

Problem Statement

PS-2: Integrated common Services to Common people.

Unique Idea Brief (Solution)

Here we provide the services like transportation, health, finance, education, government and housing to common people for accessing corresponding websites and support them. Our goal is to provides the users at a free experience in our website and make their job extremely easy and effortless. Our main features are providing map services for users and connecting users with service providers. The main advantages of our solution are that all the people can access the necessary services within their homes by using our websites and we connect the service providers with users after a huge process of verification to avoid fake id. This website is really helpful for aged people and others who have less time to spend in their life.

For an example, if we want to book an appointment through our website, we display the registered doctor's details from that the user can easily find the doctor they need and book the appointment easily and effectively.

In our website, we implemented this as a demo and other service providers will be displayed like this after the successful verification by the team.

Features Offered

- User-friendly interfaces
- Responsive design
- Interactive map service
- Secure authentication
- Email verification
- Team member profiles
- Modern UI or UX
- Admin Notification
- Database Connectivity
- Data Encryption

Processflow

1. Project Initialization

- Define Requirements: Gather all the requirements for the project.
- Set Up Environment: Install necessary software (Node.js, PostgreSQL, etc.).

2. Database Design and Setup

- Design Database Schema: Plan the structure of our database.
- Set Up Database: Create the database and tables in PostgreSQL using psql shell.

3. Backend Development

- Initialize Node.js Project: Create a new Node.js project using npm init.
- Express Set up: Install and configure Express for handling routes.
- Integrate the Nodemailer: Configure Nodemailer for sending emails.
- Connect to Database: Use pg library to connect Node.js to PostgreSQL database.

4. Frontend Development

- Design HTML Pages: Create static HTML files.
- Style with CSS: Apply preferred CSS styles.
- Integrate Handlebars: Use Handlebars (.hbs) for dynamic content rendering.
-

5. Integration and Testing

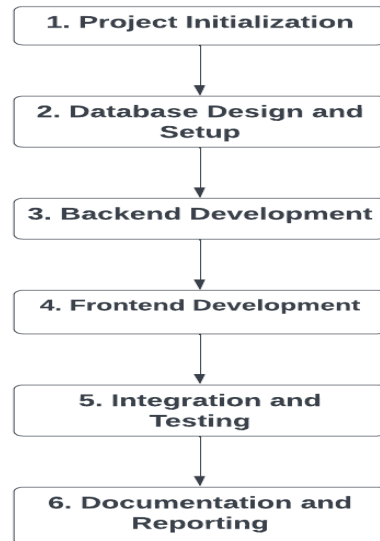
- Form Submission: Ensure forms submit data to backend and handle responses.

- Email Functionality: Verify that emails are sent correctly using Nodemailer.
- Display Data: Ensure data fetched from the database is displayed correctly on the frontend.

6. Documentation and Reporting

- Document Code: Comment and document your code for clarity.
- Prepare Report: Write our project report and template.

Visual Representation



Architecture Diagram

1. Client-Side (Frontend)

- HTML/CSS/JavaScript: Basic structure and styling of the web pages.
- Handlebars (.hbs): Template engine for rendering dynamic content.

2. Server-Side (Backend)

- Node.js: JavaScript runtime environment.
- Express.js: Web framework for handling HTTP requests and routing.
- Nodemailer: For sending emails.

3. Database

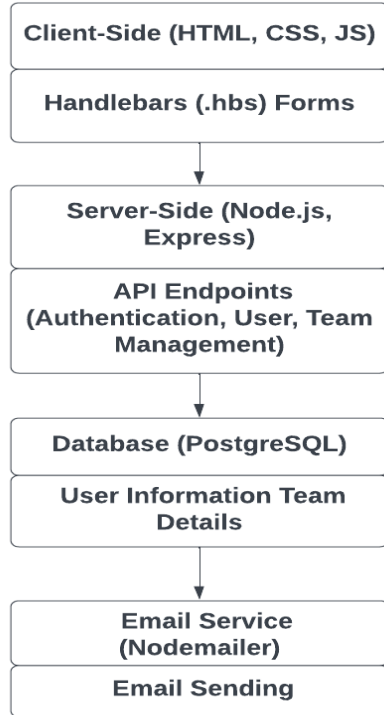
- PostgreSQL: Database for storing information.
- pg Library: Node.js library for connecting and interacting with PostgreSQL.

4. API Endpoints

- Authentication: Endpoints for user registration, login, and email verification.
- User Management: Endpoints for updating user profiles, fetching user details.
- Team Management: Endpoints for fetching and displaying team member details.

5. Integration

- Email Service: Integration with email service providers for sending emails using Nodemailer.
- Static Files: Serving static files such as CSS, images, and HTML.



Technologies used

Node.js: For server-side scripting and building scalable network applications.

Express.js: A web application framework for Node.js to build web applications.

Handlebars: Handlebars compiles templates into JavaScript functions.

Postgresql: A SQL database for storing data.

HTML/CSS: For structuring and styling web pages.

Javascript: For adding interactivity to web pages.

Nodemailer: It's widely used for sending transactional emails, such as verification emails, password resets, and other notifications from web applications.

Team members and contribution:

Team leader: Aadarsh Narayan P.S (Frontend and Backend)

Member 1: Amaldev S.S (Frontend and Backend)

Member 2: Adithya P (Designing)

Member 3: Danil R.A (Designing)

Conclusion

In our project, we explored the development and integration of various services into a Node.js-based website. Our primary goal was to create a user-friendly platform that seamlessly connects with transportation services, health services, finance services, educational services, government services, and housing services. We identified that integrating these services significantly enhances user experience by providing easy access to essential information and some functionalities. The use of Node.js and Express.js proved to be effective in building a scalable and responsive web application. Additionally, incorporating Postgresql for data storage ensured efficient data management.

This project demonstrates the potential of combining various web technologies to create a comprehensive and efficient online platform. Future work should focus on expanding the range of services offered and continuously improving the user interface to meet evolving user needs.

