Job Portal Web Application

(About my Angular project)

A Job Portal Web Application built using Angular provides a seamless platform for job seekers and employers to connect. It features secure authentication, dynamic job listings with filters, job posting and management for employers, and an enquiry section for user queries. Angular's component-based architecture, two-way data binding, and efficient routing ensure a responsive, scalable, and user-friendly experience, making it ideal for managing the recruitment process effectively.

Features used in this project:

1. **User Authentication & Authorization:**
   - Secure login/signup for job seekers and recruiters.
   - Role-based access to control user functionalities.
2. **Home Page:**
   - Provides a quick overview of the latest job postings, featured companies, and trending job categories.
3. **Job Listings:**
   - Displays a list of available jobs with filters for location, category, salary, and experience.
   - Pagination and sorting for easy navigation.

Features:

4. **Job Postings (For Employers):**
    1. Employers can post, edit, and manage job listings.
    2. Ability to add company details and define job requirements.
5. **Enquiry Section:**
    1. Allows users to submit queries or contact for assistance.
    2. Admin can manage and respond to these inquiries efficiently.
6. **Login and Authentication:**
    1. Users can log in to access job listings or manage postings.
    2. JWT tokens for session management and API protection.

Component Architecture:

- The **Job Portal Web Application** utilizes **Angular** with **Bootstrap** for a modern, responsive interface. By using **standalone components**, it eliminates the need for NgModules, improving performance and simplifying maintenance.
- Bootstrap's grid system ensures a mobile-first design, while its pre-styled components like modals, buttons, and forms enhance the user experience.
- The application also leverages Angular's **lazy loading** to optimize page load times, and **form validation** is implemented to ensure data integrity during login, registration, and job postings.
- Additionally, Angular's **routing and guards** manage secure navigation between pages, ensuring a seamless and secure user journey.

Installing Required Dependencies

- **angular.json** does not manage dependencies directly.
- Dependencies are installed and managed through the package.json file.
- After installing, Angular uses angular.json to define:
  1. **Styles** (CSS, SCSS)
  2. **Scripts** (JS, external libraries)
  3. Asset paths, build options, etc.

To add any dependency to an Angular project:
(npm install package-name)

- This adds the package to dependencies in **package.json** file.
- With npm install command Node Modules will be installed in the project

Using Standalone Component

- Standalone components allow you to create Angular components **without the need for NgModules**.
- They simplify the structure by removing the need for **app.module.ts.**
- Introduced in Angular 14+ and now fully supported in Angular 17+.
- To generate a Standalone Component:

  ng g c my-component –standalone

Benefits of Using Standalone Components:

- No need to create app.module.ts
- Faster startup and smaller bundle size
- Improved maintainability and modularity
- Easier to manage dependencie

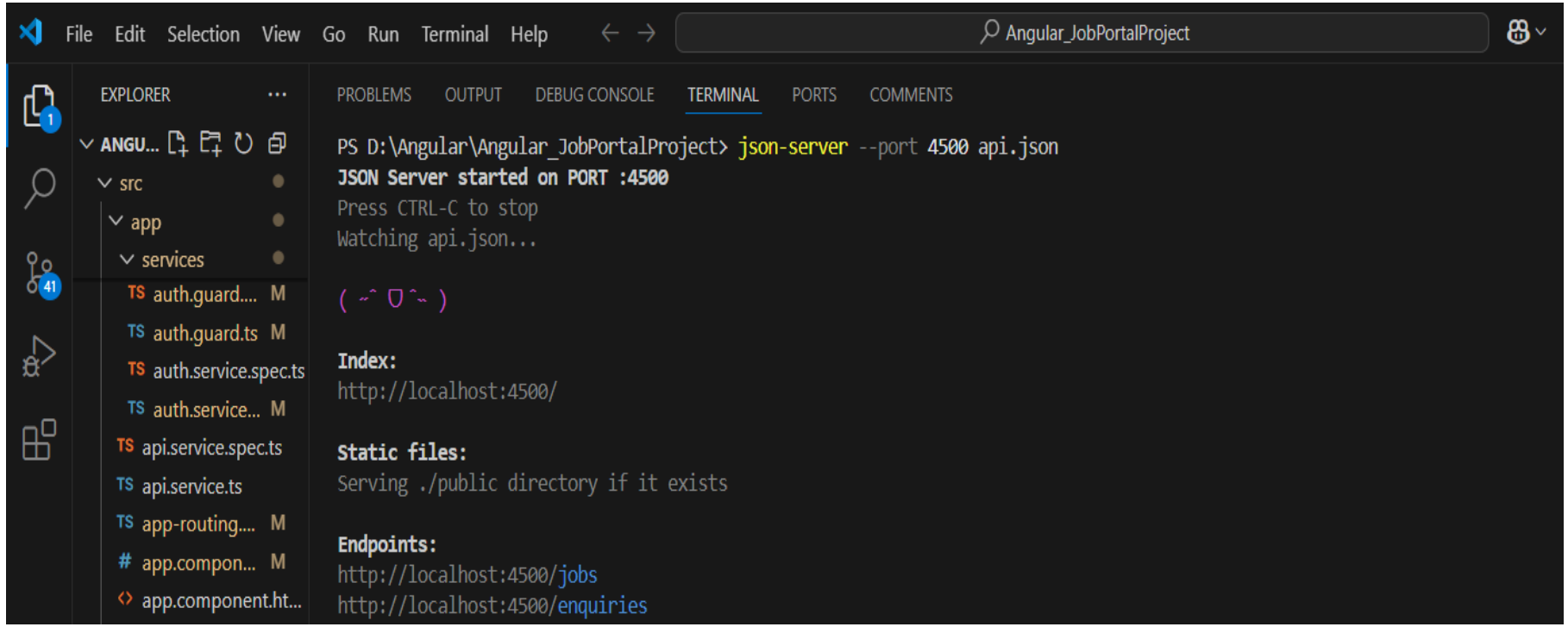
Bootstrapping:

•Bootstrapping is the process of **starting the Angular application** by loading the root component.
•Traditionally, this is done through AppModule with bootstrap: [AppComponent] inside @NgModule.
•With **standalone components**, there's no AppModule required. Instead, you bootstrap the app directly using bootstrapApplication().

