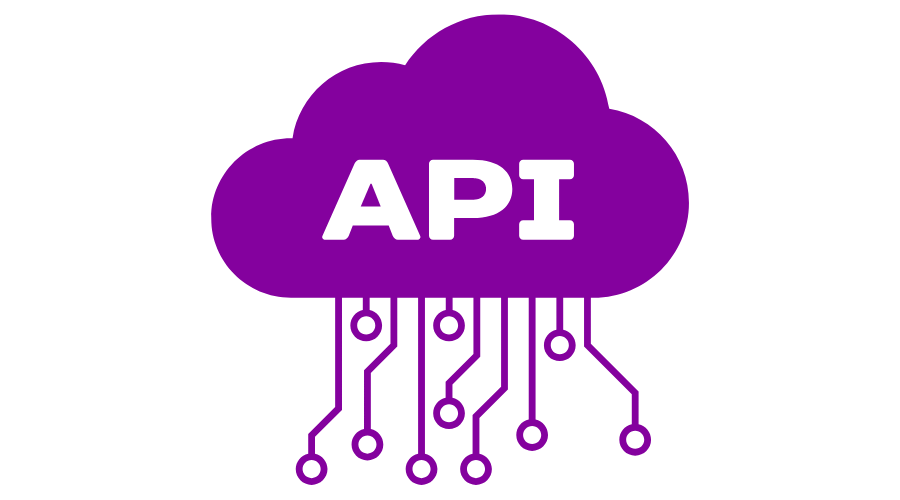
Assalomu alaykum. NewEra Cash & Carry uchun ASP.NET API-ni ishlab chiqish va hujjatlashtirish bo'yicha topshirig'imga xush kelibsiz. Ushbu loyiha modulimizning o'quv maqsadlariga erishish uchun amalga oshirialdi. Mening maqsadim nafaqat ushbu tushunchalarni tushunishimni namoyish etish, balki ularni ASP.NET API ishlab chiqish doirasida amalda qo'llashdir. Ushbu kirish bo'limi nazariy bilimlarni qanday qilib funktsional dasturiy echimlarga aylantirish mumkinligini chuqurroq o'rganishga yordam beradi.

1-QISM

**LO1**

**A.P1**

API so'rovlarni ko'rib chiqish va tegishli javoblarni taqdim etish uchun aniq jarayonni kuzatish orqali ishlaydi. So'rov yuborilganda, API birinchi navbatda uning to'g'ri qabul qilinishini ta'minlaydi va so'rovning URL manzili va HTTP usuli (GET, POST, PUT yoki DELETE kabi) asosida mo'ljallangan oxirgi nuqtani aniqlaydi.



Keyinchalik, JWT Bearer tokeni bilan autentifikatsiya hisob maʼlumotlarini tekshirish orqali soʻrov vakolatli manbadan ekanligini tekshiradi. Shuningdek, foydalanuvchi so'ralgan manbaga kirish uchun zarur ruxsatlarga ega ekanligini tekshiradi. So'rov ushbu xavfsizlik tekshiruvlaridan o'tgandan so'ng, API taqdim etilgan ma'lumotlarni tekshdiradi va barcha kerakli maydonlar mavjudligini va to'g'ri formatlanganligini ta'minlaydi.

Tekshirishdan so'ng, API so'rovni tegishli boshqaruvchiga va harakat usuliga yo'naltirish orqali qayta ishlaydi. So'rovning maqsadiga qarab, bu bosqich ma'lumotlar bazasi yoki boshqa ma'lumotlar manbalari bilan o'zaro aloqani o'z ichiga olishi mumkin. Misol uchun, Admin API so'rovda taqdim etilgan Product ma'lumotlari asosida yozuvlarni olishi, yasashi, yangilashi yoki o'chirishi mumkin. Agar customer bo’lsa faqat Mahsulot ma’lumotlarini ko’ra oladi.

Kerakli operatsiyalar bajarilgandan so'ng, MVC mijoz uchun javob tayyorlaydi. Bu javob JSON kabi tuzilgan formatdagi maʼlumotlarni oʻz ichiga oladi va soʻrov muvaffaqiyatli yoki xatolik yuz berganligini koʻrsatish uchun HTTP holat kodini oʻz ichiga oladi (masalan, muvaffaqiyat uchun 200, notoʻgʻri soʻrov uchun 400, ruxsatsiz kirish uchun 401 va hokazo). .

