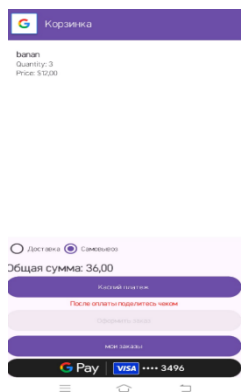
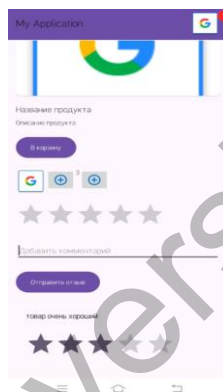


және тұтынушыларға сапалы өнімдерге тікелей қол жеткізуге мүмкіндік береді.



7-Сурет. Өнім терезесі



8-Сурет. Жәшік терезесі

Қолданылған әдебиеттер.

- [1] Х. Ю, Дж. Ким, Дж. К. Ли, " Жергілікті Фермерлерді Қосуға Арналған Мобильді Қосымшаны Өзірлеу Және Тұтынушылар", IEEE Access, том. 8, 108046-108055 беттер, 2020 жыл.
- [2] А. Р. А. Рахман, А. М. А. Рахим, М. А. Шукор, " Ұялы Телефонды Жобалау және Дамыту Көкөніс Саудасының Онлайн Жүйесіне өтінім," 2020 жылғы 7-Ші Халықаралық Конференция Электротехника Және Электроника (ICEEE), Дакка, Бангладеш, 2020, 1-6 беттер.
- [3] С. Ким, Дж. Ким, " Жергілікті Қолдауға Арналған Мобильді Қосымшаны Жобалау және Өзірлеу Ауыл Шаруашылығы Өнімдері, " Ақпарат Және Коммуникациялар Бойынша 2020 Халықаралық Конференциясы Технологиялық Конвергенция (ICTC), Чеджудо Аралы, Оңтүстік Корея, 2020, 1242-1245 беттер.

Кәден Ә., студент

Найзағараева А.А., аға оқытушы

С. Сейфуллин атындағы Қазақ агротехникалық зерттеу университеті

ПАЙДАЛАНУШЫ АҚПАРАТЫН САҚТАУҒА АРНАЛҒАН ҚОЛДАНБА

Жобаның өзектілігі: деректермен жұмыс жасау.

Жобаның пайдалану перспективалары: веб қосымшыларға, мобильдік қосымшыларға, деректермен жұмыс жасау.

«Пайдаланушы ақпаратын сақтауға арналған қолданба» атты жоба пайдаланушы ақпаратын (аты, тегі, әкесінің аты, жынысы, телефон нөмірі, электрондық пошта, электрондық пошта, электрондық пошта, құпия сөз) сақтау үшін CRUD (жасау, оқу, жаңарту, жою) қосымшасын жазыңыз. Барлық пайданушы ақпараты PostgreSQL реялизациялық деректер қорымен сақталуы керек. Құпия сөзді bcrypt көмегімен хэштеу керек. Сондай-ақ пайдалушыны құруға сұраныс жасалған клиенттік IP мекенжайы бойынша елді анықтау мүмкіндігін іске асыру өажет. Алынған дерекқорда сақталуы керек.

Стек: Python, FastAPI, SQLAlchemy, Pydantic, Alembic.

Деректерді тексеру: Pydantic модельдерінің көмегімен кірістерді белгілі бір форматтарға сәйкестігін тексеру (мысалы, дұрыс email, телефон нөмірі).

Логинг: жазбаларды жасау, оқу, жаңарту және жою, қателер мен ерекшеліктер сияқты қолданбадағы барлық маңызды әрекеттерді тіркеу.

Қателер мен ерекшеліктерді өңдеу: қолданба жұмыс істеп тұрған кезде пайда болатын ақпараттық қате туралы хабарламаларды өңдеу және клиентке қайтарады.

