

План за работа на проектот

Проектот претставува мобилна апликација за управување и преглед на информации за летови од аеродроми, каде што податоците ќе се преземаат од надворешен API. Целта е да се овозможи ефикасно прикажување, пребарување и филтрирање на летови, како и генерирање на резервации преку апликацијата.

Траење на проектот: 2 месеци

Тим од два члена: Елена Марковска (216009) и Марија Вржовска (211073)

Улоги:

- Елена – full-stack
- Марија – backend и chatbot интеграција

Недела	Активност	Време	Улога
1	Дефинирање цели и функционалности	1 недела	Backend & Frontend
2	Избор и интеграција на API	1 недела	Backend
3 - 5	Развој на Backend (API + chatbot)	3 недели	Backend
5 - 7	Развој на Frontend (Flutter + chatbot UI)	3 недели	Frontend
8	Финално тестирање и оптимизација	1 недела	Заедно

1. Цел и функционалности

Апликацијата ќе овозможува:

- Приказ на летови (број на лет, дестинација, време на поаѓање, време на пристигнување, статус, авионска компанија) преземени од API.
- Пребарување и филтрирање на летови по аеродром, компанија, време и други параметри.
- Генерирање резервации или примери на „букирање“ за демонстрација на апликацијата.
- Приказ на податоците преку Flutter апликација.
- Интерактивен чатбот асистент кој нуди:
 1. Сугестии за ефтини или најбрзи летови

2. Предлози за алтернативни датуми или блиски аеродроми
3. Помош при користење на апликацијата

2. Избор и интеграција на API

- **Избор на API:** Бесплатни или freemium API за летови и аеродроми (на пр. AviationStack, AirLabs, AirportsAPI).
- **API повици:** REST методи (GET) за прибирање на листа на достапни летови и аеродроми.
- **Обработка на податоците:** Backend делот ќе ја превзема информацијата од API-то, ќе ја филтрира и форматира пред да ја испрати до фронтендот.
- **Чатбот поддршка:** Користење на API податоците за да се нудат персонализирани сугестии при едноставен rule-based или keyword-based алгоритам.

3. Backend дел

Backend делот ќе биде изработен со **Java Spring Boot**:

- **Controller слој** – дефинира REST endpoints (на пр. /flights, /airports) кои комуницираат со API-то.
- **Service слој** – ја содржи логиката за повикување на API-то, трансформација и валидација на податоците.
- **HTTP методи:**
 - GET – прикажување на летови и аеродроми од API
 - POST – можност за креирање примери на резервации (локално во апликацијата)
- **Обработка на грешки:** Проверка на response кодови од API, timeout, retry логика.

4. Frontend дел

Frontend делот ќе биде изработен со **Flutter** или **React/JSP**, и ќе овозможува:

- Приказ на листа со летови, аеродроми и достапни опции.
- Филтрирање и пребарување според параметри (дестинација, време, компанија).
- Генерирање резиме или receipt за „резервации“.
- Комуникација со backend-от преку REST API endpoints, кои всушност користат надворешно API за податоците.

- Chatbot UI: Chat-интерфејс со едноставен input field кој прикажува препорачани летови директно во разговорот

5. Тестирање и проверка

- Тестирање на REST endpoints со **Postman** или **Insomnia**.
- Проверка дали backend-от правилно ја презема и прикажува информацијата од API-то.
- Проверка на филтрирање, пребарување и резиме генерација.
- Валидација на внесените податоци и error handling.
- Чатбот логика

6. Технологии и алатки

Backend:

- Java 17+
- Spring Boot (Web, REST)
- Spring WebClient или RestTemplate (за повикување на API)

Frontend:

- Flutter (за мобилна апликација) или React / JSP (за веб-интерфејс)

Дополнителни алатки:

- Postman – за тестирање на API повици
- GitHub – верзионирање
- IntelliJ IDEA – развој на backend
- Visual Studio Code – развој на frontend