**Grievance Web Application Documentation**

Submitted my Meenakshi M(Amrita Vishwa Vidyapeetham)

**1. Project Overview**

The Grievance Web Application is a platform that empowers individuals to voice their concerns and grievances directly to the appropriate authorities. The platform ensures that every issue is heard and addressed promptly, fostering a transparent and responsive environment where users can submit, track, and resolve grievances with ease.

**2. Technologies Used**

* **Frontend**:
  + React.js: For building the user interface.
  + React Router: For handling navigation within the app.
  + Axios: For making HTTP requests to the backend.
  + React Icons: For incorporating icons into the interface.
  + CSS: For styling the application.
* **Backend**:
  + Node.js: For running the backend server.
  + Express.js: For building RESTful APIs.
  + MongoDB: For storing user and grievance data.
  + Mongoose: For modeling the application data.
  + JSON Web Tokens (JWT): For user authentication.
* **Tools**:
  + Visual Studio Code: For code editing.
  + Git: For version control.
  + Postman: For API testing.

**3. Pages and Features**

**3.1 Home Page**

* **Description**: The home page provides an introduction to the platform, highlighting its purpose and how it empowers users to voice their concerns.
* **Features**:
  + Navigation links to other parts of the site (Login, Register, Submit Grievance, Admin Dashboard).
  + A search bar for quick access to external articles related to grievances.

**3.2 Login Page**

* **Description**: The login page allows existing users to access their accounts by entering their email and password.
* **Features**:
  + Login form with email and password fields.
  + Options to login via Google or Facebook.
  + Links to the registration page for new users.

**3.3 Register Page**

* **Description**: The registration page enables new users to create an account by providing their name, email, and password.
* **Features**:
  + Registration form with fields for name, email, password, and password confirmation.
  + Options to sign up via Google or Facebook.

**3.4 Submit Grievance Page**

* **Description**: This page allows authenticated users to submit their grievances, providing details like type, title, description, and an option to upload supporting documents.
* **Features**:
  + Form to input grievance details.
  + File upload functionality.

**3.5 Admin Dashboard**

* **Description**: The admin dashboard provides an overview of all grievances submitted by users. Admins can view, track, and update the status of each grievance.
* **Features**:
  + Table listing all grievances with columns for email, type, title, description, status, and action.
  + Buttons to mark grievances as resolved.

**3.6 Profile Page**

* **Description**: The profile page displays a list of grievances submitted by the logged-in user, along with their current status.
* **Features**:
  + List of user’s grievances.
  + Status of each grievance.

**3.7 Navbar with Search Functionality**

* **Description**: The navigation bar appears at the top of every page, providing easy access to the home page, grievance submission page, admin dashboard, and search functionality.
* **Features**:
  + Links to key pages: Home, Submit Grievance, Admin Dashboard, Profile.
  + Search bar for searching related articles on Google Scholar.
  + Logout option for authenticated users.

**4. Setting Up and Running the Project**

**4.1 Prerequisites**

* Node.js installed on your system.
* MongoDB installed and running on your local machine or access to a MongoDB cloud instance.
* Git for version control.
* A code editor like Visual Studio Code.

### 4.2 Cloning the Repository

git clone <https://github.com/meenakshi-m/grievance-web.git>

### 4.3 Setting Up the Backend

1. Navigate to the backend directory: cd backend
2. Install the dependencies: npm install
3. Create a .env file in the root of the backend directory with the following contents: MONGO\_URI=your\_mongodb\_uri JWT\_SECRET=your\_jwt\_secret
4. Start the backend server: node app.js

**4.4 Setting Up the Frontend**

1. Navigate to the frontend directory: cd frontend
2. Install the dependencies : npm install
3. Start the frontend development server: npm start

**4.5 Accessing the Application**

* Open your web browser and go to http://localhost:3000 to view the application.

**Future Enhancements**

* Implementing role-based access control for different user types.
* Adding email notifications for grievance status updates.
* Integrating third-party authentication (Google, Facebook).
* Adding a reporting feature for grievance analytics.

**Grievance Web Application Documentation**

**Submitted by: Meenakshi M**

**Institution**: Amrita Vishwa Vidyapeetham

**1. Project Overview**

The Grievance Web Application is a powerful platform designed to enable users to submit, track, and resolve grievances seamlessly. This system ensures transparency and prompt action on issues reported by users, fostering a responsive environment where concerns are handled efficiently by the appropriate authorities.

