

Team 11 - Software Engineering Project

This repository contains the entire application (backend and frontend) for the Team 11 Software Engineering Project. The backend is built using Flask and integrates with Firebase and Google Cloud's Gemini API (Generative Language API) to implement all the user stories, while the frontend is built separately using ReactJS and interacts with the backend.

This README provides detailed, step-by-step instructions for setting up and running the full application, as well as configuring Firebase and the Gemini API.

Prerequisites

Ensure you have the following installed on your machine:

- **Python Version 3.11 or lower (If you have a Python version higher than 3.11, uninstall it before installing Python 3.11 or lower)**
 - A terminal (Command Prompt, PowerShell, or Git Bash)
 - An internet connection
-

How to Use the Application

You have two options to run and use the application:

1. Use the Deployed Application
2. Manually Set Up and Run Backend and Frontend

Option 1: Use the Deployed Application

The easiest way to access the application is through the fully deployed version available at:

🔗 <https://team-11-frontend-v1-457986151866.us-central1.run.app/>

Simply open the link in your browser - no setup required.

Firebase Firestore Setup (Required for Option 2)

1. Create a Firebase Project:

- Go to [Firebase Console](#).
- Click "**Create a project**" and follow the on-screen instructions to create a new project.

2. Create a Firestore Database:

- Go to **Project Overview** → **Cloud Firestore** → **Create Database** and follow the on-screen instructions to create a new Firestore Database (Test Mode).

3. Update .env with Firebase Credentials:

- In the Firebase Console, go to **Project Settings (the Settings Icon next to Project Overview)** → **Service accounts**.
- Under 'Admin SDK configuration snippet' select Python, click on "**Generate new private key**".
- Copy the following fields from the downloaded file and add them to your .env file
(**IMPORTANT: Rename .env.example to .env for this, if not already done**):

```
FIREBASE_PROJECT_ID=your_firebase_project_id  
FIREBASE_PRIVATE_KEY=your_firebase_private_key  
FIREBASE_CLIENT_EMAIL=your_firebase_client_email
```

Google Cloud Gemini API & Firestore API Setup (Required for Option 2)

1. Select Your Project:

- Visit [Google Cloud Console](#).
- Select the project you created in Firebase using the Project ID (located beside the Google Cloud icon).

2. Enable the Google Cloud Firestore API:

- Go to the [Google Cloud Firestore API page](#).
- Click "**Enable**".

3. Enable the Gemini API:

- Go to the [Gemini API page](#).
- Click "**Enable**".

4. Create Credentials:

- Go to **APIs & Services** → **Credentials**.
- Click "**Create Credentials**" and select "**API Key**".
- Copy the created API key.

5. Update Environment Variables:

- In the .env file, add:

```
GOOGLE_API_KEY=your_google_gemini_api_key
```

Option 2: Manually Set Up Backend and Frontend

If you prefer, you can manually set up both the backend and frontend by following these steps: (Make sure the .env file is fully ready as per the instructions before moving forward)

Setting Up and Running the Backend

1. Clone or Copy the Project Folder:

- Ensure you have the complete project folder.

2. Create and Activate a Virtual Environment:

- Navigate to the backend folder:

```
cd backend
```

- Create a virtual environment:

```
python -m venv .venv
```

- Activate the virtual environment:

```
.venv\Scripts\activate
```

3. Install Dependencies:

- Install required packages:

```
pip install -r requirements.txt
```

4. Run the Backend Application:

- Start the Flask server:

```
python run.py
```

Setting Up and Running the Frontend

1. Navigate to the frontend folder:

```
cd frontend
```

2. Install dependencies using Yarn:

```
yarn
```

3. Start the frontend application:

```
yarn dev
```

4. Access the Application:

- Open <http://localhost:5173/> in your browser to use the full application.
 - **Note:** The backend must be up and running before starting the frontend.
-