Көші-қон механизмі: дерекқор схемасының өзгеруін және оның көші-қонын басқару үшін Alembic пайдалану.

Байланыс ХАТТАМАСЫ-HTTP, REST

UI қажет емес. Unit тестілері бар

Clean code:

Стандарттарды сақтау: Python үшін PEP 8.

Модульдер мен сыныптарға бөлу: тапсырыс беру және құрылымдау үшін.

Ақпараттық атау: айнымалылар, функциялар, сыныптар.

Қайталанудан аулақ болу: Жалпы функционалдылықты бөлектеу.

Пікірлер: тривиальды емес тармақтарды түсіндіру.

Қосымша:

Git және GitHub пайдалану: нұсқаларды басқару үшін барлық кодты GitHub-қа жүктеу керек.

Құрылымдық міндеттемелер: өзгерістерді нақты бақылау үшін Conventional Commits форматына сәйкес құрылымдалуы керек.

Жұмыстың жаңалығы.

Деректермен жұмыс жасау, өңдеу, оқу және өшіру. Деректерді визуалдау.

Бұл операцияларды пайдалану арқылы мәліметтердің бөлімдерінің көптеген қатарларын жеке көруге мүмкіндік береді. Деректерді деректер қорына салған ақпараттар қосымша валидациядан өтеді. Пайдаланушының айпи адресін қағып алып алып қай қаладан екенін деректер қорына сақталады.

«Пайдаланушы ақпаратын сақтауға арналған қолданба» жобаның бағдарламалық құрлымы.

```
1 from fastapi import FastAPI
2
3 app = FastAPI()
4
5
6 @app.get("/")
7 async def root():
8     return {"message": "Hello World"}
9
10 new *
```

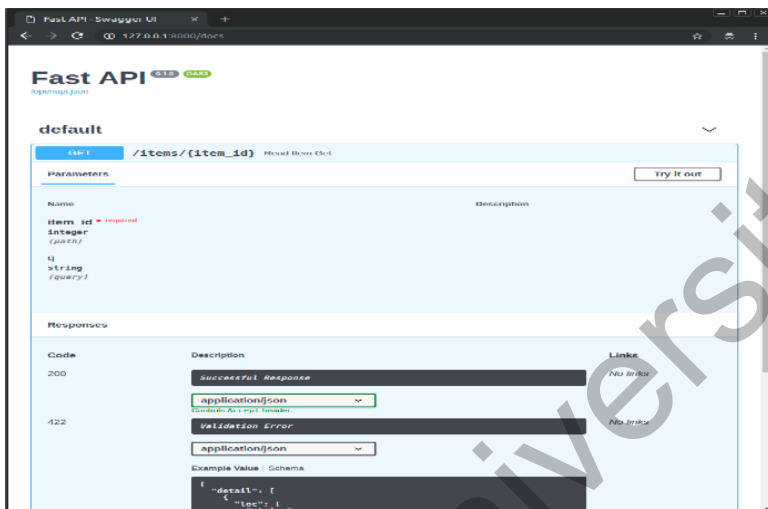
Бастапқы бетке бағдарлама кодын жазылу үлгісі

```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

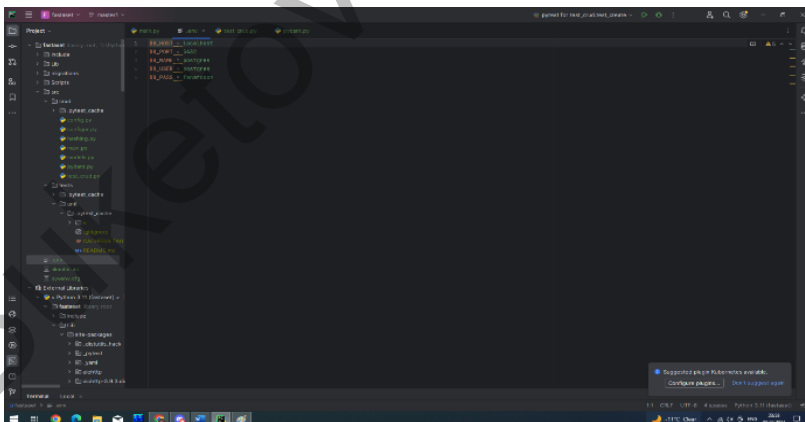
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

(fastaset) PS D:\Python1\ProjectFastAPI\fastaset> uvicorn main:app --reload
```

