

A
Project Report
On
Job Portal

*Submitted In Partial Fulfillment of the
Requirement For the Degree of*

Bachelor of Technology
in
Computer Science & Engineering
Submitted by

Khushi 21118038

Pallavi 21118039

Tanu 21118040

Lalit 21118041

Lavi 21118042

Under the Supervision of
Mr. Bhupendra Kumar
(Assistant Professor)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
SCHOOL OF COMPUTER SCIENCE & APPLICATIONS,
IFTM UNIVERSITY LODHIPUR RAJPUT MORADABAD, UP
May 2025

DECLARATION

I hereby declare that the work, which is being presented in the project report, titled “**JOB PORTAL**” Submitted to the **Department of Computer Science and Engineering**, School of Computer Science & Applications, IFTM University Lodhipur, Rajput, Delhi Road, Moradabad for the award of degree of “**Bachelor of Technology**” of the University or other institute of higher learning except where due acknowledgement has been made in the text.

Name: Pallavi
Roll Number: 21118039

Name: Khushi
Roll Number: 21118038

Name: Tanu
Roll Number: 21118040

Name: Lalit
Roll Number: 21118041

Name: Lavi
Roll Number: 21118042



आईएफटीएम विश्वविद्यालय, मुरादाबाद, उत्तर प्रदेश
IFTM University, Moradabad, Uttar Pradesh
NAAC ACCREDITED

CERTIFICATE

This is to certify that this report on the project titled "Job Portal" is a bonafide record of the final project work carried out by **Pallavi, Tanu Ved, Khushi, Lavi, and Lalit Kumar** during the eighth semester of the academic year 2024–2025 in the Department of Computer Science and Engineering, IFTM University, Moradabad, in partial fulfillment of the requirements for the award of the B.Tech degree from IFTM University, Moradabad.

Mr. Bhupendra Kumar
Project In charge
Department of Computer Science and Engineering

Dr. Devender Singh
Head of Department
Department of Computer Science and Engineering

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B.Tech. Final Year. We owe special debt of gratitude to Assistant Professor Bhupender Sir, Department of Computer Science & Engineering, School of Engineering & Technology, IFTM University, Moradabad for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Dr. Devendra Singh, Head, Department of Computer Science & Engineering, IFTM University Moradabad for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

I wanted to express my sincere appreciation for the excellent service and quality of products provided by your e-commerce website for furniture. The seamless browsing experience, extensive selection, and hassle-free purchasing process truly made my shopping experience enjoyable and convenient.

From stylish designs to sturdy craftsmanship, your range of furniture offerings exceeded my expectations. Additionally, the efficient delivery and professional customer service further underscored your commitment to customer satisfaction.

Thank you for setting the standard for excellence in the online furniture industry. I look forward to continuing to support your business and recommending it to friends and family

ABSTRACT

The **Job Portal** is an online platform designed to connect job seekers with employers, enabling both parties to find and engage with each other in a streamlined manner. The portal provides an intuitive user interface that allows employers to post job vacancies, manage applications, and search for suitable candidates. Similarly, job seekers can browse available job listings, submit applications, and create a profile to attract potential employers.

This system is designed to handle a variety of job-related processes efficiently and effectively, ensuring that users have a smooth experience whether they are applying for jobs, recruiting candidates, or managing job postings.

The core features include:

User Management: Job seekers and employers can create and manage profiles. Job seekers can update their resumes, track applications, and receive job recommendations, while employers can post job openings and manage applications.

Job Posting and Search: Employers can post detailed job descriptions, including required skills, qualifications, experience, salary range, and work mode (e.g., remote, hybrid, in-office). Job seekers can search and filter jobs based on different criteria such as location, industry, job title, and salary.

Application Process: Job seekers can apply for positions through the portal, submitting their resumes and personal information. Employers can view and manage applications, shortlist candidates, and communicate directly with them through the system.

Admin Control Panel: Administrators have the ability to manage the entire platform, including verifying job posts, reviewing user activity, and ensuring the system is running smoothly.

Security and Privacy: The portal ensures user privacy and data protection through secure authentication, authorization mechanisms, and encrypted communication. Sensitive information such as resumes, job applications, and employer details are stored securely.

Responsive Design: The portal is built with a responsive design to support users on various devices, such as desktops, tablets, and smartphones, ensuring accessibility and ease of use from any location.