**2. Technologies Used**

**Frontend:**

* **React.js**: For building dynamic and interactive user interfaces.
* **React Router**: Manages navigation within the application, ensuring smooth user experience.
* **Axios**: Simplifies HTTP requests and handles communication between the frontend and backend.
* **React Icons**: Adds visually appealing icons to enhance user interaction.
* **CSS**: Custom styling to ensure a clean, user-friendly interface.

**Backend:**

* **Node.js**: Provides a robust runtime environment for server-side scripting.
* **Express.js**: Facilitates the creation of RESTful APIs and handling of HTTP requests.
* **MongoDB**: A NoSQL database for storing user data and grievances efficiently.
* **Mongoose**: An ODM library for MongoDB, simplifying data modeling.
* **JSON Web Tokens (JWT)**: Secures user authentication and session management.

**Tools:**

* **Visual Studio Code**: An efficient code editor with powerful extensions.
* **Git**: Version control system to manage codebase and track changes.
* **Postman**: Used for testing API endpoints and ensuring the backend functions correctly.

**3. Pages and Features**

**3.1 Home Page**

* **Description**: The landing page introduces users to the platform, highlighting its key features and benefits.
* **Features**:
  + Navigation links for easy access to other sections (Login, Register, Submit Grievance, Admin Dashboard).
  + A search bar enabling quick access to external articles related to grievances.

**3.2 Login Page**

* **Description**: This page allows existing users to log in using their credentials.
* **Features**:
  + Secure login form with email and password fields.
  + Options for social login via Google or Facebook.
  + Link to the registration page for new users.

**3.3 Register Page**

* **Description**: Enables new users to sign up by providing necessary details such as name, email, and password.
* **Features**:
  + A user-friendly registration form with input validation.
  + Options to sign up via Google or Facebook.

**3.4 Submit Grievance Page**

* **Description**: Authenticated users can submit grievances, specifying details like type, title, description, and attaching supporting documents.
* **Features**:
  + A comprehensive form for submitting grievances.
  + File upload functionality to attach relevant documents.

**3.5 Admin Dashboard**

* **Description**: Provides administrators with a comprehensive view of all grievances, allowing them to manage and resolve issues.
* **Features**:
  + A dynamic table listing all grievances, with options to mark them as resolved.
  + Real-time updates to the status of grievances.

**3.6 Profile Page**

* **Description**: Displays the grievances submitted by the logged-in user, along with their current status.
* **Features**:
  + A detailed list of the user's grievances.
  + Status indicators showing the progress of each grievance.

**3.7 Navbar with Search Functionality**

* **Description**: The navbar provides quick navigation to all key sections of the site, including a search feature to find relevant external articles.
* **Features**:
  + Links to key pages: Home, Submit Grievance, Admin Dashboard, Profile.
  + Search bar for querying Google Scholar articles related to grievances.
  + Logout option for users to securely exit the platform.

**4. Setting Up and Running the Project**

**4.1 Prerequisites**

* Node.js installed on your system.
* MongoDB running locally or access to a cloud instance.
* Git for version control.
* A code editor like Visual Studio Code.

### 4.2 Cloning the Repository

git clone <https://github.com/meenakshi-m/grievance-web.git>

### 4.3 Setting Up the Backend

1. Navigate to the backend directory: cd backend
2. Install the dependencies: npm install
3. Create a .env file in the root of the backend directory with the following contents: MONGO\_URI=your\_mongodb\_uri JWT\_SECRET=your\_jwt\_secret
4. Start the backend server: npm install

**4.4 Setting Up the Frontend**

1. Navigate to the frontend directory: cd frontend
2. Install the dependencies : npm install
3. Start the frontend development server: npm start

Running the Application

Start the backend server:

* cd backend
* npm start

Start the frontend development server:

* cd frontend
* npm start

**4.5 Accessing the Application**

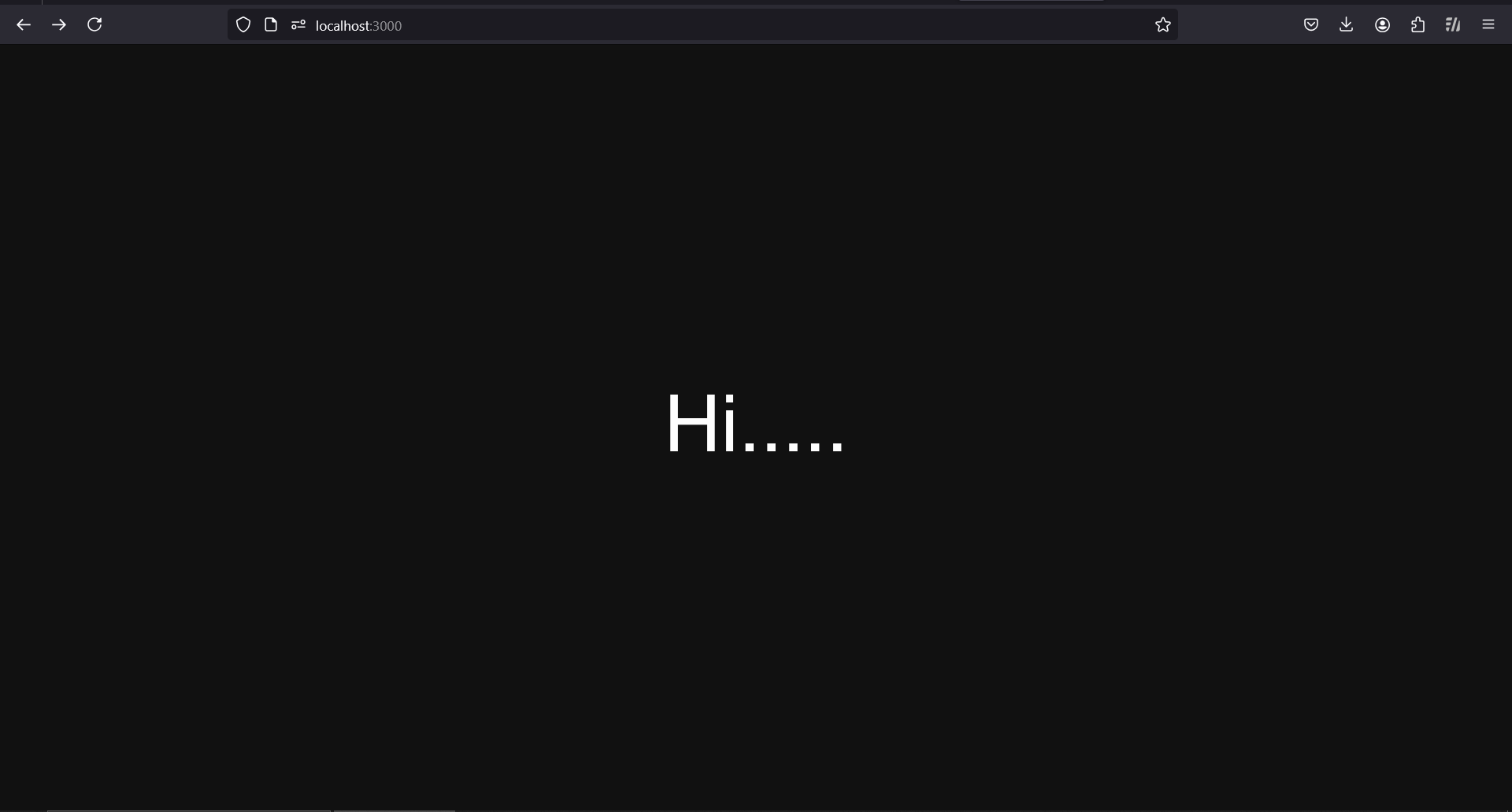
* Open your web browser and go to http://localhost:3000 to start using the application.