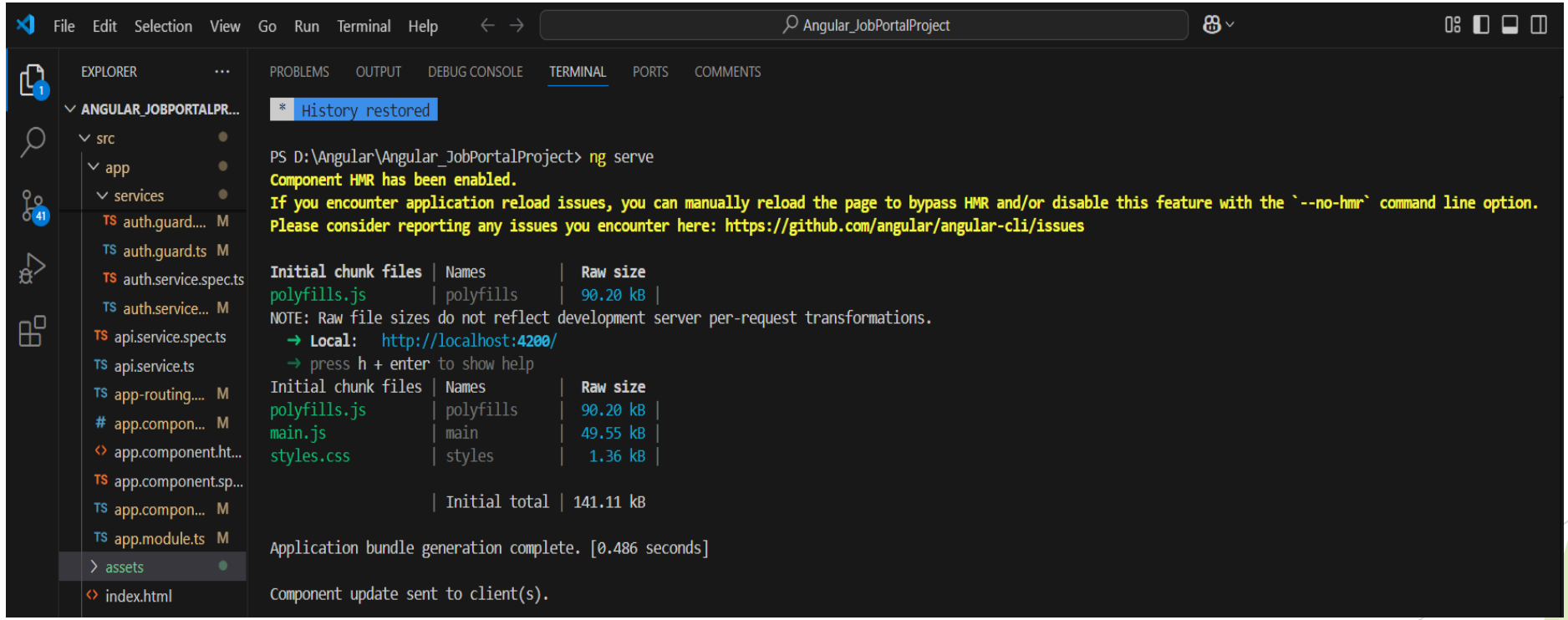
Benefits of using Standalone Bootstrapping:

- Faster load time and smaller bundle size
- No need to manage AppModule
- Directly bootstrap multiple components if needed
- Easier to manage and maintain

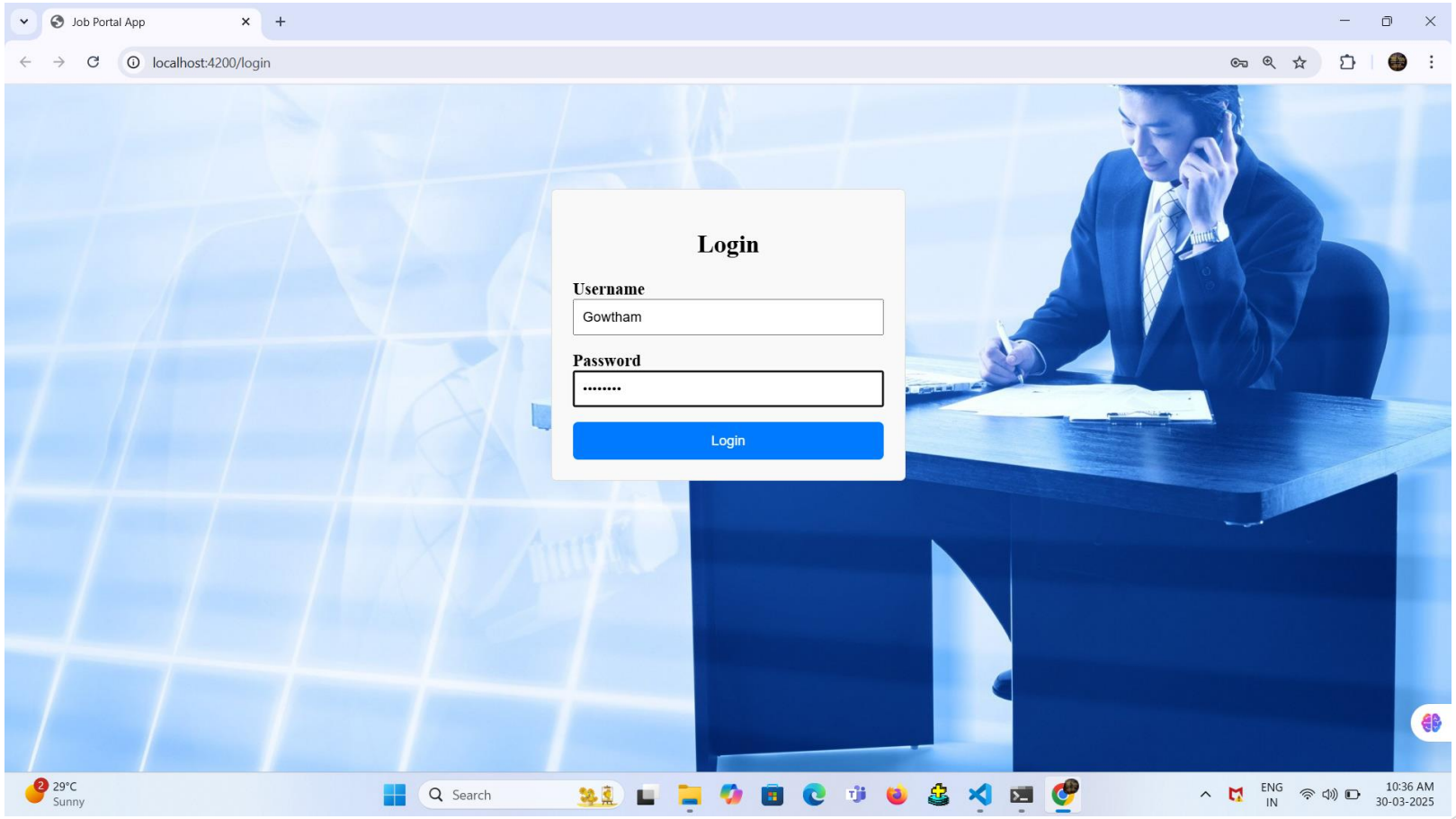
Running the Angular Project by starting an json server:



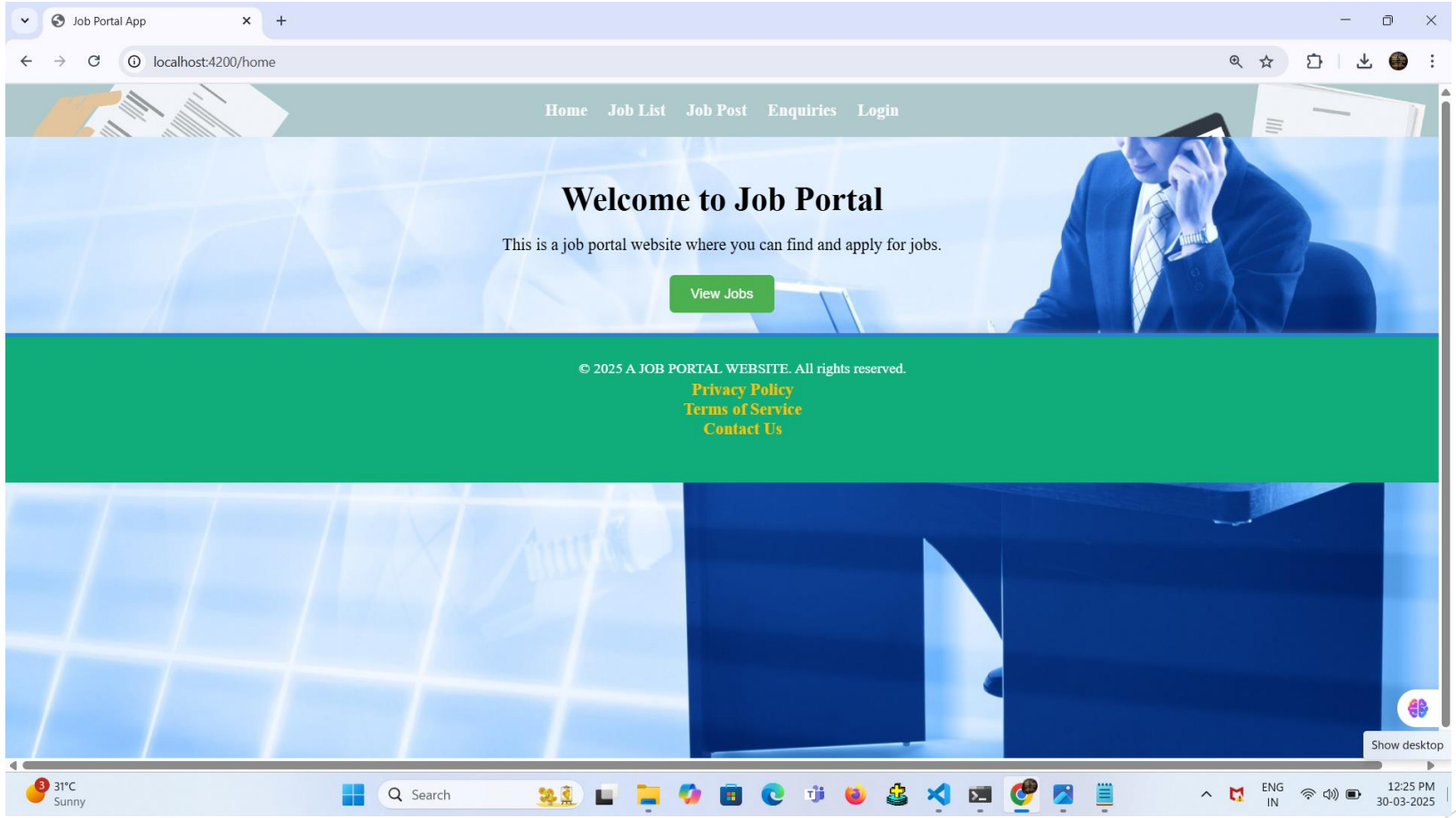
After running json server……need to put the ng serve command to receive the port with url:



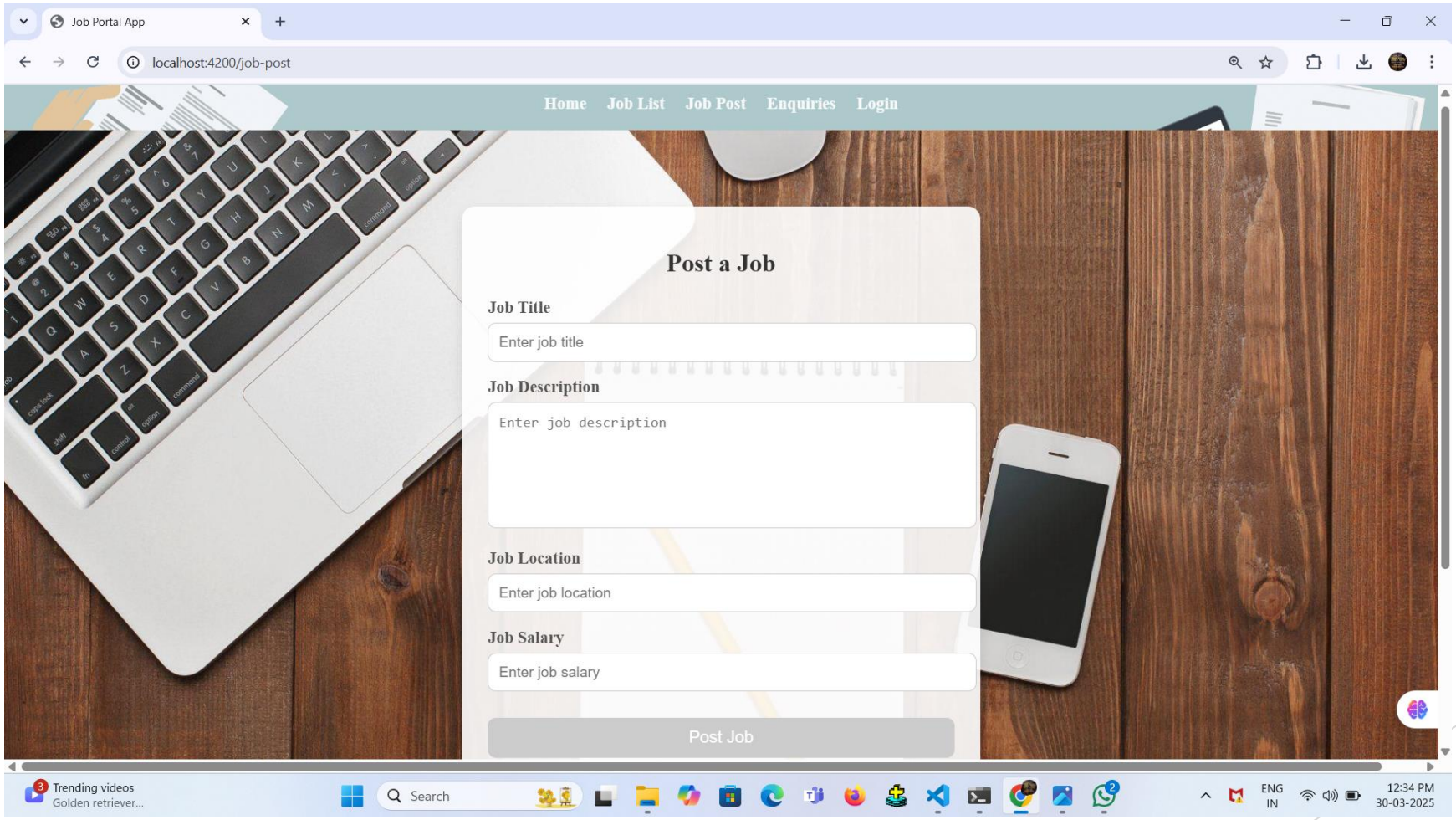
Login Page screenshot



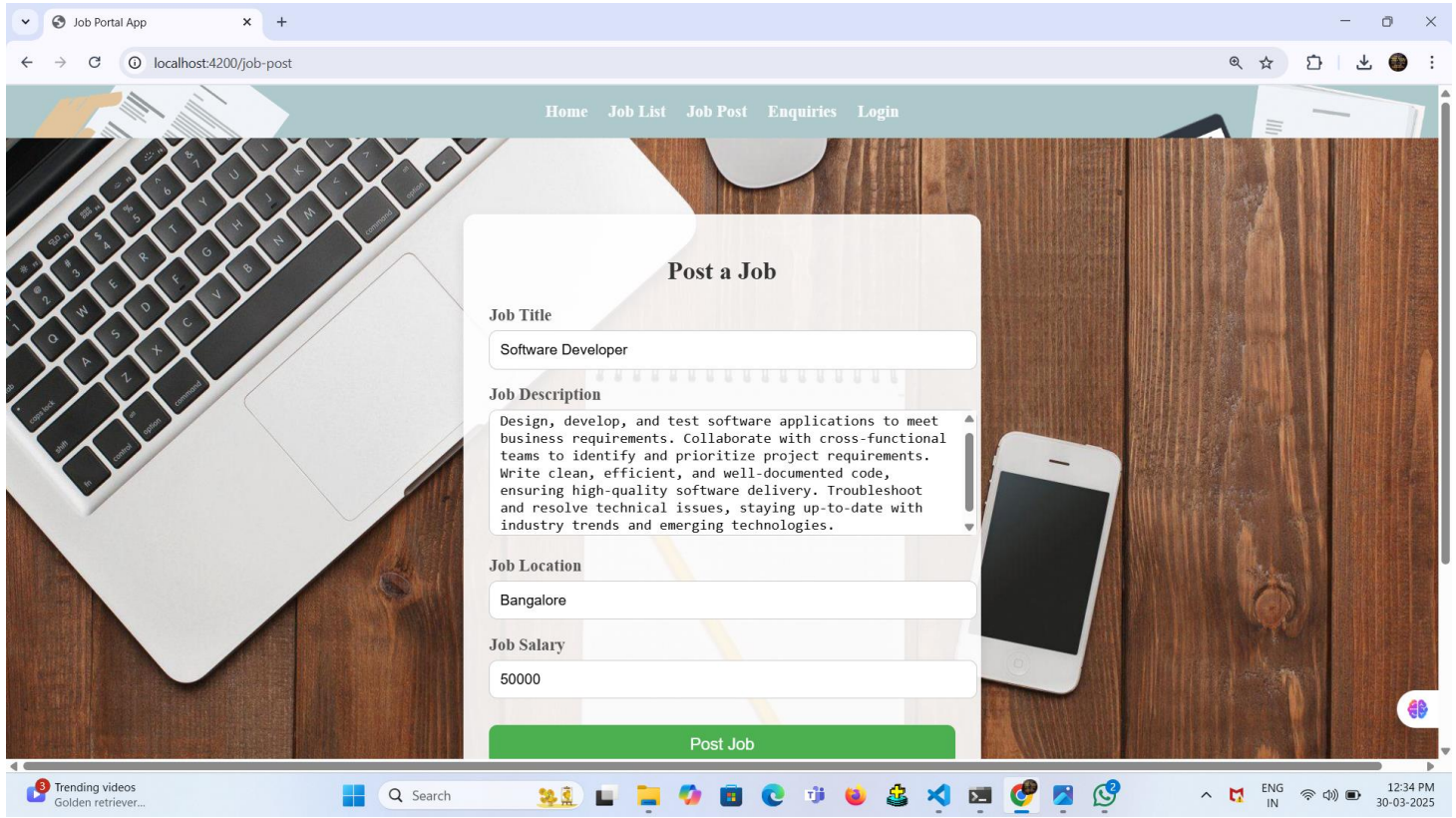
Home Page of Job Portal



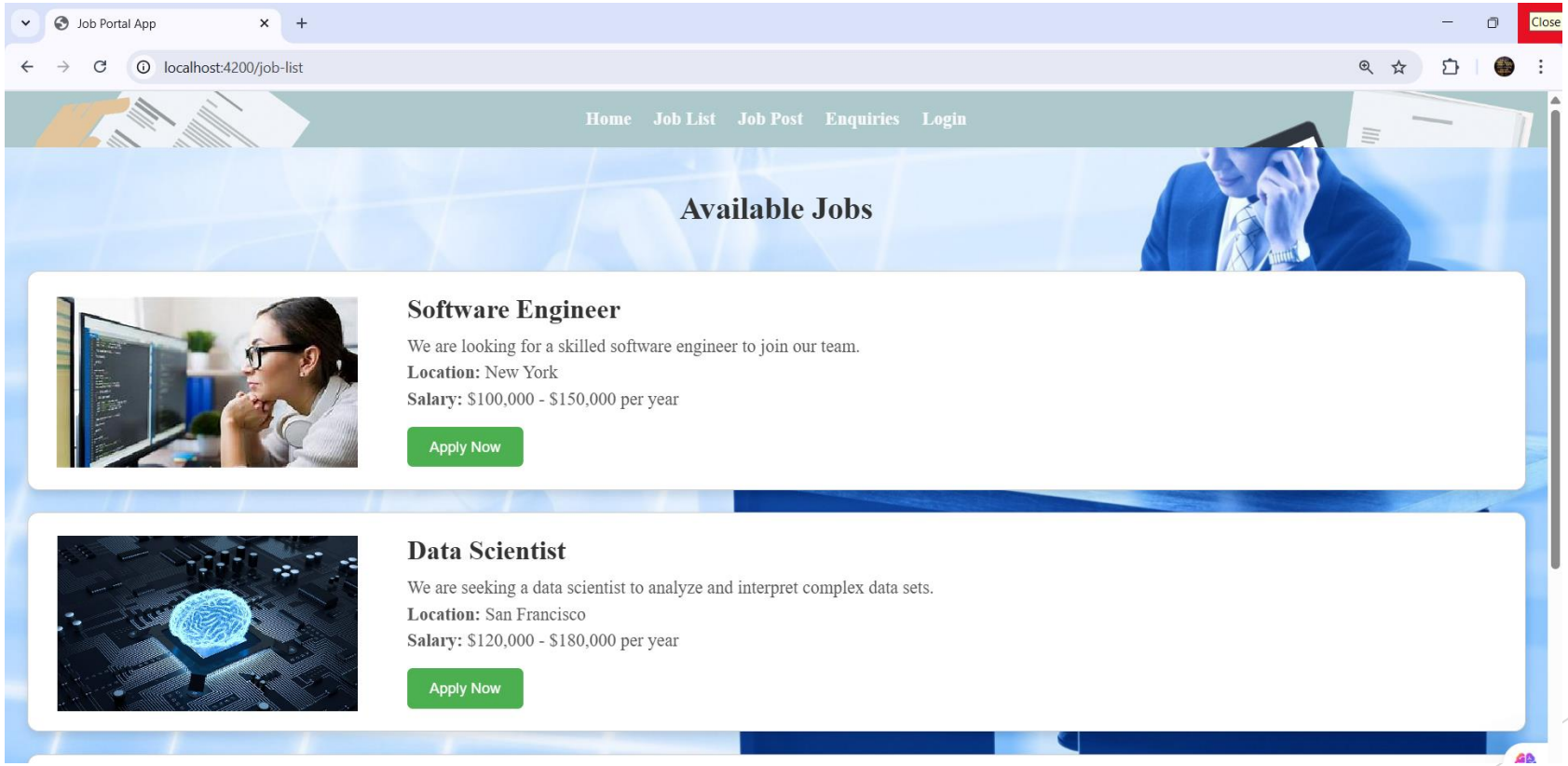
Post Job Page of Job Portal



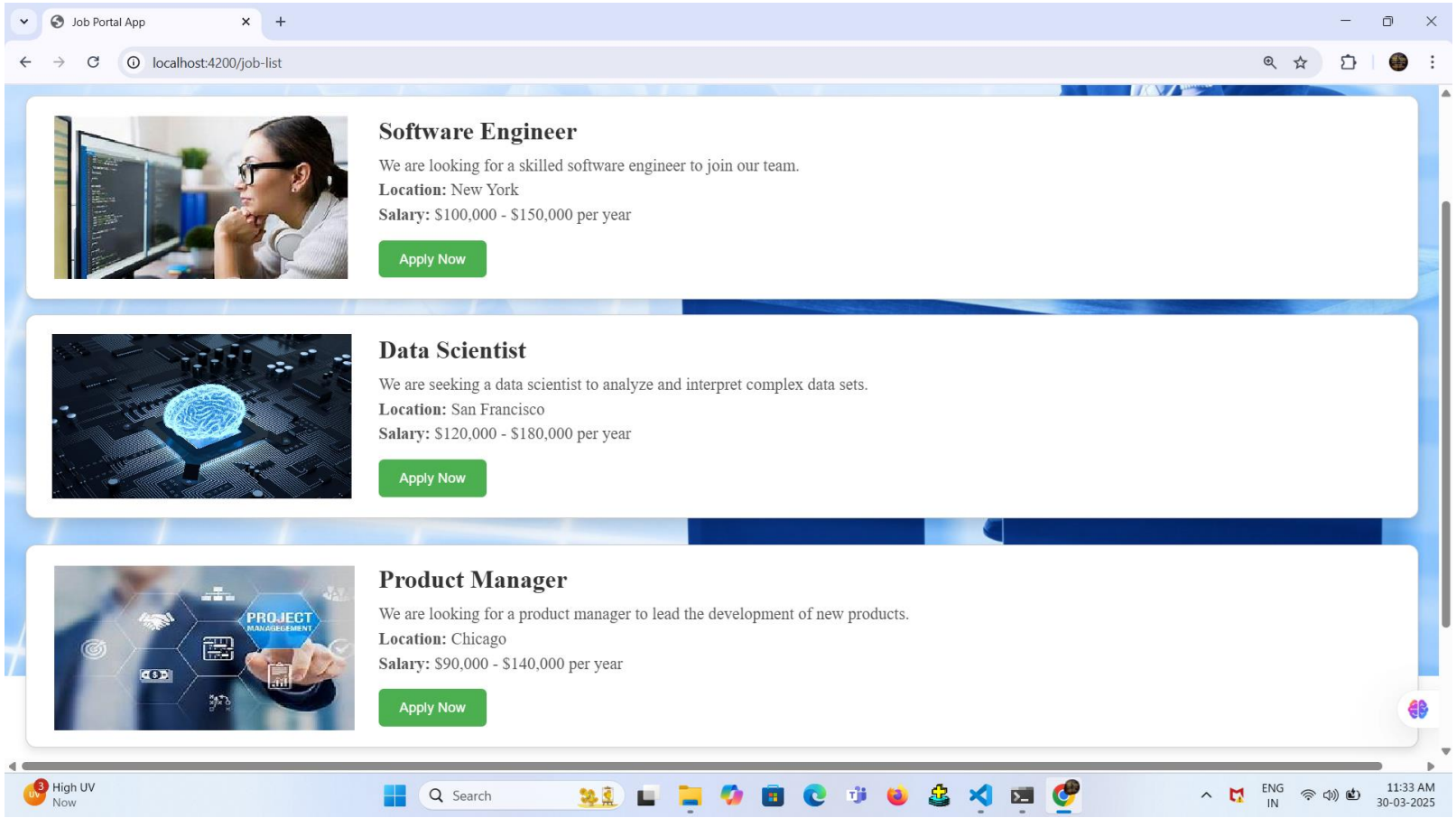
If any job need to post, it can be posted by filling the form with its description,…



