

Project Documentation

Project Title:

NAP – Neighbour Assistant Platform

Developer Info:

- **Name:** Siddaarth P T
- **Email:** siddaarth28@gmail.com
- **College:** SRM Easwari Engineering College
- **Branch:** Mechanical Engineering

 **Abstract:**

In today's fast-paced world, locating reliable nearby services quickly and efficiently is essential, especially in urban environments where demand for local services like plumbing, tutoring, and housekeeping is high. The **Neighbour Assistant Platform (NAP)** is a web-based application designed to address this need by providing users with a convenient and intuitive platform to search for nearby service providers based on their current geographic location or manually entered location.

The platform leverages the power of **Google Maps Search API via SerpAPI** to fetch real-time data about local services around the user. What makes NAP stand out is its simplicity and privacy-conscious approach — it allows users to access the service without requiring a user account or password. Instead, minimal but essential user information such as name, email ID, and mobile number is collected through a mandatory form to keep a basic record of users accessing the platform. This information is securely stored in a backend database using Django models.

Key features of the platform include:

- **Live Location Access:** Automatically fetches the user's geographic coordinates using browser-based geolocation.
- **Manual Location Input:** Users can enter their city or location if GPS access is denied or unavailable.
- **Service Type Search:** Users can search for services like “electrician,” “doctor,” or “tuition” with flexibility.
- **Dark and Light Mode UI:** A toggle switch allows users to choose a theme that suits their preference.
- **Responsive UI:** Built using Bootstrap 5.3 for modern design and seamless experience across devices.

The backend is powered by the **Django web framework**, with **data stored in SQLite** during development. The platform was designed as a student project and aims to demonstrate how location-based services can be integrated with real-world APIs to create functional and scalable applications. As an added advantage, NAP does not store or expose any sensitive data publicly, keeping the platform safe and GDPR-friendly.

Ultimately, the Neighbour Assistant Platform showcases how simple technologies can be combined to solve a common, real-life problem while emphasizing user-friendliness, privacy, and practical implementation. The project is ideal for further extension into a full-fledged mobile app or integration with booking and feedback systems in future phases.

Objectives:

- To build a user-friendly platform for discovering local services.
 - To simplify location-based service search using live geolocation.
 - To store minimal user data for tracking user interest and basic analytics.
 - To integrate third-party APIs (Serp API) to extract real-time data from Google Maps.
-

Technologies Used:

Component	Technology
Backend	Django (Python)
Frontend	HTML, Bootstrap 5.3
API Integration	SerpAPI (Google Maps)
Styling	Bootstrap (with Dark/Light mode)
Database	SQLite (default Django DB)
Hosting	Render.com / PythonAnywhere (for deployment)

System Modules:

1. User Input Module

- Takes basic user data: **Name, Email, Mobile Number**
- Mandatory form submission before accessing search.

2. Geolocation Fetch Module

- Uses JavaScript navigator.geolocation to get current latitude and longitude.

- Falls back to manually entered location if GPS is disabled.

3. Search Functionality

- Users can enter a service keyword (e.g., “plumber”, “tutor”).
- Sends query + coordinates to **SerpAPI**.
- Fetches and displays a list of nearby results from Google Maps.

4. Data Storage Module

- Stores user details in the Django model/table (`UserQuery`).
 - Helps track how many users used the platform and when.
-



Functional Workflow:

1. User opens the website.
 2. They see a form asking for name, email, mobile number, and service type.
 3. The browser tries to fetch their GPS coordinates.
 4. On form submission:
 - Data is stored in the database.
 - The query and coordinates are sent to SerpAPI.
 - Results are displayed as cards with service name, address, phone, rating, and website.
 5. Dark/light theme toggling is supported for better accessibility.
-



Database Design:

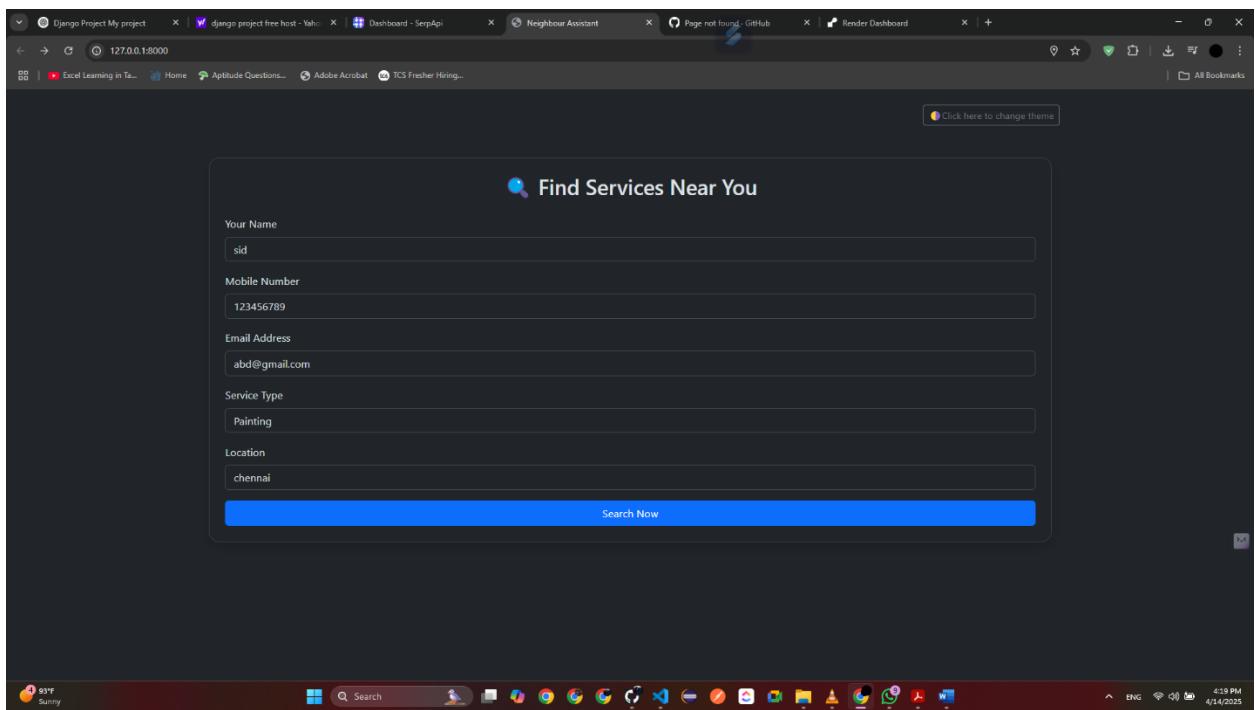
Model: `UserQuery`

Field Name	Type	Description
name	CharField	User's name

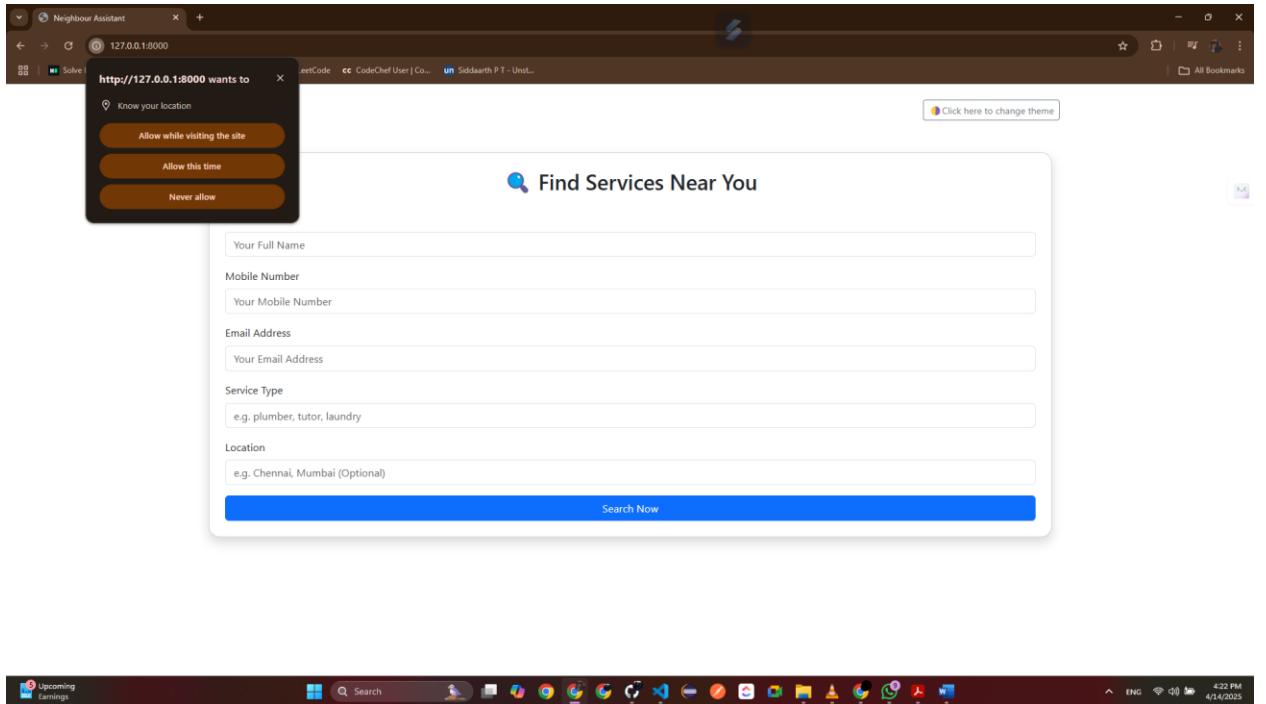
email	EmailField	User's email
phone	CharField	User's mobile number
service_query	CharField	Search keyword
latitude	FloatField	GPS latitude
longitude	FloatField	GPS longitude
created_at	DateTime	Timestamp of submission

Screenshots:

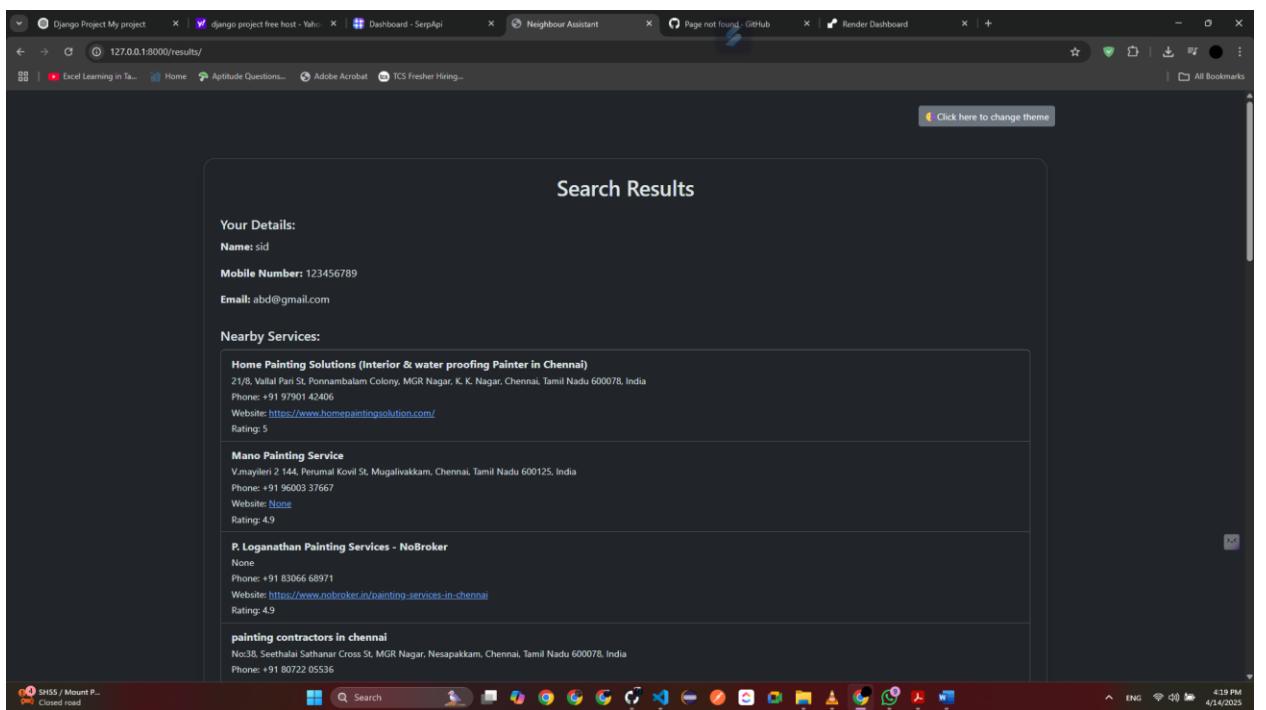
1. Home Page with input form



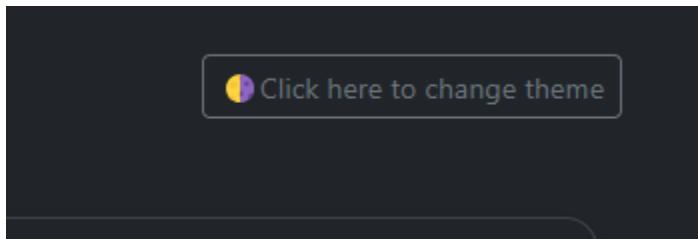
2. Location permission



3. Search results displayed in cards



4. Dark/light mode toggle



Future Enhancements:

- Allow users to review or rate services.
 - Add login/logout for advanced analytics .
 - Add service filters (ratings, distance, etc.).
 - Use Google Maps API for direct map integration.
-

Conclusion

The **Neighbour Assistant Platform (NAP)** effectively demonstrates how modern web technologies can be integrated with third-party APIs to solve real-world problems. Through a user-friendly interface, location-based service search, and minimal data input, the platform allows users to quickly find local service providers without the need for complex registration processes.

By using **Django** for backend development and **Bootstrap 5.3** for frontend design, the project emphasizes responsiveness, scalability, and maintainability. The use of **SerpAPI** to extract real-time Google Maps data further enriches the service search experience, making the application highly relevant and functional.

The inclusion of **dark/light mode switching**, real-time **geolocation**, and a secure data capture mechanism highlights the focus on usability and user experience. The project's simplicity in design combined with practical features makes it a great example of a lightweight yet impactful web application.

This student-level project lays a strong foundation for future enhancements, such as integrating user reviews, service provider profiles, booking features, and even a mobile app version. It proves that even with limited resources and simple tools, real-world solutions can be effectively prototyped and implemented.

References

1. **Django Documentation** – <https://docs.djangoproject.com/>
2. **Bootstrap 5.3 Documentation** – <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
3. **SerpAPI (Google Maps API wrapper)** – <https://serpapi.com/google-maps-api>
4. **HTML5 Geolocation API** – https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API
5. **JavaScript LocalStorage for Theme Switching** – <https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>