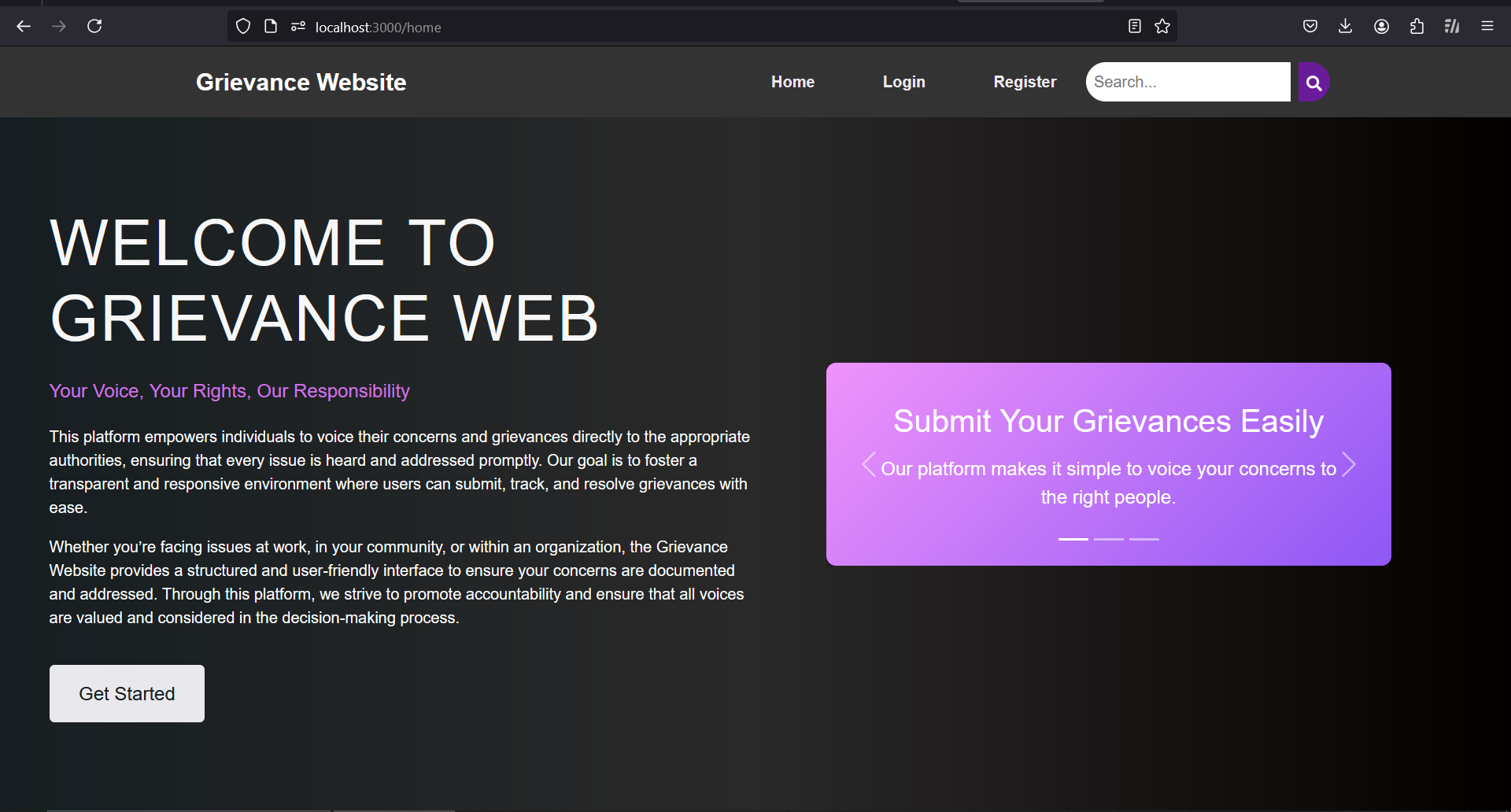
**5. Future Enhancements**

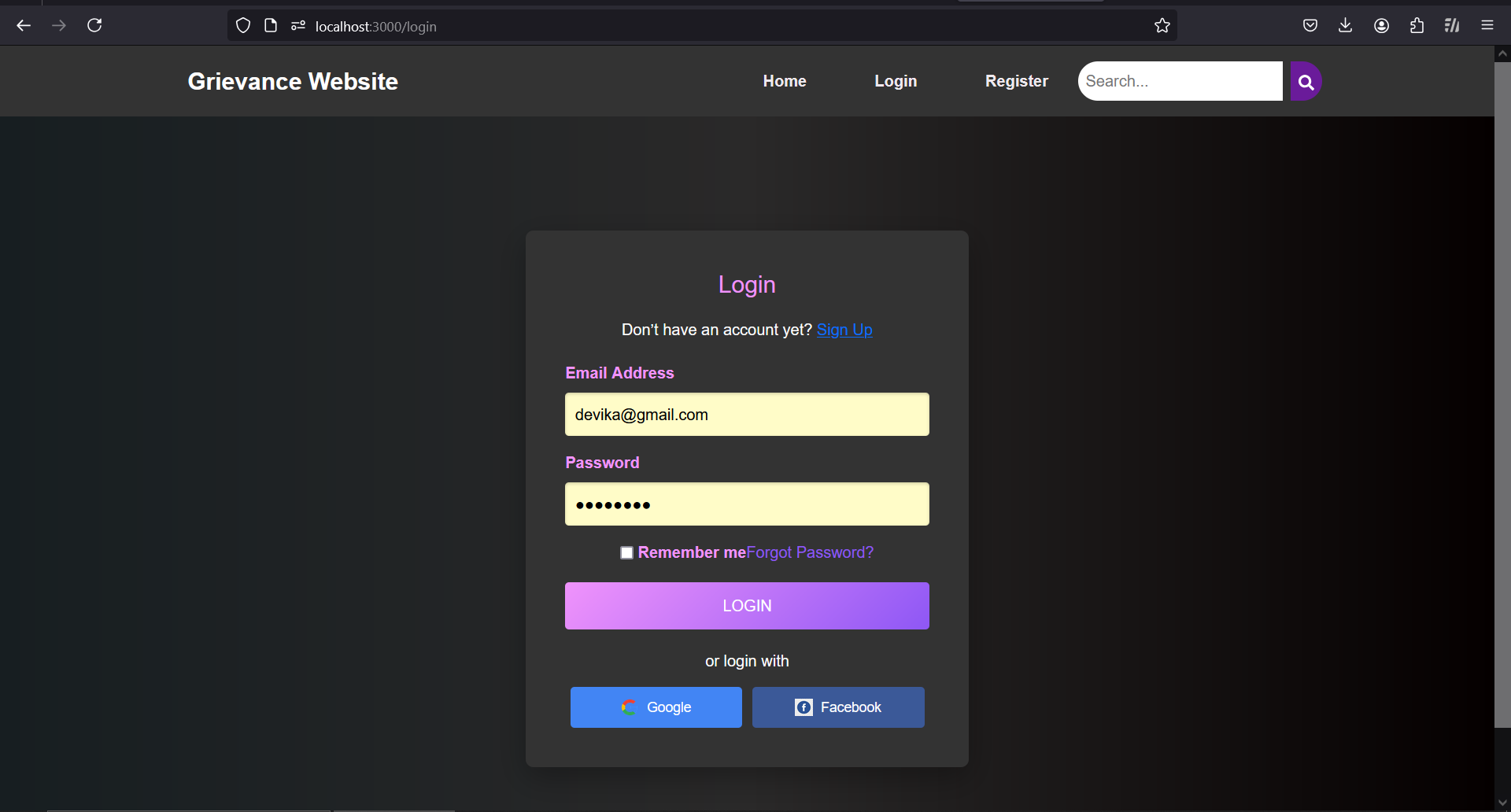
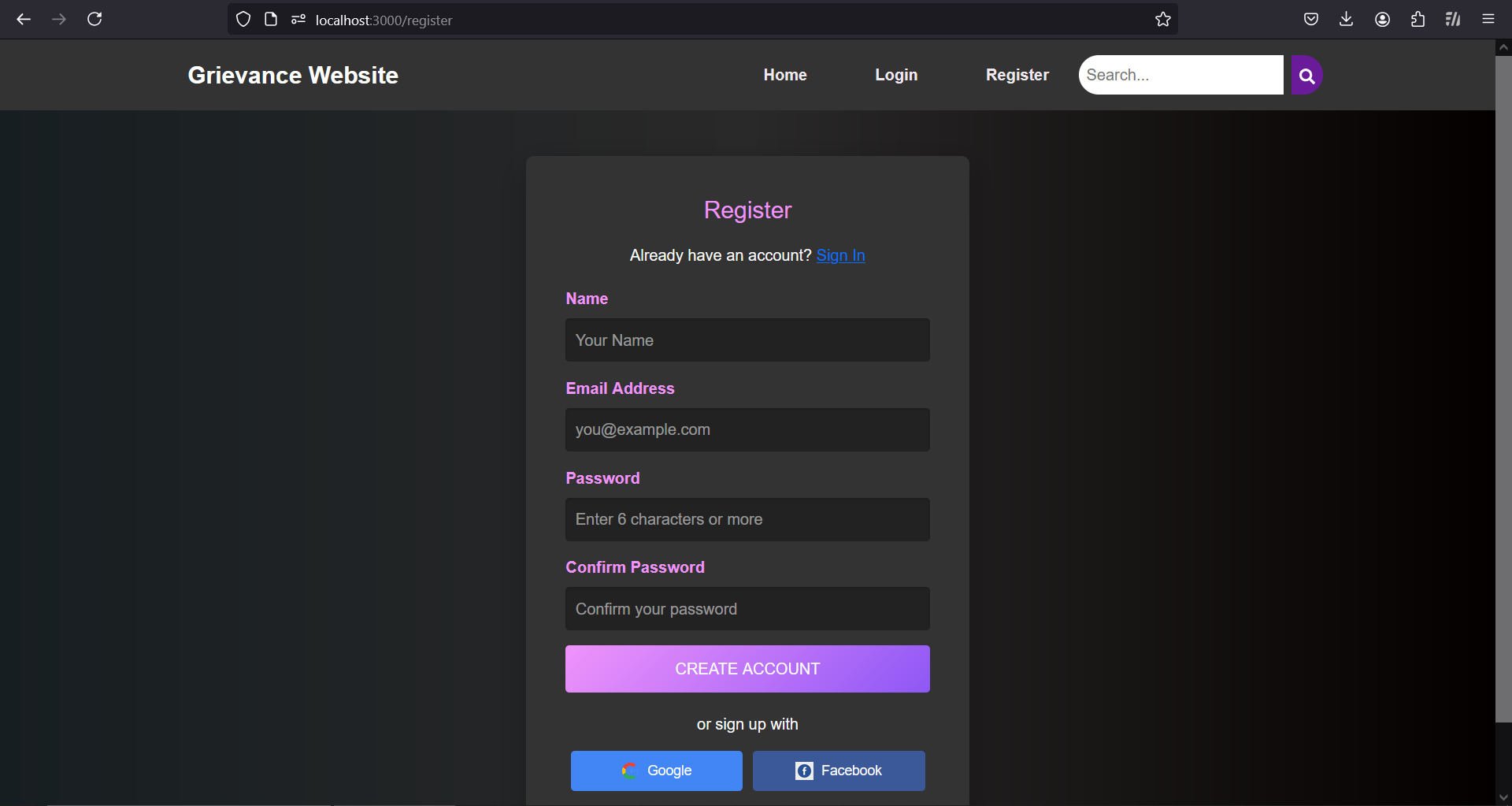
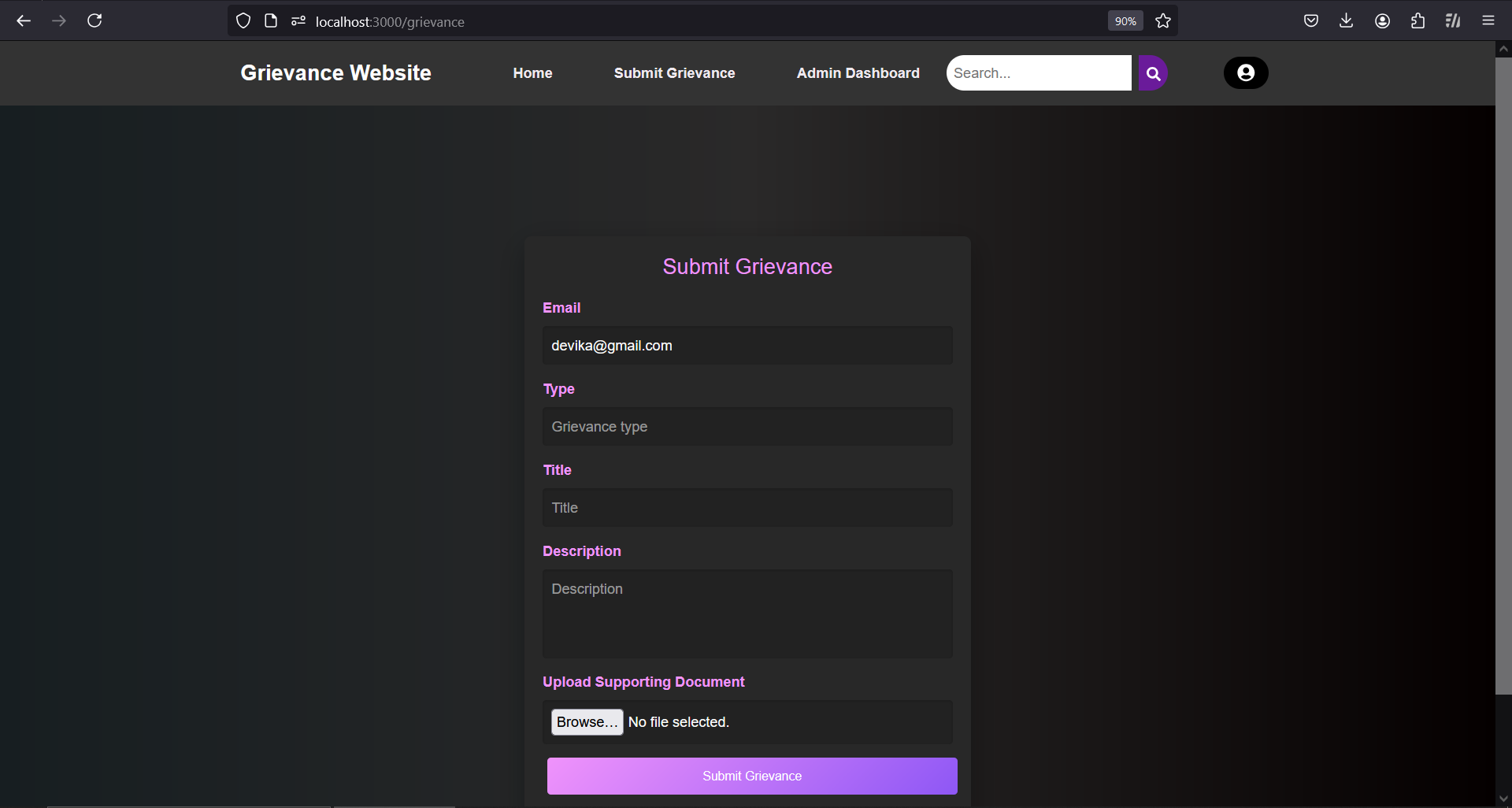
