**JOBPORTAL**

**INTRODUCTION**

The "Job Portal Project" is a comprehensive and innovative endeavor aimed at creating a cutting-edge online platform that revolutionizes the way job seekers and employers connect in the modern job market. This project seeks to bridge the gap between job seekers looking to advance their careers and employers seeking talented individuals to join their teams. By leveraging advanced technology and user-centric design, the Job Portal Project aims to streamline the job search and recruitment process, providing a seamless and efficient experience for all stakeholders.

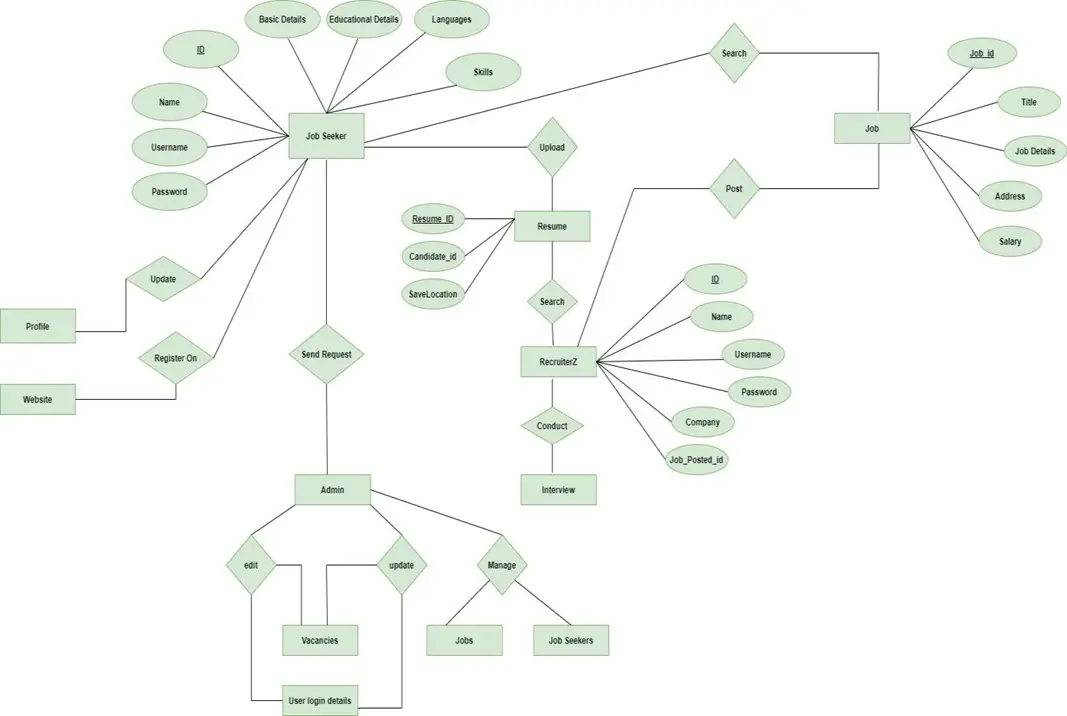
In a rapidly evolving professional landscape, traditional methods of job search and recruitment are being reshaped by technology. The Job Portal Project addresses this transformation by introducing a digital ecosystem that transforms the way job seekers and employers interact. This platform serves as a hub where job seekers can showcase their skills, experiences, and aspirations, while employers can present their organizations and job opportunities to a wide and diverse talent pool.

The Job Portal Project is a progressive leap forward in the world of job search and recruitment, combining technological innovation with the human element of career aspirations. It reimagines how individuals find meaningful work and organizations discover exceptional talent, ultimately contributing to a more dynamic, inclusive, and efficient job market.

**Technical Architecture**



# ER Diagram:



The Entity-Relationship (ER) diagram for a flight booking app visually represents the relationships between different entities involved in the system, such as job seeker, admin, job, recruiter. It illustrates how these entities are related to each other and helps in understanding the overall database structure and data flow within the application.It can certainly help you describe the components of an ER diagram for a job portal. An Entity-Relationship (ER) diagram is a visual representation of the entities (objects) within a system, the relationships between those entities, and the attributes associated with the entities. Keep in mind that creating a complete ER diagram requires understanding the specific requirements and features of your job portal.