Purpose:

The purpose of the Job Portal is to provide an integrated digital platform that connects job seekers with potential employers, streamlining the entire recruitment lifecycle from job posting to hiring. This system is designed to address the inefficiencies and time-consuming nature of traditional recruitment methods by offering a centralized, automated, and accessible solution for both parties. For job seekers, the portal offers a personalized

dashboard where users can create and manage professional profiles, upload resumes, search and filter job listings based on their preferences, receive automated job recommendations, and apply directly to relevant opportunities. It empowers candidates by providing timely updates on application status, interview scheduling, and communication with recruiters, thereby improving the overall job search experience.

For employers and recruiters, the portal provides advanced features such as job posting management, applicant tracking, automated resume screening, and real-time communication tools. Employers can filter applicants based on predefined criteria like skills, experience, and location, significantly reducing the time to shortlist suitable candidates. Additionally, analytics and reporting tools offer insights into hiring metrics, helping HR departments make data-driven recruitment decisions.

The Job Portal System aims to foster a transparent, inclusive, and accessible job market. It supports different industries and job types—from internships and freelance gigs to full-time and remote positions. With built-in security and privacy protocols, it ensures that user data is handled safely and responsibly.

In essence, the portal serves as a bridge between talent and opportunity, enhancing the recruitment process's efficiency, accuracy, and user experience. It is built to evolve with user needs, technological advancements, and market trends, positioning itself as a scalable and sustainable solution for modern hiring challenges.

By integrating cutting-edge technologies such as artificial intelligence and machine learning, the portal continuously refines its job matching algorithms and personalized recommendations to better align candidates with relevant opportunities. Moreover, it encourages continuous feedback and community engagement to improve platform features and usability. This adaptive and user-centric approach ensures the Job Portal remains a future-ready ecosystem that supports diverse career paths, promotes equal opportunity, and empowers users to achieve their professional goals with confidence.

The system also incorporates seamless integration with social media and professional networking sites to enhance user engagement and broaden visibility for both job seekers and employers. Additionally, the platform offers customizable notifications and alerts, ensuring that users never miss important updates or deadlines.

Furthermore, the Job Portal promotes skill development by providing access to resources such as online courses, webinars, and career advice, helping candidates improve their employability and stay competitive in the evolving job market.

With a focus on accessibility, the portal is optimized for multiple devices and supports various assistive technologies, making it inclusive for users with

disabilities. Its modular design allows easy customization and integration with existing HR systems, making it a flexible solution for organizations of all sizes. Job Portal also emphasizes data-driven decision-making by leveraging predictive analytics to forecast hiring trends and identify skill gaps in the market. This empowers employers to proactively adjust their recruitment strategies and helps job seekers align their skills with emerging demands.

User privacy and compliance with global data protection regulations are paramount, with robust encryption and consent management features ensuring trust and transparency throughout the platform. Moreover, the portal supports multilingual interfaces to cater to a diverse, global user base, breaking down language barriers and fostering greater participation across regions. The platform also facilitates collaboration between recruiters and hiring managers through shared workspaces, enabling efficient team-based evaluation and decision-making.

Technologies Used:

The Job Portal System is developed using modern and robust web technologies to ensure reliability, scalability, and an excellent user experience, including:

- **ASP.NET** for powerful and secure server-side development
- **SQL SERVER** for efficient and reliable database management
- **HTML5, CSS3, JAVASCRIPT** for building dynamic and interactive frontend interfaces
- **BOOTSTRAP** for creating responsive, mobile-friendly designs that adapt seamlessly across devices

These technologies combine to deliver a fast, secure, and user-friendly platform capable of handling complex recruitment workflows.

Key Features of Job Portal:

1. Admin

- **Job Post Moderation:** Admins verify and approve job postings to ensure content quality and compliance.
- **Reporting & Analytics:** Admins can view platform statistics, such as the number of active users, job applications, and popular job categories.
- **Access Control:** Admins have full access to all features and can assign roles or permissions to other users.

2. Job Seeker

- **Profile Creation:** Job seekers can create and update their profiles, including uploading resumes, educational details, and work experience.
- **Job Search & Apply:** Job seekers can search for job listings using filters (location, salary, role) and apply for jobs directly on the portal.

3. Job Recruiter (Employer)

- **Job Posting:** Recruiters can post job openings, including job descriptions, salary details, qualifications, and requirements.
- **Applicant Management:** Recruiters can view, shortlist, and communicate with applicants through the portal.
- **Job Listing Management:** Edit or remove job listings, monitor how many people have applied, and manage vacancies.

TABLE OF CONTENTS

Contents	Page no.
Declaration	i
Certificate	ii
Abstract	iii-iv
Acknowledgment	v
CHAPTER 1 INTRODUCTION	1
1.1 Project Detail	1
1.2 Hardware and Software	7
CHAPTER 2 TECHNICAL FEASIBILITY	8
2.1 Technical Feasibility	8
2.2 Technology Description	9
2.2.1 HTML	9
2.2.2 CSS	10
2.2.3 BOOTSTRAP	11
2.2.4 MY SQL	13
2.2.5 JavaScript	15
2.2.5 ASP.NET	16
CHAPTER 3 ER DIAGRAM	19
3.1 E R Diagram	19
CHAPTER 4 BACKEND DESIGN	20
4.1 Database Tables	20
CHAPTER 5 FRONTEND DESIGN	21
5.1 Home Page	22
5..2 Login Form	22
5.3 Dashboard	23
5.4 Admin Panel	23
5.5 Jobseeker Pane	24

5.6 JobRecuriter Panel	25
CHAPTER 6 ADVANTAGE AND LIMITATION	26
7.1 Advantages	27
7.1 Disadvantages	27
CHAPTER 7 FUTURE SCOPE	25
8.1 Future Scope	29
CHAPTER 8 CONCLUSION	30
APPENDIX	34
9.1 Code	34-49
9.1 REFERENCE	50

LIST OF FIGURE's

Contents	Page no.
1. E-R Diagram	9
2. Home Page	11
3. Login Page	11
4. Job Seeker Page	12
5. Recruiter Page	12
6. Admin Page	13
7.Admin JobMange	13

CHAPTER 1

INTRODUCTION

1.1 Project Detail

A Job Portal is an online platform that connects job seekers with potential employers, serving as a vital link in the modern recruitment ecosystem. This portal allows employers to post job vacancies across various industries and job categories, while job seekers can conveniently browse listings, apply for suitable positions, and manage their professional profiles and resumes—all within a single, user-friendly interface. By centralizing these functions, the system streamlines and simplifies the recruitment process, reducing the time and effort traditionally required for job searching and hiring.

The platform is designed to support multiple user roles, each with distinct capabilities to ensure smooth operation and a tailored experience:

- **Admin:** The Admin plays a critical role in managing the overall health and security of the portal. They oversee user account management, including the registration and verification of job seekers and recruiters. Admins also handle content moderation by reviewing and approving job postings to maintain the quality and relevance of listings. Additionally, they monitor platform activity through analytics and reports, and manage user permissions to ensure the system remains secure and compliant with policies.
- **Job Seeker:** Job seekers use the portal to create and maintain personalized profiles, including uploading resumes, listing skills, and detailing work experience and educational background. They can search for jobs using advanced filters such as location, salary, job type, and industry, making it easier to find opportunities that match their preferences and qualifications. The portal allows direct application to jobs, and job seekers receive timely notifications on application status, interview schedules, and employer communications, enhancing their engagement and job search efficiency.
- **Job Recruiter:** Recruiters benefit from a comprehensive suite of tools to manage the hiring process. They can post new job openings with detailed descriptions and requirements, update or remove listings as needed, and track applicant progress through integrated applicant tracking features. Recruiters can shortlist candidates based on skills and experience, communicate directly with

applicants via the platform's messaging system, and utilize analytics to refine their recruitment strategies and optimize hiring outcomes.

Key Features of the Job Portal:

Admin Features:

- **User Management:** Admins have the ability to create, delete, and update user profiles for both job seekers and recruiters, ensuring the platform remains secure and well-organized.
- **Job Post Management:** Admins review, approve, or reject job postings to maintain quality control and compliance with platform guidelines.
- **Analytics and Reporting:** Comprehensive dashboards allow admins to monitor portal activity, including user registrations, job postings, application volumes, and engagement trends.
- **Role-Based Access Control:** Admins assign and manage different access levels and permissions to maintain secure operations.
- **Content Moderation:** Ensure that all posted content adheres to ethical standards, preventing spam or fraudulent listings.
- **System Maintenance:** Manage backend configurations, perform backups, and update system settings to ensure continuous platform stability.

Job Seeker Features:

- **Profile Creation and Management:** Job seekers can build detailed profiles, including work history, education, skills, certifications, and personal details, which can be updated anytime.
- **Resume Upload and Builder:** Users can upload existing resumes or use integrated resume-building tools to create professional CVs directly on the platform.
- **Job Search with Advanced Filters:** Use multiple search filters such as job location, salary range, job type (full-time, part-time, remote), industry, and

experience level to find tailored job opportunities.

- **Job Application Management:** Track application statuses in real-time, receive notifications about interview calls or feedback, and manage submitted applications through a centralized dashboard.
- **Saved Jobs and Alerts:** Save interesting job listings for future reference and set up email or SMS alerts for new job postings matching specific criteria.
- **Career Resources:** Access career advice, interview tips, and skill development resources to enhance employability.

Job Recruiter Features:

- **Job Posting and Management:** Create, edit, or remove job listings with detailed descriptions, qualifications, salary packages, and application deadlines.
- **Applicant Tracking System (ATS):** Track and organize candidate applications, shortlist potential hires, and schedule interviews all within the portal.
- **Resume Screening:** Automated filtering of applicants based on predefined criteria such as skills, education, and experience to streamline candidate selection.
- **Communication Tools:** Communicate with applicants through built-in messaging or email notifications, ensuring seamless recruitment correspondence.
- **Reporting and Insights:** Generate reports on hiring metrics, application volumes, and candidate demographics to improve recruitment strategies.
- **Employer Branding:** Customize company profiles with logos, descriptions, and links to social media to attract top talent.
- **Multi-User Collaboration:** Allow HR teams to collaborate on recruitment efforts by sharing notes, ratings, and feedback on candidates.

Technologies:

Frontend Technologies:

- **HTML5:** Used for creating the structural foundation of web pages, ensuring semantic markup and accessibility standards.
- **CSS3:** Enhances the visual aesthetics of the portal by applying styles, layouts, animations, and ensuring responsiveness for a seamless user experience across desktops, tablets, and mobile devices.
- **JavaScript:** Adds dynamic functionalities such as real-time form validation, interactive job search filters, notifications, and client-side logic for faster user interactions.
- **Bootstrap:** A popular CSS framework integrated to facilitate responsive design and pre-built UI components, speeding up development and ensuring a consistent look across devices.
- **AJAX:** Enables asynchronous data loading, improving user experience by updating parts of the webpage without full reloads, for example, fetching job listings dynamically.

Backend Development:

Backend Frameworks:

- **ASP.NET Core (C#):** A robust, cross-platform framework used for developing high-performance, scalable, and secure web applications. It supports MVC architecture, REST API creation, and seamless integration with Microsoft Azure cloud services.
- **Entity Framework Core:** An ORM (Object-Relational Mapper) used alongside ASP.NET Core to simplify database operations by working with .NET objects instead of raw SQL queries.

Database Technologies:

Databases are crucial for storing and managing all platform data, including user profiles, job listings, applications, and analytics.

- **MySQL / PostgreSQL:** Reliable and widely-used open-source relational databases that handle structured data efficiently, offering scalability and data integrity for job portal applications.
- **Microsoft SQL Server:** A powerful commercial relational database system that integrates well with ASP.NET applications, providing advanced features such as in-built analytics, security, and backup solutions.
- **Caching Mechanisms:** Technologies like Redis or Memcached may be

employed to cache frequent queries and improve the portal's response time.

1.2 Hardware/Software Used

Hardware Requirements:

- **Processor:** Intel 64-bit architecture or equivalent, ensuring compatibility with modern operating systems and development tools.
- **Hard Disk:** Minimum of 80 GB HDD/SSD to store system files, database, and application data efficiently. SSD preferred for faster read/write operations.
- **RAM:** At least 512 MB for basic operations; ideally 4 GB or more for smooth multitasking during development and deployment.

Software Specifications:

- **Operating System:** Windows 11, providing a stable and modern environment with support for development tools like Visual Studio and SQL Server.
- **Frontend Tools:** HTML, CSS, Bootstrap for building responsive and user-friendly interfaces.
- **Backend Tools:** ASP.NET Core framework running on Visual Studio IDE, which offers advanced debugging, code completion, and integration with Git version control.
- **Database Management:** MySQL or Microsoft SQL Server, managed through tools like SQL Server Management Studio (SSMS) or MySQL Workbench for efficient database administration.
- **Version Control:** Git for source code management and collaboration, integrated within Visual Studio or external tools like GitHub.
- **Other Utilities:** Postman or REST for API testing, and browser developer tools for frontend debugging.

CHAPTER 2

TECHNICAL FEASIBILITY

2.1 Technical feasibility

Technical feasibility is one of the earliest and most critical assessments conducted after identifying a project. It evaluates whether the project's technical resources, skills, and infrastructure can meet the requirements for successful implementation. For large-scale engineering projects, specialized consulting agencies with teams of engineers and technicians usually perform these studies, ensuring that every technical aspect—from design and materials to processes and production—is viable.

In smaller or more specialized fields, such as agricultural projects financed by local credit organizations, technical feasibility studies are carried out by dedicated staff including agricultural engineers, irrigation experts, and construction technicians. These experts assess whether the proposed project can be realistically executed with available technologies and resources.

The initial phase of technical feasibility involves developing a working model or prototype of the product or service. This prototype need not use the exact materials or components intended for the final product; rather, it is created to demonstrate the core functionality and production viability. This working model serves as a tangible proof-of-concept, helping stakeholders visualize the idea and gain confidence in its feasibility.

In the case of physical products, mechanical working models are often easier to conceptualize and present. However, for software, e-commerce, or service-based solutions, technical feasibility involves more complex assessments. For instance, e-commerce projects require verifying the ability to integrate various technologies, including computer hardware, servers, databases, software platforms, and programming languages, to support the operational framework efficiently.

Service-oriented projects, which often package a series of value-added activities, must ensure that these services can consistently deliver measurable and observable benefits to end users. This means evaluating the technical capacity to manage workflows, deliver services reliably, and maintain quality standards throughout the lifecycle.

Key considerations in technical feasibility include:

- Availability of technical expertise and skilled personnel.
- Access to necessary hardware, software, and network infrastructure.
- Compatibility with existing systems or technologies.
- Scalability potential to accommodate future growth.
- Identification of potential technical risks and mitigation strategies.
- Assessment of development timelines relative to technical constraints.

A thorough technical feasibility study minimizes risks and lays a solid foundation for subsequent stages of project development, such as financial feasibility and market analysis, thereby increasing the likelihood of project success.

2.2 Technology Description

2.2.1 HTML

HTML stands for **HYPER TEXT MARKUP LANGUAGE**, which is the most widely used language on the web to develop web pages. HTML refers to the way in which web pages (HTML documents) are linked together. Thus, the link available on a web page is called **Hypertext**. HTML was created by Tim Berners-Lee in late 1991, but “**HTML 2.0**” was the first standard HTML specification, published in 1995. **HTML 4.01** was a major version released in late 1999 and became widely adopted. Though HTML 4.01 was dominant for many years, we currently use **HTML5**, an extension of HTML 4.01, which was officially published in 2012.

As its name suggests, HTML is a **markup language**, which means you use HTML to “mark up” a text document with tags that tell a web browser how to structure and display the content. Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format and design web pages with the help of many different tags and attributes available in the language. HTML5 introduced many new features including support for multimedia (audio and video), graphics (canvas and SVG), new form controls, semantic elements (like `<article>`, `<section>`, and `<nav>`), and APIs for better interactivity and responsiveness. These enhancements have made web pages more dynamic and user-friendly without relying heavily on external plugins.

Moreover, HTML works closely with **CSS** (Cascading Style Sheets) and **JavaScript** to create rich and visually appealing websites that are responsive across various devices such as desktops, tablets, and smartphones. Its role as the backbone of web development makes understanding HTML fundamental for anyone looking to create or manage web content.

2.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, CSS can be applied to any XML document, including plain XML, SVG (Scalable Vector Graphics), and XUL (XML User Interface Language). It is also applicable to rendering in speech or on other media types such as print and screen readers.

Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. By controlling the layout, colors, fonts, and overall visual appearance, CSS allows developers and designers to enhance the user experience and ensure consistent branding across multiple platforms.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as layout, colors, and fonts. This separation improves content accessibility by making it easier for assistive technologies to interpret and navigate the content. It also provides more flexibility and control in specifying presentation characteristics and enables multiple HTML pages to share formatting by referencing a single external CSS file. This reduces redundancy and complexity in web development, leading to cleaner, more maintainable code.

The CSS specifications are maintained by the World Wide Web Consortium (W3C), which continually updates the standards to incorporate new features and improve browser compatibility. The Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). To ensure adherence to standards, the W3C operates a free CSS validation service that checks CSS documents for syntax errors and best practices.

CSS has a straightforward syntax and uses a variety of English keywords to specify style properties such as color, margin, padding, font-size, and many others. A style sheet consists of a list of rules, where each rule or rule-set is composed of one or more selectors and a declaration block. Selectors define which HTML elements the styles will apply to, while the declaration block contains one or more declarations specifying the property and value to be applied.

With the advent of CSS3, the language introduced many advanced features such as animations, transitions, flexible box layouts (Flexbox), grid layouts, media queries for responsive design, and variables for easier theme management. These powerful tools allow developers to create complex, dynamic layouts that

adapt seamlessly to different screen sizes and devices, making CSS an essential part of modern web design and development.