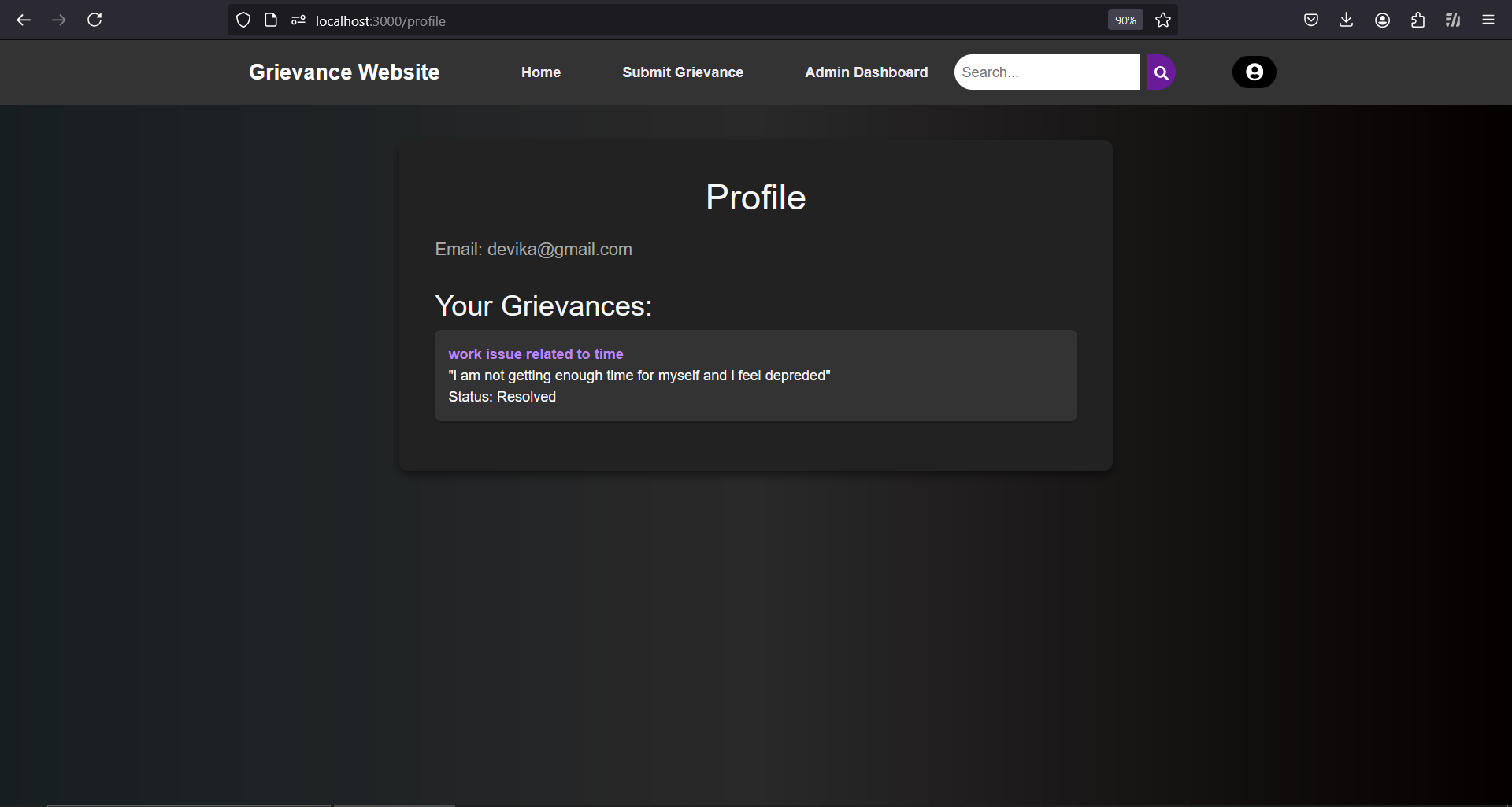
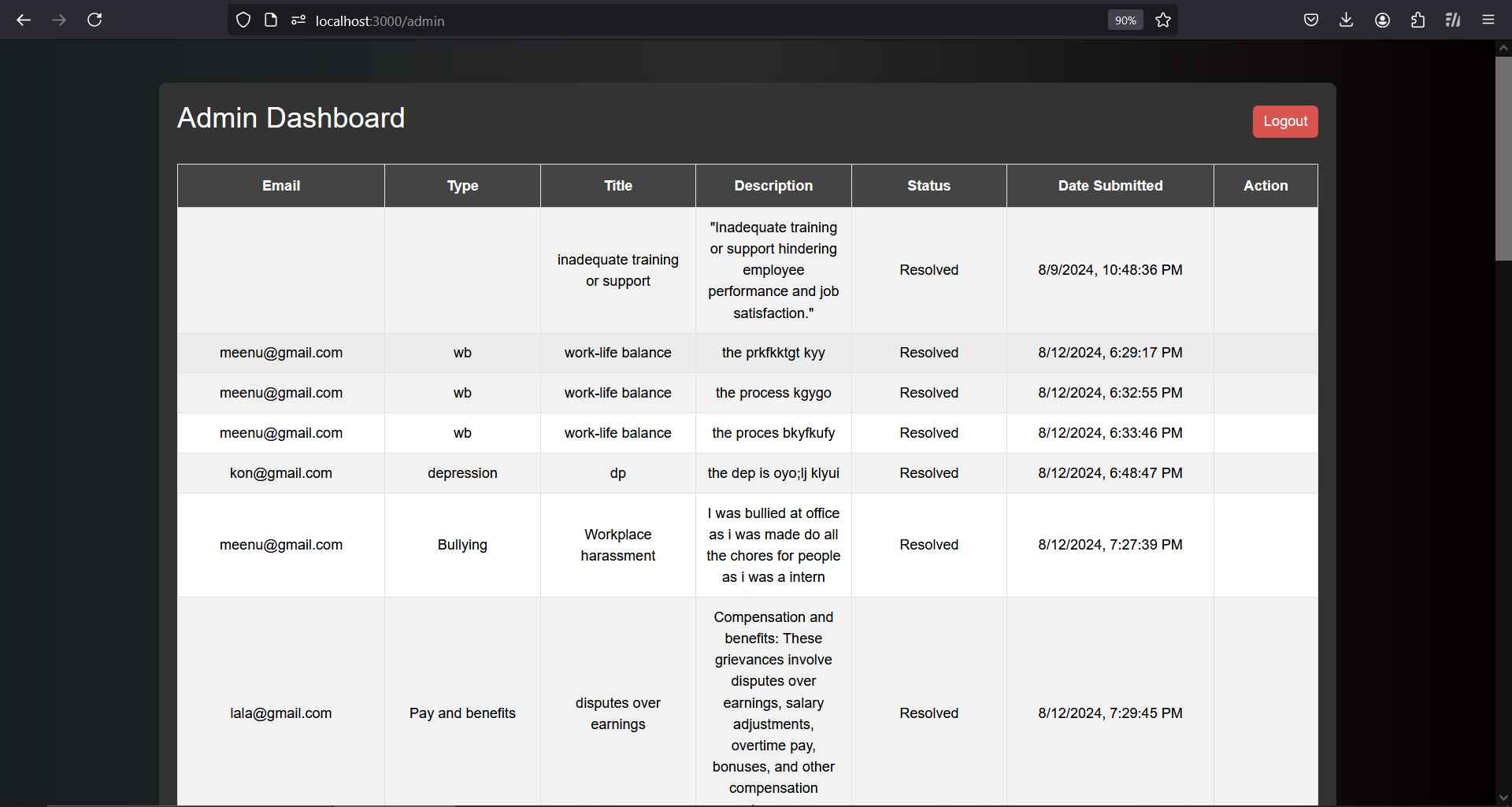
* **Role-based Access Control**: Implement distinct roles and permissions for different types of users.
* **Email Notifications**: Automatically notify users about the status of their grievances via email.
* **Third-Party Authentication**: Integrate with popular authentication providers like Google and Facebook.
* **Analytics and Reporting**: Provide detailed reports and analytics on grievances for better decision-making.

**6. Screenshots of the Application**

* **Home Page**:





* **Login Page**:   
  
* **Registration Page**:   
  
* **Submit Grievance Page**:   
    
  
* **Profile Page**:   
  
* **Admin Dashboard**:   
    
  
* **logout**:

