8-тәжірибелік тапсырма: Әзірлеу үшін технологиялық қатарды (Tech Stack) таңдау

Міндеті: Жобаның техникалық және бизнес талаптарына сүйене отырып, ең тиімді технологиялық қатарды таңдау және оны негіздеу процесін толыққанды бағалау.

Мақсаты: Өнімнің ерекшеліктерін, масштабталу мүмкіндігін, әзірлеу жылдамдығын және команданың біліктілігін ескере отырып, технологияларды (бағдарламалау тілі, фреймворк, дерекқор) саналы түрде таңдау дағдысын қалыптастыру.

Жалпы сипаттама:
Бұл тапсырмада сіз бұрынғы тапсырмаларда анықталған өнім (мысалы, "Бұлттық деректерді шифрлау құралы") үшін технологиялық шешімдерді таңдайсыз. Сізге нарықтағы кем дегенде үш түрлі технологиялық қатарды қарастырып, оларды алдын ала белгіленген критерийлер бойынша салыстырып, талдау жасау қажет. Соңында, жоба үшін ең оңтайлы деп тапқан стекке тоқталып, өз таңдауыңызды толық негіздеп беруіңіз керек.

Орындау қадамдары:

1. **Жоба талаптарын қысқаша сипаттау:** Алдымен, өнімнің негізгі функционалын (мысалы, Google Drive-пен интеграция, файлдарды шифрлау, қарапайым интерфейс) және негізгі техникалық емес талаптарды (мысалы, MVP-ді тез шығару, болашақта масштабтау мүмкіндігі) тізіп шығыңыз.

2. **Әлеуетті технологиялық қатарларды анықтау:** Жобаңызға сәйкес келуі мүмкін 3 технологиялық қатарды таңдаңыз. Мысалы:
   - **1-нұсқа:** Python (Flask/Django) + SQLite + Google Drive API
   - **2-нұсқа:** JavaScript (Node.js + Express) + MongoDB + Google Drive API
   - **3-нұсқа:** Java (Spring Boot) + PostgreSQL + Google Drive API

3. **Салыстыру критерийлерін таңдау:** Технологиялық қатарларды бағалау үшін кемінде 3-4 маңызды критерийді анықтаңыз. Мысалы:
   - **Әзірлеу жылдамдығы:** MVP-ді қаншалықты тез жасап шығуға болады?
   - **Масштабталуы (Scalability):** Пайдаланушылар саны өскенде жүйе қаншалықты тұрақты жұмыс істейді?
   - **Қоғамдастық және қолдау (Community & Support):** Проблема туындағанда дайын шешімдер, құжаттама немесе көмек табу қаншалықты оңай?
   - **Өнімділік (Performance):** Жүйенің жұмыс істеу жылдамдығы мен ресурстарды тұтынуы.

- **Интеграция мүмкіндіктері:** Google Drive API сияқты сыртқы сервистермен жұмыс істеудің ыңғайлылығы.

4. **Салыстырмалы талдау кестесін жасау:** Таңдалған стектерді критерийлер бойынша салыстыратын кесте құрыңыз. Әр ұяшыққа қысқаша түсініктеме беріңіз.

5. **Соңғы таңдауды негіздеу:** Кестедегі талдауға сүйеніп, жобаңыз үшін ең тиімді технологиялық қатарды таңдаңыз. Неліктен басқа нұсқалардан бас тартқаныңызды және бұл таңдау жобаның мақсаттарына қалай сәйкес келетінін егжей-тегжейлі түсіндіріңіз.

Күтілетін нәтиже (Шешімнің мысалы):

1. Жобаның негізгі талаптары:

- Негізгі мақсат: MVP-ді 4-6 апта ішінде әзірлеу.
- Негізгі функция: Google Drive-тағы файлдарды автоматты шифрлау.
- Техникалық талап: Google Drive API-мен оңай интеграция.
- Болашаққа болжам: Пайдаланушылар саны артқанда, жүйенің тұрақты жұмыс істеуі.

2. Салыстырмалы талдау кестесі:

Критерий	Python + Flask	Node.js + Express	Java + Spring Boot
Әзірлеу жылдамдығы	**Өте жоғары.** Қарапайым синтаксис пен Flask-тың минимализмі MVP-ді тез құруға мүмкіндік береді.	**Жоғары.** Асинхронды табиғаты мен NPM экожүйесі жылдам прототиптеуге жақсы.	**Орташа.** Код көлемі көп, бірақ Spring экожүйесі көптеген дайын шешім ұсынады.
Масштабталуы	**Жақсы.** Дұрыс архитектурамен жақсы масштабталады, бірақ жоғары жүктемеде Node.js-тен баяу болуы мүмкін.	**Өте жақсы.** Оқиғаға негізделген архитектурасы жоғары жүктемелі I/O операцияларына өте тиімді.	**Өте жақсы.** Көп ағынды (multi-threaded) ортада үлкен, күрделі жүйелер үшін өте сенімді.
Қоғамдастық және қолдау	**Өте үлкен.** Data Science, веб-әзірлеу саласында көптеген кітапханалар мен белсенді қоғамдастық бар.	**Өте үлкен.** Әсіресе веб-әзірлеуде ең танымал экожүйелердің бірі. NPM-де миллиондаған пакет бар.	**Үлкен.** Enterprise-деңгейдегі жүйелерде кеңінен қолданылады, өте тұрақты және жақсы құжатталған.

3. Негізделген таңдау:

Жобаның негізгі мақсаты – MVP-ді ең қысқа мерзімде нарыққа шығару. Осыған сүйене отырып, **Python + Flask** технологиялық қатары таңдалды.

Негіздеме:

- **Жылдамдық:** Python-ның қарапайымдылығы мен Flask-тың жеңілдігі әзірлеу уақытын айтарлықтай қысқартады.
- **Интеграция:** Google Drive API-мен жұмыс істеуге арналған Google-дың ресми және өте ыңғайлы Python кітапханасы бар.
- **Масштабталу:** MVP кезеңі үшін Flask-тың өнімділігі толықтай жеткілікті. Болашақта жүйе күрделенсе, Django-ға көшуге немесе микросервистік архитектураны қолдануға болады.
- **Бас тарту себептері:** Node.js жылдам болғанымен, команданың Python-да тәжірибесі мол. Java + Spring Boot өте қуатты, бірақ MVP үшін тым күрделі және әзірлеу уақытын ұзартады.