2.2.2 **Bootstrap**

Bootstrap is a free and open-source front-end framework used for designing responsive and mobile-first websites and web applications. It provides a comprehensive collection of **HTML and CSS-based design templates** for typography, forms, buttons, navigation, modals, carousels, and many other user interface components, along with optional JavaScript extensions to enhance interactivity.

Unlike many web frameworks that cover both front-end and back-end development, Bootstrap focuses solely on **front-end development**, making it lightweight and easy to integrate with various backend technologies. Its modular architecture consists of a series of **Sass (previously Less) stylesheets** that implement different components of the toolkit. Developers can include the entire Bootstrap library or selectively include only the components they need, which helps in optimizing page load times and performance.

Bootstrap provides a powerful **customization system** through its configuration variables, allowing developers to easily modify colors, padding, margins, fonts, and other design elements to fit their project's branding and style guidelines. This level of customization ensures that Bootstrap-based sites can maintain a unique look and feel while leveraging the consistency and responsiveness the framework offers.

Since Bootstrap 2, the official documentation has featured a **customization wizard** that allows users to generate a tailored version of Bootstrap by selecting specific components and adjusting settings such as grid breakpoints, container widths, and component behaviors. This wizard helps developers build streamlined CSS and JavaScript bundles that suit their project requirements.

With the release of **Bootstrap 4**, Sass replaced Less as the preprocessor for stylesheets, providing more advanced features, better performance, and a larger ecosystem of Sass tools. Bootstrap 4 also introduced improvements like a new flexbox-based grid system, enhanced utility classes, and better support for responsive design.

Each Bootstrap component is a combination of three key elements:

- An **HTML structure**, which defines the markup and layout.
- Corresponding **CSS declarations** that style the component according to Bootstrap's design system.
- **JavaScript code** (optional in many cases), which adds dynamic behavior such as dropdown toggling, modals, tooltips, and carousels.

Bootstrap's widespread popularity and active community support have led to

numerous third-party themes, templates, and plugins, making it a highly versatile framework for rapid front-end development. Its built-in responsiveness ensures websites are visually appealing and functional across all device types and screen sizes, from desktops to smartphones.

Overall, Bootstrap significantly accelerates the web development process, reduces the need for custom CSS coding, and ensures design consistency, making it an ideal choice for developers building modern, responsive web applications.

2.2.3 My SQL

MySQL is an open-source Relational Database Management System (RDBMS) that relies on Structured Query Language (SQL) for managing and manipulating data within databases. It provides robust APIs for multiple programming languages, including C, C++, Eiffel, Java, Perl, PHP, and Python, making it highly versatile and easy to integrate into diverse application environments.

MySQL is widely used for web applications and embedded applications due to its speed, reliability, and ease of use, making it a popular alternative to proprietary database systems. It supports multiple operating systems such as UNIX, Windows, and Mac OS, allowing developers to deploy it across a wide variety of platforms.

One of the critical advantages of MySQL is its integration with open-source technologies, particularly in the PHP/MySQL stack, which powers many dynamic websites and content management systems such as WordPress, Joomla, and Drupal. These platforms leverage MySQL's capabilities to efficiently store, retrieve, and manage content, user data, and application states.

MySQL's scalability and flexibility allow it to handle everything from small personal projects to large-scale enterprise applications with millions of transactions per day. It supports numerous storage engines, including InnoDB, which provides ACID-compliant transactions, foreign keys, and crash recovery, ensuring data integrity and consistency.

Due to its open-source nature, MySQL has a large and active community that continuously contributes to its development, offering extensive documentation, tools, and third-party extensions that enhance its functionality. It also benefits from strong commercial support through Oracle, which maintains and offers enterprise editions with advanced features.

MySQL is a fundamental part of the widely-used LAMP stack (Linux, Apache, MySQL, PHP/Python/Perl), which forms the backbone of many web hosting environments and enterprise solutions. Its performance in handling large volumes of data efficiently and securely makes it an essential component for web-based software applications, online publishing platforms, and e-commerce systems.

For effective MySQL deployment, it is important to choose a specialized hosting environment optimized for database operations, which can ensure high availability, performance tuning, and secure data management.

In summary, MySQL remains one of the most popular and trusted open-source RDBMS choices worldwide, thanks to its cross-platform support, robust performance, extensive language APIs, and active community backing, making it indispensable in modern web and enterprise development.

2.2.4 JavaScript

JavaScript is one of the most popular and widely used programming languages in the world today. It is primarily known as a client-side scripting language that runs in web browsers, but with the advent of Node.js, it is now also widely used on the server-side for building scalable, high-performance network applications.