Entities:

1. User:

- Attributes: UserID (Primary Key), Username, Email, Password, UserType (Job Seeker or Employer), RegistrationDate

2. Job Seeker: - Attributes: JobSeekerID (Primary Key, Foreign Key referencing User.UserID), FirstName, LastName, ContactNumber, Skills, Experience, ProfileImage

3. Employer:

- Attributes: EmployerID (Primary Key, Foreign Key referencing User.UserID), CompanyName, ContactPerson, CompanyDescription, Industry

4. Job Listing:

- Attributes: JobID (Primary Key), Title, Description, Requirements, Location, Salary, PostedDate

- Relationships: PostedBy (Foreign Key referencing Employer.EmployerID)

Relationships:

- 1-to-many relationship between JobSeeker and Application

- 1-to-many relationship between Employer and JobListing

- 1-to-many relationship between JobListing and Application

- many-to-1 relationship between JobSeeker and Application, and between JobListing and Application

**Key Features**

Key features of a job portal can vary based on the platform's design, target audience, and technological capabilities. However, here are some essential and commonly found features in a comprehensive job portal.

**User Registration and Profiles:** Allow job seekers and employers to create accounts and profiles with relevant information. Collect details such as skills, experiences, education, and preferences.

**Job Listings:**Enable employers to post job openings with detailed descriptions, requirements, and application instruction. Allow job seekers to search, filter, and view job listings based on various criteria.

**Advanced Search and Filters:** Provide search and filtering options based on location, industry, job type, salary range, and more.Help job seekers find relevant opportunities efficiently.

**Application Submissions:** Allow job seekers to apply for jobs directly through the platform.Provide options to attach resumes, cover letters, and other required documents.

**Application Tracking:**Provide a dashboard for job seekers to track the status of their applications. - Offer employers tools to manage received applications and review candidates.

**Notifications and Alerts:**Send notifications to job seekers about new job listings, application updates, and relevant events.Alert employers about new applications and important updates.

**Company Profiles:**Offer employers the opportunity to create detailed company profiles with information about their culture, values, and benefits.

**Resume Building and Editing:**Provide tools for job seekers to create and edit their resumes directly on the platform.Allow for the import of external resumes.

**Support and Help Center:**Offer user support and a knowledge base to assist users in navigating the platform effectively.

**Social Media Integration:**- Allow users to share job listings on social media platforms for wider visibility.

# PRE REQUISITES:

To develop a full-stack jobportal app using AngularJS, Node.js, and MongoDB,there are several prerequisites you should consider. Here are the key prerequisites for developing such an application:

**Node.js and npm:** Install Node.js, whichincludes npm (NodePackage Manager), on your development machine. Node.js is required to run JavaScript on the serverside.

* 1. Download: <https://nodejs.org/en/download/>
  2. Installation instructions: <https://nodejs.org/en/download/package-manager/>

**MongoDB:** Set up a MongoDB database to store hotel and booking information. InstallMongoDB locally or use a cloud-based MongoDBservice.

* 1. Download: <https://www.mongodb.com/try/download/community>
  2. Installationinstructions: <https://docs.mongodb.com/manual/installation/>

**Express.js:** Express.js is a web application framework for Node.js.Install Express.js to handle server-side routing, middleware, and API development.

* 1. Installation: Open your commandprompt or terminaland run the following command:**npm install express**

**Angular:** Angular is a JavaScript framework for building client-side applications. Install Angular CLI (CommandLine Interface) globally to create and manage your Angular project.

### 

### Install AngularCli

### 

* 1. Angular provides a command-line interface (CLI) tool that helps with project setup and developmet.
  2. Install the Angular CLI globally by running the following command:

### npm install -g @angular/cli

**Verify the Angular CLI installation:**

* 1. Run the following commandto verify that the AngularCLI is installed correctly: ng version

You shouldsee the version of the Angular CLI printed in the terminalif the installation was successful.

