Deployment Plan

Project Name: TerraSense

List of Contributors:

- Ahmet Yusuf ŞAHİN, Backend Developer
- Furkan YAHŞİ, UI Designer, Frontend Developer
- Mehmet Ali USLU, API Integrations, Frontend Developer
- Yusuf Can BAHADIRLIOĞLU, Test Specialist, Quality Assurance

Table of Contents

1. Task Matrix

Task Description	Ahmet Y. Şahin	Furkan Yahşi	Mehmet Ali Uslu	Yusuf Can Bahadırlıoğlu
Deployment Overview		**X**		
Deployment Process	**X**	**X**		
Configuration Plan		**X**	**X**	**X**

2. Deployment Plan

2.1 Deployment Overview

For the demo, the project was deployed in a hybrid environment where the backend is hosted locally (Flask on http://127.0.0.1:5000) and the Flutter application is run in web mode (Chrome) during the presentation.

- **Tools Used:**
  - **Backend:** Python/Flask, deployed on local machine using a virtual environment.
  - **Frontend:** Flutter Web (running via `flutter run -d chrome`), using the `percent_indicator, flutter_markdown` and some other Flutter packages.

- **Environment:**
  - Development: Localhost on Windows with Developer Mode enabled.
  - Communication between frontend and backend is managed via RESTful API calls.

2.2 Deployment Process

1. **Backend Deployment:**
   a. Activate the Python virtual environment.
   b. Ensure all dependencies are installed using `pip install -r requirements.txt`.
   c. Set environment variables (e.g., `SECRET_KEY`) via a `.env` file or system environment.
   d. Run the backend using `python run.py` ensuring the Flask server is running on http://127.0.0.1:5000.
2. **Frontend Deployment:**
   a. In the Flutter project directory, run `flutter pub get` to fetch dependencies.
   b. Enable Developer Mode in the operating system for symlink support.
   c. Launch the Flutter application in Chrome using `flutter run -d chrome`.
3. **Integration Testing:**
   a. Verify that API calls from Flutter (using URLs like http://127.0.0.1:5000/auth/kayit and /auth/giris) are successful.
   b. Use Postman or browser network tools to confirm endpoints are reachable.

2.3 Configuration Plan

- **Backend Configurations:**
  - `.env` file includes `SECRET_KEY` and `DATABASE_URL`.

    - CORS is enabled via `flask_cors`.
- **Frontend Configurations:**
    - `ApiService.dart` is configured to use `baseUrl='http://127.0.0.1:5000'`.
    - Debug mode is active for development and testing.
- **Version Control:**
    - All changes are committed to GitHub with clear commit messages.
    - The project uses a standard branch structure and the latest stable Flutter SDK.