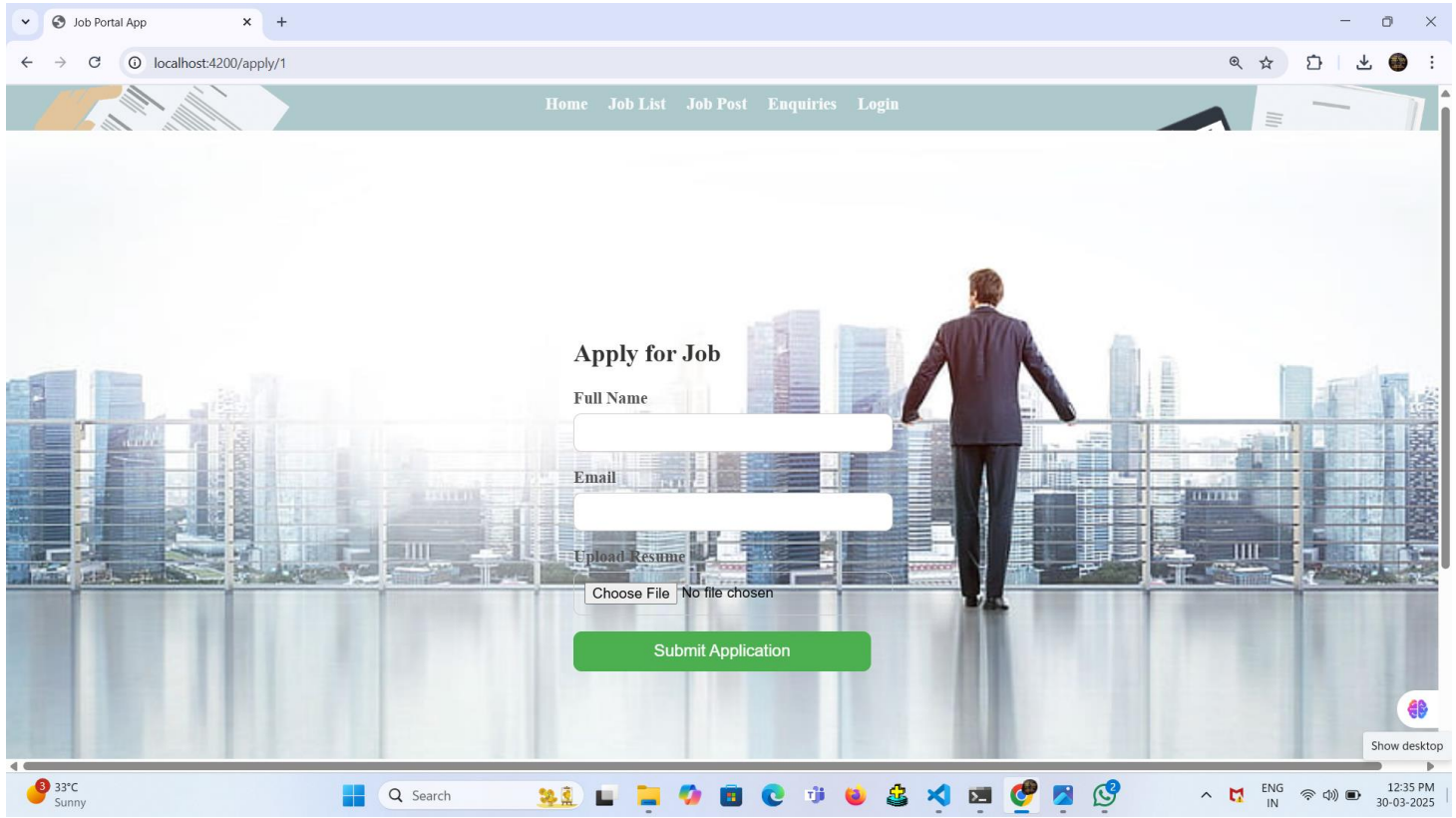
Job List Page shows the available jobs…where the candidate can see and apply here…



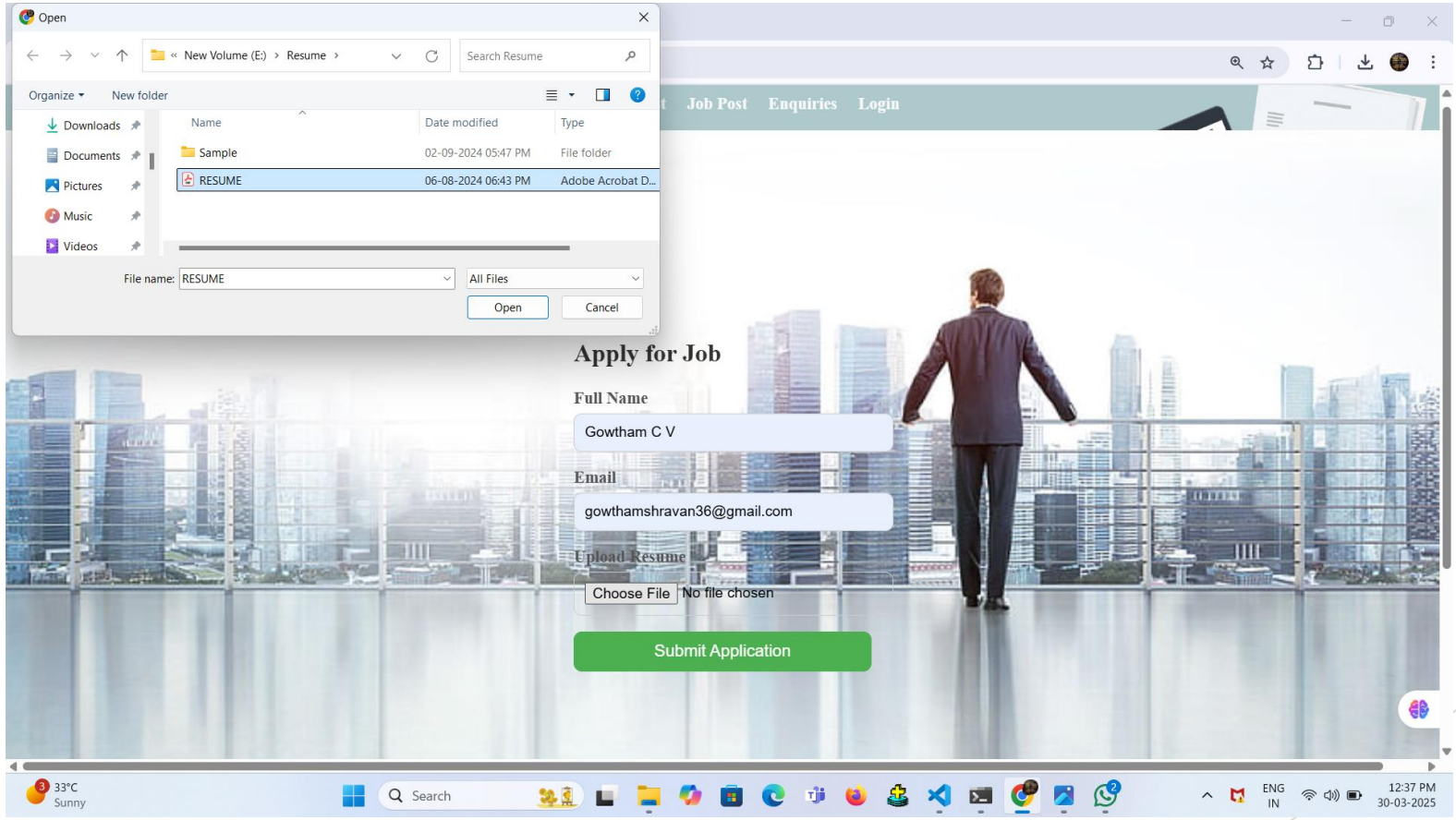
Full Page view of Available jobs:



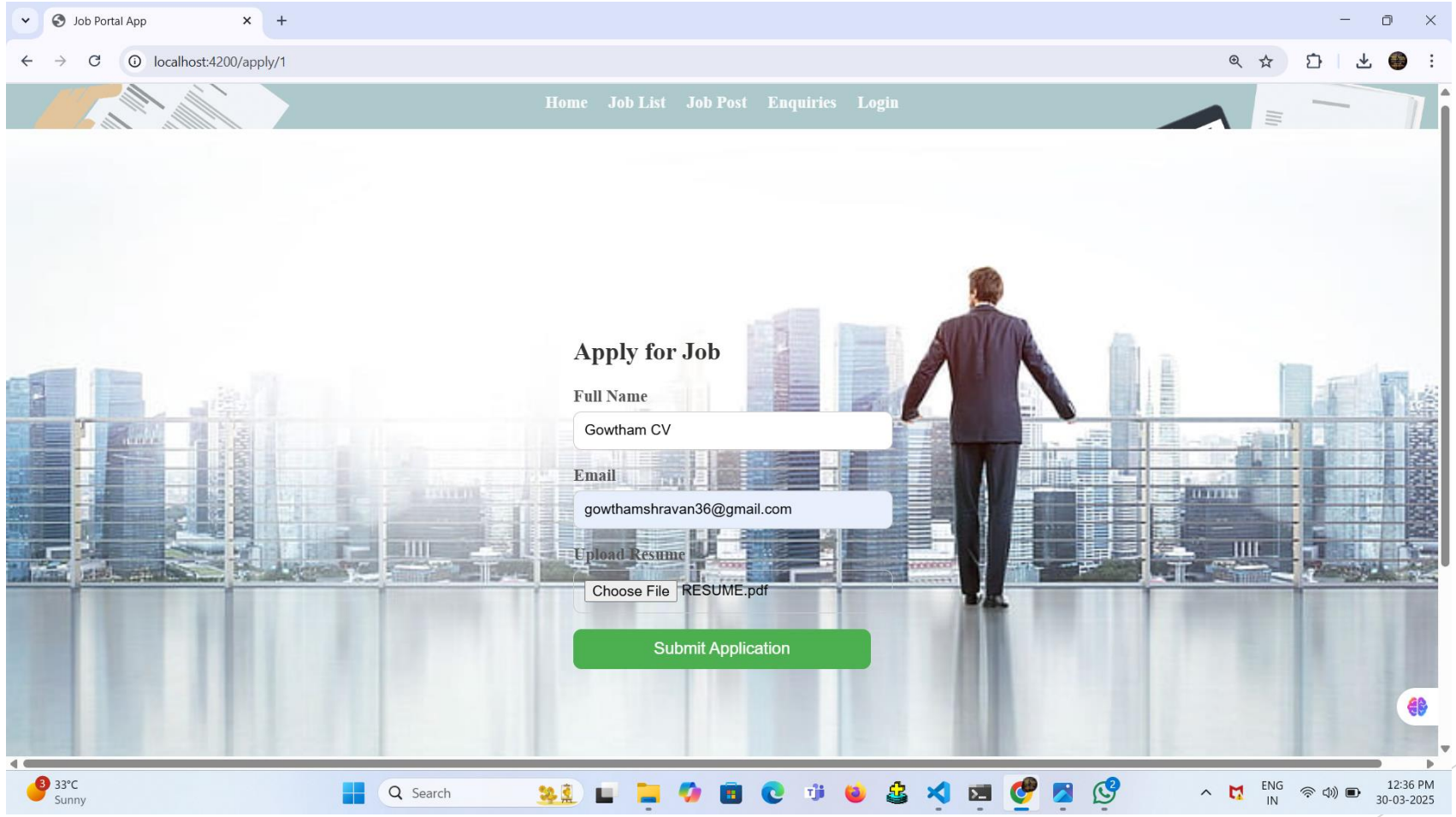
When Clicking Apply Now button, it will redirects to the Apply for Job…