Nihoyat, API javobni mijozga yuboradi va so'rov va javob tafsilotlarini qayd qiladi. Ushbu Request Response recordlari Middleware yordamida fayllarga saqlanadi. Ushbu jarayon API so'rovlarni samarali va xavfsiz tarzda ko'rib chiqishni ta'minlaydi.

**A.M1**

**Bu bo’limda biz kodni yozishdan to ijro etishgacha bo’lgan jarayonlarni ko’rib chiqamiz:**

**Kodni yozishdan bajarishgacha bo'lgan jarayon bir nechta muhim bosqichlarni o'z ichiga oladi, bu dastur funktsiyalarini mo'ljallangan tarzda ta'minlaydi. Bu kod yozishdan boshlanadi, bu yerda ishlab chiquvchi algoritm yoki mantiqni ASP.NET uchun C# kabi dasturlash tiliga tarjima qiladi. Ushbu bosqichda eng yaxshi amaliyotlarga rioya qilish, jumladan toza, modulli va yaxshi hujjatlashtirilgan kodni saqlash juda muhimdir.**

**Kod yozilgach, u kompilyatsiya jarayonidan o'tadi. Kompilyatsiya - bu kod inson tomonidan o'qiladigan yuqori darajadagi sintaksisdan oraliq yoki mashinada o'qiladigan kodga aylantiriladi. Masalan, ASP.NET da yozma kod .NET ish vaqti sharhlashi mumkin bo'lgan oraliq tilga (IL) kompilyatsiya qilinadi. Ushbu bosqich sintaktik xatolarni hal qilishni o'z ichiga oladi va kodning sintaktik jihatdan to'g'riligini va bajarishga tayyorligini ta'minlaydi.**

**Muvaffaqiyatli kompilyatsiyadan so'ng, dastur qurish bosqichiga o'tadi, bu erda loyihaning barcha komponentlari, masalan, kutubxonalar va resurslar joylashtiriladigan birlikka to'planadi. Bu ko'pincha server ishga tushirishi mumkin bo'lgan bajariladigan fayllarni (masalan, .dll yoki .exe) yaratishga olib keladi.**

**Qurilishdan so'ng dastur sinov bosqichiga o'tadi. Ushbu bosqichda kod kutilgandek harakat qilishini tekshirish uchun boshqariladigan muhitda bajariladi. Bu har qanday mantiqiy yoki ish vaqtidagi xatolarni aniqlash va hal qilish uchun birlik testi, integratsiya testi va funktsional testni o'z ichiga olishi mumkin.**

**Ilova sinovdan o'tgan va xatolar bartaraf etilgandan so'ng, u o'rnatishga tayyor. Joylashtirish ilovani maqsadli muhitga, masalan, ishlab chiqarish serveriga ko'chirishni o'z ichiga oladi, bu erda u oxirgi foydalanuvchilar uchun ochiq bo'ladi. ASP.NET API bo'lsa, o'rnatish APIni IIS kabi veb-serverda yoki Azure kabi bulut xizmatida joylashtirishni o'z ichiga olishi mumkin.**

**Joylashtirilgandan so'ng, dastur ishlashga tayyor. Foydalanuvchilar yoki mijoz tizimlari so'rovlarni yuborish orqali dastur bilan o'zaro aloqada bo'lib, server o'rnatilgan mantiq asosida qayta ishlaydi. Amalga oshirish jarayonida dastur xotira, mavzular va boshqa resurslarni samarali boshqarish uchun .NET Core ish vaqti kabi ish vaqti muhitlariga tayanadi.**

**Butun jarayon takrorlanadi, chunki ishlab chiquvchilar funksiyalar qo‘shish, xatolarni tuzatish yoki ish faoliyatini optimallashtirish uchun kod yozish va o‘zgartirishga qaytishi mumkin, bu esa ilova foydalanuvchi ehtiyojlari va tizim talablariga javob berishda davom etishini ta’minlashi mumkin.**

**A.D1**

**Matn.**

**LO2**

**A.P2**

OOP (Obyektga Yo'naltirilgan Dasturlash)

OOP - bu dasturlash paradigmasi bo'lib, u dasturni obyektlarga ajratish orqali murakkab muammolarni hal qilishga yordam beradi. Obyektlar esa o'z navbatida ma'lumotlar (xususiyatlar) va ular ustida amallarni bajaruvchi funksiyalar (metodlar)ni o'z ichiga oladi.

OOPning asosiy xususiyatlari:

Abstraksiya: Murakkab tizimlarni soddalashtirish uchun faqatgina kerakli ma'lumotlarni o’zida Abstract qilish. Huddi Class larning contexti kabi. Abstract classlardan voris olgan classlarni kontraktini tuzib bergandek.

Inkapsulyatsiya: Obyektning ichki tuzilishini yashirish va tashqi kod bilan faqatgina ma'lum metodlar orqali muloqot qilish imkonini berish. Bu xavfsizlikni oshiradi va kodni o'zgartirishni osonlashtiradi. Masalan, bank hisobi obyektida balansni o'zgartirish uchun faqatgina pul\_qo'shish va pul\_yechish metodlaridan foydalanish mumkin.

Inheretence: Yangi obyektlarni mavjud obyektlar asosida yasash. Bu kodni qayta ishlatish imkonini beradi. Masalan, "elektromobil" obyekti "avtomobil" obyektidan meros olishi va uning barcha xususiyatlari va metodlariga ega bo'lishi mumkin, lekin qo'shimcha ravishda batareya quvvati kabi yangi xususiyatlarga ham ega bo'lishi mumkin.

Polimorfizm: Bir xil nomdagi metodlarning turli xil obyektlar uchun turlicha ishlashi. Bu kodni moslashuvchan qiladi. Masalan, "tovush\_chiqarish" metodi it obyekti uchun "vovulash" degan ma'noni anglatsa, mushuk obyekti uchun "miyovlash" degan ma'noni anglatadi.

Xususiyatlar o'rtasidagi munosabatlar:

Abstraksiya inkapsulyatsiyani qo'llab-quvvatlaydi, chunki u obyektning ichki tuzilishini yashirishga yordam beradi.

Merosxo'rlik kodni qayta ishlatish imkonini beradi va shu bilan birga polimorfizmni qo'llab-quvvatlaydi, chunki meros olingan obyektlar bir xil nomdagi metodlarni turlicha ishlatishi mumkin.

OOP tushunchalari

OOP (Obyektga Yo'naltirilgan Dasturlash) - bu dasturlashda obyektlarga asoslangan yondashuv. Dastur turli xil obyektlarga bo'linadi va ularning o'zaro ta'siri belgilanadi. Har bir obyekt o'z xususiyatlari va metodlariga ega bo'ladi.

**A.M2**

Matn

**A.D2**

Matn

**LO3**

**A.P3**

Matn

**A.M3**

Matn

**A.D3**

Matn

**LO4**

**A.P4**

Matn

**A.P5**

Matn

**A.M4**

Matn

**A.D4**

Matn

XULOSA

Veb saytning maqsadi va foydalanuvchining ehtiyojlari uning ishlab chiqish samaradorligiga katta ta'sir ko'rsatadi. Aniq belgilangan maqsad va ehtiyojlar ishlab chiqish jarayonini samarali va tez bajarishga yordam beradi. Django frameworkining tanlanishi va HTML, CSS yordamida dizaynning yaratilishi jarayonni yanada osonlashtiradi. Foydalanuvchi tajribasini (UX) yaxshilashga qaratilgan qadamlar, sahifalarning tez yuklanishi va responsiv dizayn, saytdan foydalanishni qulaylashtiradi. Admin panel orqali ma'lumotlarni boshqarish imkoniyati esa saytni doimiy yangilanib turishini ta'minlaydi. API qo'shilishi esa ma'lumotlarni qo'shish, o'zgartirish, ko'rish va qidirish imkoniyatlarini beradi, bu esa foydalanuvchilarga va dasturchilarga qulaylik yaratadi. Umuman olganda, maqsad va ehtiyojlarga asoslangan ishlab chiqish jarayoni saytni tez va samarali yaratishga olib keladi.

adabiyotlar ro'yxati

1. Grinstein, G., and Plaisant, C., 2009. Interactive Information Visualization. Springer.
2. Nielsen, J., 1994. Usability Engineering. Academic Press.
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., and Diakopoulos, N., 2017. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th ed. Pearson.
4. Krug, S., 2014. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. 3rd ed. New Riders.
5. Cooper, A., Reimann, R., and Cronin, D., 2007. About Face 3: The Essentials of Interaction Design. 3rd ed. Wiley.
6. Garrett, J.J., 2010. The Elements of User Experience: User-Centered Design for the Web and Beyond. 2nd ed. New Riders.
7. Norman, D.A., 2013. The Design of Everyday Things: Revised and Expanded Edition. Basic Books.