### Create a new Angular project:

* 1. Choose or create a directory where you want to set up your Angular project.
  2. Open your terminal or command prompt.
  3. Navigate to the selecteddirectory using the cd command.
  4. Create a new Angular project by running the following command: ng new client Wait for the project to be created:
  5. The Angular CLI will generate the basic project structure and install the necessary dependencies

**Navigate into the project directory:**

* 1. After the project creationis complete, navigateinto the projectdirectory by runningthe following command**: cd client**

## Start the development server:

* 1. To launchthe development serverand see your Angular app in the browser, run the followingcommand: **ng serve/ npm start**
  2. The Angular CLI will compileyour app and start the development server.
  3. Open your web browser and navigate to http://localhost:4200 to see your Angularapp running.

You have successfully set up Angularon your machine and createda new Angular project. You can now start building your app by modifying the generated project files in the src directory.

Please note that these instructions provide a basicsetup for Angular. You can explore more ad- vanced configurations and features by referring to the official Angular documentation: https://angular.io

**HTML, CSS, and JavaScript:** Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential

**Database Connectivity:** Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and performCRUD (Create, Read, Update,Delete) operations.

**Front-end Framework:** Utilize Angular to build the user-facing part of the application, including products listings, bookingforms, and user interfaces for the admin dashboard.

**Version Control**: Use Git for versioncontrol, enabling collaboration and tracking changesthroughout the development process. Platforms like GitHub or Bitbucket can host your repository.

1. Git: Download and installation instructions can be found at: [https://git-](https://git-scm.com/downloads) [scm.com/downloads](https://git-scm.com/downloads)

**Development Environment:** Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

1. Visual Studio Code: Downloadfrom<https://code.visualstudio.com/download>
2. Sublime Text: Download from <https://www.sublimetext.com/download>
3. WebStorm: Download from <https://www.jetbrains.com/webstorm/download>

### To Connectthe Database with Node JS go throughthe below providedlink:

1. Link:[https://www.section.io/engineering-education/nodejs- mongoosejs-](https://www.section.io/engineering-education/nodejs-%20mongoosejs-mongodb/) [mongodb/](https://www.section.io/engineering-education/nodejs-%20mongoosejs-mongodb/)

## To run the existing Expense Tracker App project downloaded from github:

**Follow below steps:**

## 1.Clone the Repository:

* Open your terminal or command prompt.
* Navigate to the directory where you want to storethe movie ticketbooking app.
* Execute the following commandto clone the repository:

**git clone** https://github.com/Rakeshvinay/Job-Hungry.git

## 2.Install Dependencies:

Navigate into the cloned repository directory: cd job portal portal

* + Install the required dependencies by running the following command:npm install

## 3.Start the Development Server:

* To start the development server,execute the followingcommand: npm run startart
* The e-commerce app will be accessible at http://localhost:5100 by default. You can changethe port configuration in the .env file ifneeded.

**4 Access the App:**

* + Open your web browserand navigate to http://localhost:5100.
  + You should see the movie-ticket-booking app's homepage, indicating that the installation and setup were successful.

**Project Repository Link:**  https://github.com/Rakeshvinay/Job-Hungry.git

Congratulations! You have successfully installed and set up the movie-ticket-booking app on your local machine. You can now proceed with further customization, development, and testingas needed.

**Roles and Responsibilities**

1**. Job Seeker**:Create and Maintain Profile:Job seekers need to create a detailed profile with their resume, skills, experience, and preferences. They should keep their profile updated.

* Search and Apply for Jobs: Job seekers use the portal to search for relevant job listings based on their skills and preferences. They need to review job descriptions and apply to positions of interest.

* Networking:Some job portals offer networking features, allowing job seekers to connect with professionals in their field, join groups, and participate in discussions.

* Receive Notifications: Job seekers should set up notifications to receive updates on new job listings, application status, and interview invitations.

**Employer/Recruiter:**

* Post Job Listings:Employers and recruiters create and post job listings on the portal, including detailed job descriptions, qualifications, and application instructions.