Басты бет көрінісі



Деректер қорын қосу үшін .env файл ашып оған деректер қорының деректерін жазу қажет, .env файл керегі ертең гитхабқа салған кезде басқа адамдар біздің деректер қорына кіре алмауға және жеке қосымша ақпараттарды көре алмау үшін алған.



Pydant.py ашылу

```
from typing import List
from pydantic import BaseModel, validator

class UserCreate(BaseModel):
    """
    Модель для создания пользователя
    """
    name: str
    email: str
    password: str

    @validator("password")
    def password_must_be_longer_than_8_characters(cls, v):
        """
        Проверка, что пароль длиннее 8 символов
        """
        if len(v) < 8:
            raise ValueError("Пароль должен быть длиннее 8 символов")
        return v
```

Pydantic.py файл ашып оған ақпараттарды салып, Class UserCreate (BaseModel) -біздің болашақ деректеріміз. @validator(“қандай ақпаратты тексеру керек”) және функция ашып керекті функционал жазамыз.

```
from typing import List
from pydantic import BaseModel, validator

class UserCreate(BaseModel):
    """
    Модель для создания пользователя
    """
    name: str
    email: str
    password: str

    @validator("password")
    def password_must_be_longer_than_8_characters(cls, v):
        """
        Проверка, что пароль длиннее 8 символов
        """
        if len(v) < 8:
            raise ValueError("Пароль должен быть длиннее 8 символов")
        return v
```

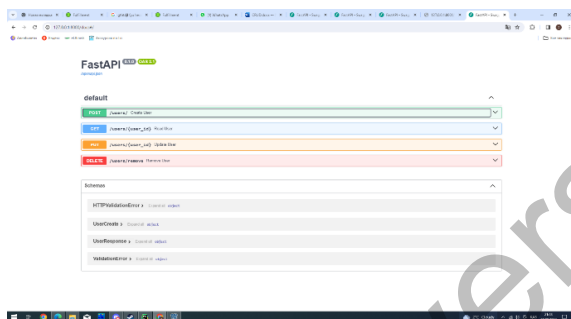
Main.py біздің ең басты файылымыз біз оған crud еңгізіміз

```
from typing import List
from pydantic import BaseModel, validator

class UserCreate(BaseModel):
    """
    Модель для создания пользователя
    """
    name: str
    email: str
    password: str

    @validator("password")
    def password_must_be_longer_than_8_characters(cls, v):
        """
        Проверка, что пароль длиннее 8 символов
        """
        if len(v) < 8:
            raise ValueError("Пароль должен быть длиннее 8 символов")
        return v
```

Бағдарламалық жасақтаманы әзірлеудегі рөлі деректерді басқару мен өңдеудің қарапайым және құрылымдық әдісін қамтамасыз ету болып табылады.



Тест



Қателермен жұмыс

Қорытынды, «Пайдаланушы ақпаратын сақтауға арналған қолданба» жобасы пайдалану әзірлеушілердің қолданба пайдаланатын деректерді өңдеу жолын жеңілдетеді. Жасау, оқу, жаңарту және жою әрекеттерін қосу арқылы әзірлеушілер деректерді оңай жасай алады, оқи алады, жаңартады және жоя алады. Бұл код стандартталған және қызмет көрсету оңай дегенді білдіреді, пайдалану арқылы әзірлеушілер уақытты үнемдей алады, сонымен қатар кодты қарапайым және стандартталған етеді.