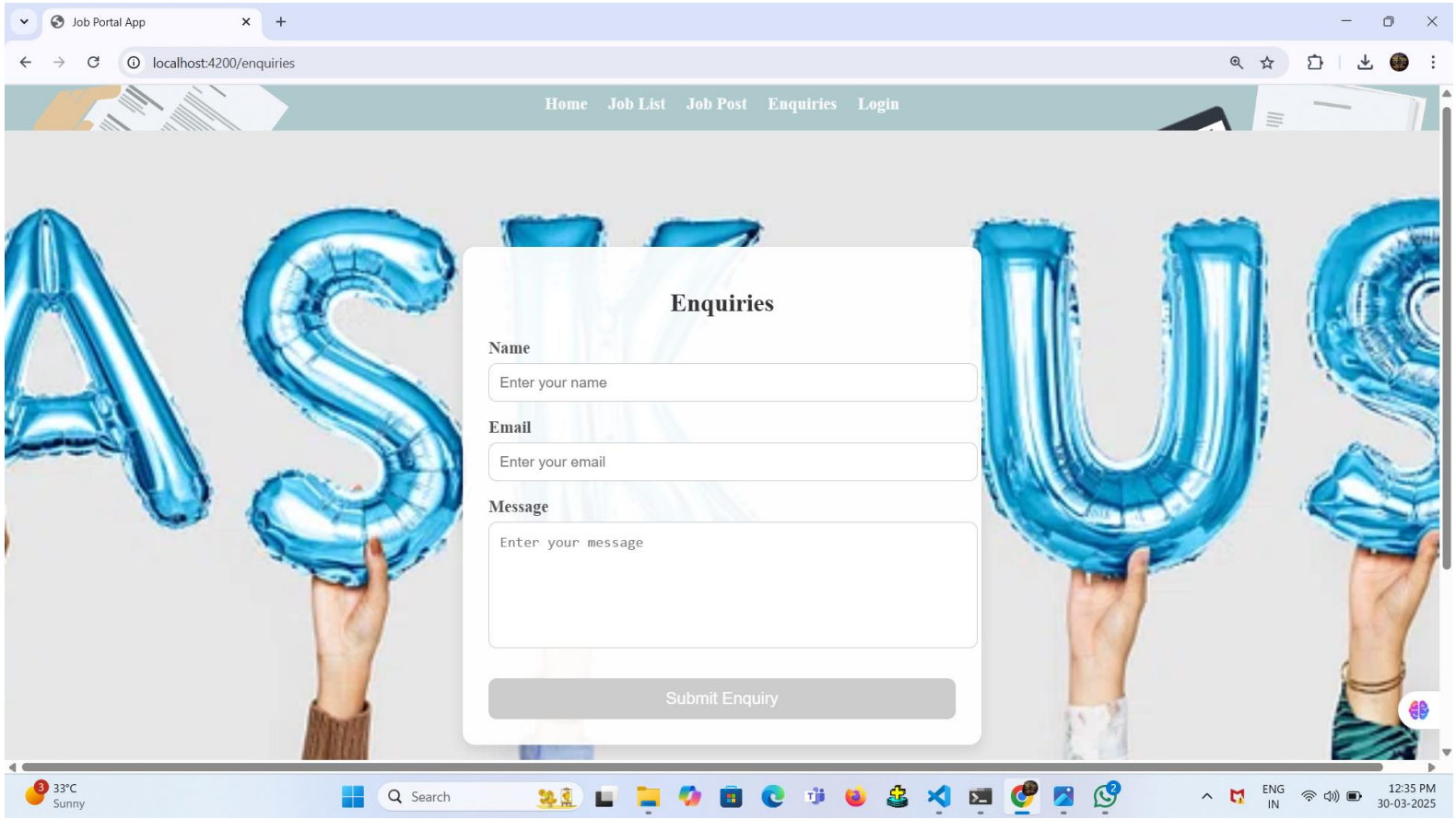
The candidate can apply for the job by filing the details along with their resume…



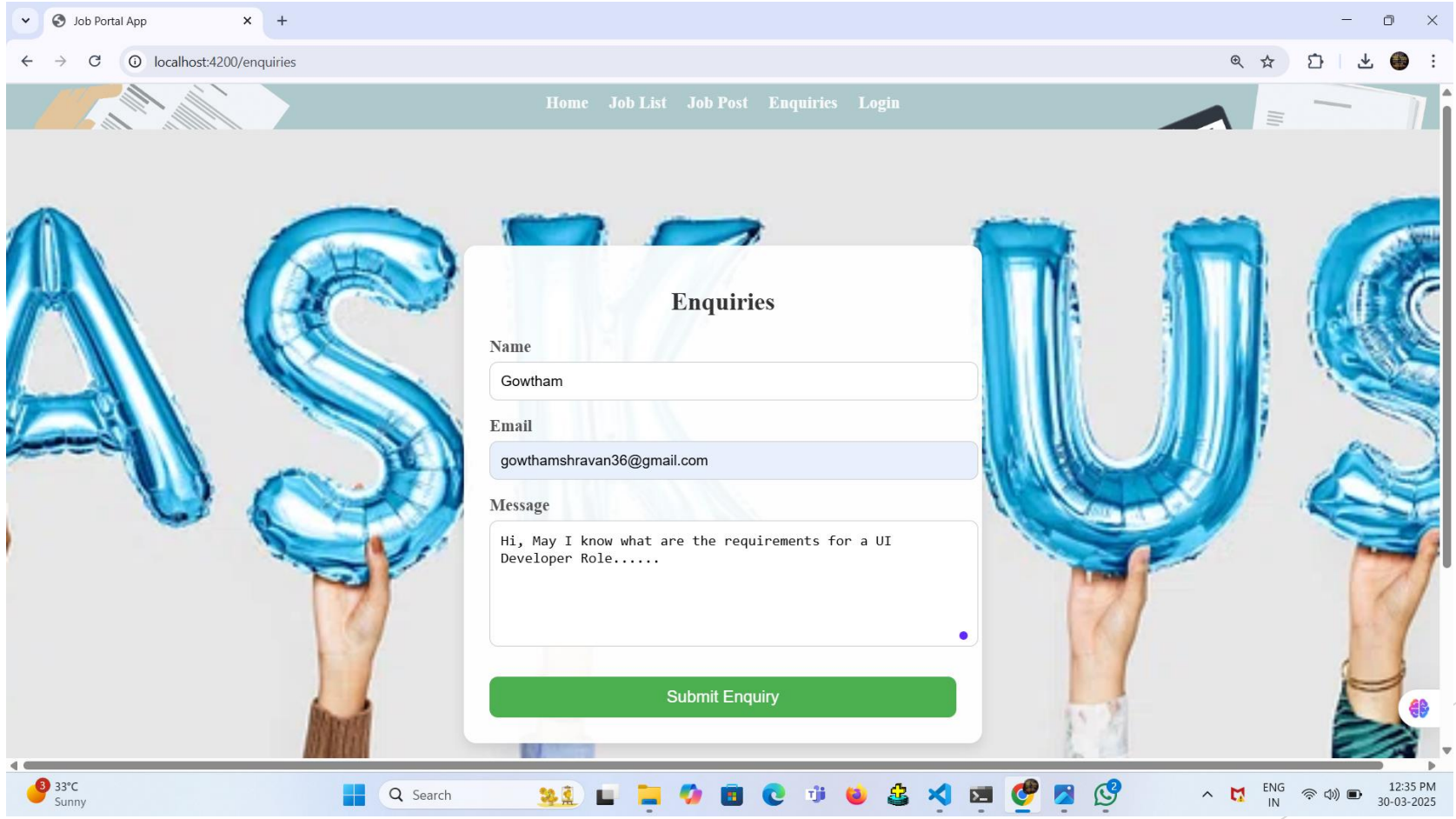
After submitting their resume, they can submit the application…



If any enquiries regarding the job role, they can ask by filling the enquiry form…



Finally the enquiry form will be submitted….after filling the details



Conclusion

The **Angular Job Portal Application** is a dynamic and responsive platform that enables job seekers to explore listings and apply for positions seamlessly. Built using **Angular with standalone components**, it ensures a modular and maintainable architecture. The project integrates **REST APIs** for data fetching, provides secure **authentication**, and uses **routing** for smooth navigation. It supports **mock APIs** using JSON Server during development and is optimized for production with **NGINX deployment**. This project effectively demonstrates practical knowledge of Angular concepts and serves as a scalable foundation for future enhancements.