* Review Applications: They review applications submitted by job seekers, screen resumes, and shortlist candidates who match the job requirements.

**Communicate with Candidates**:

* Employers can use the portal to communicate with candidates, schedule interviews, and provide feedback on applications.

* Manage Job Listings: Employers may need to manage their active job listings, close positions that are filled, or update job descriptions based on evolving requirements.

**Job Portal Administrator**:

* Platform Maintenance: Administrators are responsible for maintaining the technical aspects of the job portal, ensuring it's functioning properly and resolving any technical issues.

* User Support: They provide support to job seekers and employers who may encounter problems using the platform or have questions about its features.

* Security and Privacy: Administrators need to ensure the security of user data and protect privacy by implementing security measures and adhering to data protection regulations.

* Analytics and Reporting: Administrators monitor the performance of the portal, gather usage data, and generate reports that provide insights into user behavior and trends.

**Content Management:**

* Job Content: Responsible for creating and editing job listings, ensuring they are accurate, clear, and well-structured.

* Content Updates: Regularly update and refresh job listings, ensuring that they reflect current openings accurately.

* SEO Optimization: Optimize job listings and other content to improve visibility on search engines and within the job portal's search functionality.

**Customer Service:**

* Assistance: Provide assistance to users who may have questions about the platform, encounter technical issues, or need guidance on using the portal effectively.
* Issue Resolution:Address and resolve user concerns, whether related to accounts, job applications, or any other aspect of the portal.

* Remember that the specific roles and responsibilities within a job portal can vary based on the platform's features, target audience, and business model. The goal is to create a seamless experience for job seekers and employers, facilitating successful matches between talent and job opportunities.

## 

## User Flow:

## A user flow for a job portal outlines the steps that a user, whether a job seeker or an employer, takes while navigating through the platform. Here's an example of a simplified user flow for a job portal:

## User Flow for Job Seeker:

## 1. Homepage:

## - User arrives at the job portal's homepage.

## - They see options to sign in or create a new account.

## 2. Account Creation:

## - New user selects "Create Account."

## - They enter personal details, upload a resume, and set preferences.

## - User's account is created.

## 3. Profile Setup:

## - User is prompted to complete their profile with skills, experience, and education.

## - They also set job preferences such as industry, location, and job type.

## 4. Job Search:

## - User logs in and is directed to the job search page.

## - They use filters (location, job type, etc.) to narrow down job listings.

## - Job listings matching their preferences are displayed.

## 

## 5. Job Listing Details:

## - User clicks on a job listing to view details.

## - They read the job description, qualifications, and responsibilities.

## 6. Application:

## - User decides to apply for a job.

## - They click the "Apply" button and submit their resume and a cover letter.

## 7. Application Tracking:

## - User can track the status of their applications on their dashboard.

## - They receive notifications if there's a change in status.

**8.Interview Scheduling:**

## - User receives an interview invitation.

## - They can accept or decline the invitation through the platform.

## 9. Networking (Optional):

## - User can connect with professionals in their field or join groups for networking.

## - They can participate in discussions and expand their professional network.

## 

## User Flow for Employer:

## 1. Homepage:Employer arrives at the job portal's homepage.

## -They have options to sign in or create a new account for their company.

## 2. Company Account Creation:

## - Employer selects "Create Company Account."

## - They provide company details, logo, and create an account.

## 3. Job Posting: Employer logs in and accesses their company dashboard.

## They create a new job listing, including job title, description, qualifications, and application instructions.

## 4. Job Listing Management: Employer can view and manage active job listings.They can edit, close, or duplicate job listings as needed.

**Project Repository Link:**  https://github.com/Rakeshvinay/Job-Hungry.git

Congratulations! You have successfully installed and set up the movie-ticket-booking app on your local machine. You can now proceed with further customization, development, and testingas needed.

**UserFlow**

User Flow for Job Seeker:

1.**Homepage:**

* User arrives at the job portal's homepage.
* They see options to sign in or create a new account.

