Google Developer Groups

VIT BHOPAL

Name: Divyansh

Registration No.: 25BCE11162

Domain: Web Development

GitHub Repository: https://github.com/divyanshsinha305-sudo/GDGC_Members_Profile_Showcase.git

SUMMARY

This project is a compact member-profile showcase built for GDGC. The frontend is a simple and fast single-page layout designed using HTML, CSS, and JavaScript. The interface uses a clean glass-blur style inspired by modern iOS designs to keep the experience smooth and visually appealing.

The backend is a lightweight Python Flask API that serves structured member information through the `/members` and `/members/:id` endpoints. Since I only started doing practical backend projects a few months ago, I am not fully experienced with backend development or deployment yet. However, for this task I learned the essential parts, understood how basic APIs work, and implemented the required functionality.

I built the entire UI and wrote the JavaScript logic myself. I also used ChatGPT for learning backend concepts, correcting errors, and understanding API behaviour, especially because this was my first proper backend-connected project. Even then, I made sure I personally wrote and tested the code, integrated it with my frontend, and completed the full flow end-to-end.

The final app includes search, filtering, dark mode with saved preferences, debounced interactions, and smooth transitions. All components were developed and tested by me using the technologies listed in the document.

Technologies & Skills Used

Frontend

- HTML, CSS, JavaScript
- Responsive layouts using Flexbox and Grid
- Basic animations and transitions
- Understanding of UI/UX fundamentals

Backend

- Python (Flask) – small REST API
- JSON responses

Database

- JSON data file for this project
- Basic familiarity with MySQL

Tools & Platforms

- VS Code
- GitHub (repo + GitHub Pages for frontend hosting)
- Thunder Client / Postman for API testing

Project 1

GDGC Member Profile Showcase

(Main Submission)

Project Overview

A small web application that displays GDGC member profiles in an interactive grid. Users can search, filter by role/skills/location,

change theme modes, and view profile details. Member data is fetched from the backend API.

My Approach

- Keep the UI light, modern, and friendly
- Build around the required features: splash screen, card grid, filters, persistent theme, and API structure
- Make the app fast to load and easy to use on all screen sizes

.

Key Technical Decisions

- **Debounced searching :** prevents unnecessary repeated fetch calls
- **Server-side filtering:** backend receives query parameters and returns only relevant members
- **Client caching:** browsers naturally cache API responses, improving speed
- **Lazy image loading:** better initial render time
- **Clean, modular code:** separate JS for API, UI functions, and components

Project 2

Weather App

(Previous Work)

Description

A simple weather application that fetches data from a public API and shows the current weather along with a short forecast.

What I Focused On

- Understanding how APIs work (fetch requests)
- Handling invalid API responses gracefully
- Converting raw data into readable weather details
- Clean and minimal UI

Project 3

Personal Portfolio

(Previous Work)

Description

A static personal portfolio created using HTML, CSS, and JavaScript. It highlights projects, skills, and contact information.

Core Ideas

- Responsive layout suitable for all screens
- Light/Dark mode toggle
- Clear CV-style structure
- Deployed on GitHub Pages

Deliverables Completed for This GDGC Task

- Splash screen with animation and auto-redirect
- Member cards showing : name, photo, skills, role, and short bio
- Search bar + filters for roles, skills, and location
- Light/Dark theme switch with local Storage persistence
- Functional endpoints:
  - `/members`
  - `/members/:id`
- Loading indicators and error handling
- Working deployment links for frontend and backend
- Screenshots and commit history included in the final repository

How to Run the Project Locally

1. Clone the Repository

```
git clone https://github.coM/YourUser/GDGC.git
```

2. Backend Setup (Flask)

Windows:
```
venv\Scripts\activate
```

Install dependencies:

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

Run the server:

```
python app.py
```

Backend runs at **http://127.0.0.1:5000**
Endpoints: `/members`, `/members/<id>`

3. Frontend Setup

VS Code's Live Server

Set:

```
API_BASE_URL = "http://127.0.0.1:5000"
```

for local testing.

THANK YOU !

Team : GDGC