2**.Account Creation:**

* New user selects "Create Account."
* They enter personal details, upload a resume, and set preferences.
* User's account is created.

3.**Profile Setup:**

* User is prompted to complete their profile with skills, experience, and education.
* They also set job preferences such as industry, location, and job type.

4. **Job Search:**

* User logs in and is directed to the job search page.
* They use filters (location, job type, etc.) to narrow down job listings.
* Job listings matching their preferences are displayed.

5.**Job Listing Details**:

* User clicks on a job listing to view details.
* -They read the job description, qualifications, and responsibilities.

User Flow for Employer:

1. **Homepage:**

* Employer arrives at the job portal's homepage.
* They have options to sign in or create a new account for their company.

2**.Company Account Creation**:

* Employer selects "Create Company Account."
* They provide company details, logo, and create an account.

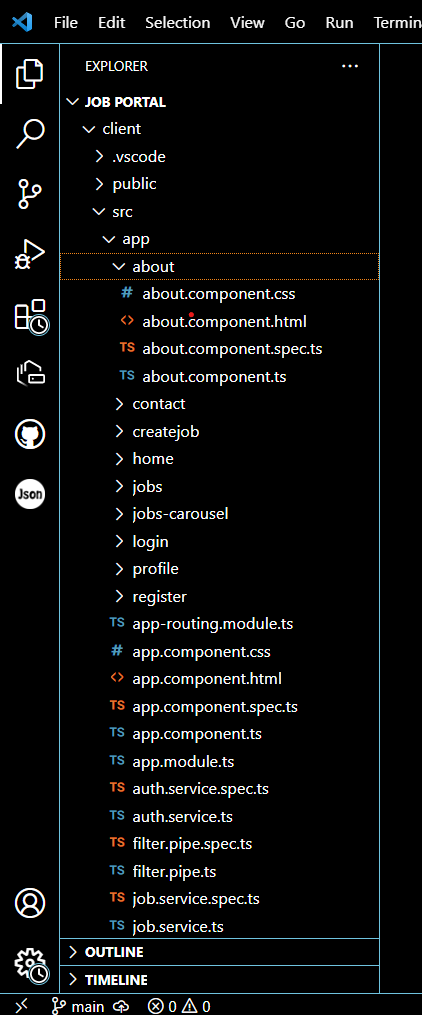
3**.Job Posting:**

* Employer logs in and accesses their company dashboard.
* They create a new job listing, including job title, description, qualifications, and application instructions.

4**.Job Listing Management**:

* Employer can view and manage active job listings.They can edit, close, or duplicate job listings as needed.

**Project Structure:**



This structure assumes an Angular app and follows a modular approach. Here's a brief explanation of the main directories and files:

1. src/app/components: Containscomponents related to the customerapp, such as register, login,home, bookings, owner-login, owner-register and more.
2. src/app/modules: Containsmodules for differentsections of the app. In this case,the admin module is included with its own set of components like add-flight, bookings, dashboard, flights, home, users, and more.
3. src/app/app-routing.module.ts: Definesthe routing configuration for the app, specifying whichcomponents should be loaded for each route.
4. src/app/app.component.ts, src/app/app.component.html, src.

## PROJECT FLOW:

**Milestone 1: Project Setupand Configuration:**

### Install required tools and software:

* Node.js.
* MongoDB.
* Angular CLI.

### Create projectfolders and files:

* + Client folders.
  + Server folders.

## Milestone 2: Backend Development:

### Setup express server:

* + Install express.
  + Create app.jsfile.
  + Define API’s

### Configure MongoDB:

* + Install Mongoose.
  + Create databaseconnection.
  + Create Models.

### Implement API end points:

* + Implement CRUD operations.
  + Test API endpoints.

## Milestone 3: Web Development:

### Setup AngularApplication:

* 1. Create Angular application using angular CLI.
  2. Configure Routing.
  3. Install required libraries.

### Design UI components:

* 1. Create Components.
  2. Implement layout and styling.
  3. Add navigation.

### Implement frontendlogic:

* 1. Integration with API endpoints.
  2. Implement data binding.