Key Features of JavaScript:

Client-Side Scripting:

JavaScript was originally designed as a client-side language, meaning it runs directly in the user's web browser. This enables developers to build highly interactive and dynamic web pages that can respond instantly to user actions—such as mouse clicks, keyboard inputs, and other events—without requiring the page to reload. This ability dramatically improves the user experience on the web.

Event-Driven Programming:

JavaScript operates on an event-driven model. This means the flow of a program is largely dictated by events, including user interactions, timers, or asynchronous server responses. The event-driven approach allows developers to build responsive applications that can handle multiple tasks simultaneously through callbacks, promises, and `async/await` constructs.

Lightweight and Fast:

As a lightweight scripting language, JavaScript is designed to execute efficiently and quickly within the browser environment. Since it is interpreted rather than compiled, it can be executed on-the-fly, making it ideal for rapid development and iterative testing in web development.

Dynamic Typing:

JavaScript uses dynamic typing, meaning variables can hold values of any data type without needing explicit type declarations. This flexibility accelerates coding but also requires developers to be vigilant to prevent type-related errors during execution.

Versatile and Powerful:

Beyond simple form validation and animations, JavaScript supports advanced functionality like creating single-page applications (SPAs), real-time communication through WebSockets, and integration with backend services using AJAX and Fetch APIs. It also powers frameworks and libraries such as React, Angular, and Vue.js, which streamline the development of complex user interfaces.

Supports Multiple Programming Paradigms:

JavaScript supports object-oriented, functional, and imperative programming styles. Developers can use prototypes or ES6 classes for OOP, leverage higher-order functions and closures for functional programming, or write straightforward imperative code, making JavaScript adaptable to various project requirements and coding preferences.

How JavaScript Works:

JavaScript code runs inside the browser's JavaScript engine, like Google Chrome's V8 or Mozilla Firefox's SpiderMonkey. When a web page loads, the browser fetches the HTML, CSS, and JavaScript files from the server. The JavaScript engine then interprets the script line by line, executing commands and manipulating the Document Object Model (DOM) to update the web page dynamically. Because it's an interpreted language, developers can see changes immediately without a lengthy compile step, enabling quick prototyping and debugging.

Asynchronous Programming:

JavaScript's event loop and asynchronous capabilities allow it to perform non-blocking operations. Functions like `setTimeout`, `Promises`, and `async/await` let developers write code that waits for external data or operations (like API calls) without freezing the user interface, ensuring smooth and responsive web experiences.

Security Considerations:

Since JavaScript runs on the client side, it is exposed to potential security risks such as cross-site scripting (XSS) attacks. Developers must follow best practices to sanitize inputs, handle data securely, and implement appropriate content security policies.

Tooling and Developer Environment:

Modern JavaScript development is supported by an extensive array of tools that improve productivity and code quality. These include transpilers like Babel (which allow developers to use the latest syntax while maintaining compatibility with older browsers), bundlers like Webpack or Rollup (which package multiple files into optimized bundles), and linters like ESLint (which enforce coding standards). Integrated Development Environments (IDEs) such as Visual Studio Code offer rich JavaScript support, including debugging, code completion, and version control.

Community and Resources:

The JavaScript community is vast and vibrant, continuously contributing to open-source projects, libraries, and frameworks. Resources like MDN Web Docs, Stack Overflow, and countless tutorials make learning and mastering JavaScript accessible for developers of all skill levels.

Cross-Browser Compatibility:

Historically, JavaScript developers faced challenges due to inconsistencies across different browsers. Modern standards and transpilers help smooth these differences, but developers still test code across browsers like Chrome, Firefox, Safari, and Edge to ensure consistent behavior.

Future Trends:

JavaScript continues to evolve with ongoing ECMAScript proposals aiming to add new features and syntactic sugar that improve developer experience. Its role in emerging fields like WebAssembly (for running near-native code in browsers) and progressive web apps (PWAs) ensures it remains a cornerstone of modern application development.

2.2.5 ASP.NET:

ASP.NET is a free, open-source web framework developed by Microsoft for building modern web applications, APIs, and dynamic websites. It is part of the larger .NET ecosystem and enables developers to create web applications using languages like C# and Visual Basic. ASP.NET provides powerful features for both frontend and backend development, making it one of the most popular frameworks for building robust and scalable web applications.

ASP.NET Core follows a modular architecture, allowing developers to include only the necessary components through NuGet packages. This results in leaner applications with better performance and easier maintenance. The framework supports middleware pipelines that let developers customize request handling, routing, and response generation with great flexibility.

RichDevelopmentTools:

ASP.NET integrates seamlessly with Visual Studio and Visual Studio Code, providing an excellent developer experience. Features like IntelliSense, debugging, live reload, and integrated testing streamline development and help catch errors early. Additionally, the command-line interface (CLI) tools enable quick scaffolding, building, and deployment of ASP.NET projects.

SecurityFeatures:

Security is a critical aspect of web applications, and ASP.NET provides comprehensive built-in security features. These include authentication and authorization frameworks, protection against common web vulnerabilities like Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF), data encryption, and secure cookie handling. ASP.NET Identity allows easy integration with external login providers such as Google, Facebook, and Microsoft accounts.

SupportforModernWebStandards:

ASP.NET supports RESTful APIs, WebSockets for real-time communication, and integration with frontend frameworks like Angular, React, and Blazor. Blazor, a newer addition, allows developers to build interactive client-side web UIs using C# instead of JavaScript, opening new possibilities for .NET developers.

EntityFrameworkCoreIntegration:

For database interaction, ASP.NET works smoothly with Entity Framework Core (EF Core), an Object-Relational Mapper (ORM) that facilitates data access with strongly typed LINQ queries. EF Core supports multiple database providers

including SQL Server, MySQL, PostgreSQL, and SQLite.

OpenSourceandCommunityDriven:

Since ASP.NET Core became open source, it has attracted a large community of contributors and enthusiasts. This has accelerated innovation and provided transparency in development. Microsoft regularly updates the framework, incorporating feedback from the community and evolving the platform to meet modern development needs.

ExtensiveEcosystem:

ASP.NET benefits from the rich .NET ecosystem, including libraries for logging, caching, dependency injection, and testing. NuGet package manager provides thousands of libraries that extend ASP.NET functionality, enabling rapid application development.

Cloud-Ready:

ASP.NET is designed with cloud deployment in mind. It integrates easily with Microsoft Azure services, supports containerization with Docker, and works well with Kubernetes for orchestration. This makes deploying, scaling, and managing ASP.NET applications in the cloud straightforward and efficient.

Key Features of ASP.NET:

Cross-Platform Support:

One of the most significant advantages of ASP.NET Core (the latest version of ASP.NET) is its cross-platform support. ASP.NET Core can run on Windows, macOS, and Linux, allowing developers to deploy web applications on any major platform.

Performance and Scalability:

ASP.NET applications are known for their high performance due to the optimized runtime (Kestrel web server) and compiled code. It is highly scalable and can handle a large number of requests per second, making it suitable for enterprise-level applications.

UnifiedFrameworkforWebandAPIs:

ASP.NET Core provides a unified programming model to build both web applications and RESTful APIs. This enables developers to reuse code and leverage consistent patterns across different types of applications.

Built-inDependencyInjection:

ASP.NET Core has native support for dependency injection, which enhances modularity, testability, and maintainability of applications by promoting loose coupling between components.

RobustRoutingandMiddleware:

The flexible routing system allows clean, SEO-friendly URLs and precise control over URL mapping. The middleware pipeline provides a modular approach to handle cross-cutting concerns such as authentication, logging, and error handling efficiently.

SecurityFeatures:

ASP.NET offers comprehensive security support, including built-in authentication and authorization mechanisms, integration with OAuth, OpenID Connect, JWT tokens, and ASP.NET Identity for user and role management.

SupportforModernFrontendFrameworks:

ASP.NET works seamlessly with frontend technologies like Angular, React, and Vue.js, allowing developers to build dynamic, single-page applications (SPAs) with a powerful backend.

OpenSourceandCommunity-Driven:

As an open-source framework, ASP.NET Core benefits from a large and active community that contributes to its continuous improvement, regular updates, and support for the latest web standards.

CHAPTER: 3

ER DIAGRAM

3.1 ER Diagram

The Entity-Relationship Data Model (ERD) perceives the real world as consisting of basic objects, called entity & relationship among these objects. It was developed to facilitate database design by allowing specification of an enterprise schema, which represents overall logical structure of a database. The ERD model is very useful in mapping the meaning & interactions of the outside world enterprises onto a conceptual schema.

The ERD model consists of the following major components

- **ELLIPSE** which represents attributes.
- **RECTANGLES** which represents entity-sets.
- **DIAMONDS** which represents the relationship sets.
- **LINES** which link attributes to entity sets to relationship sets.

ER-DIAGRAM:-

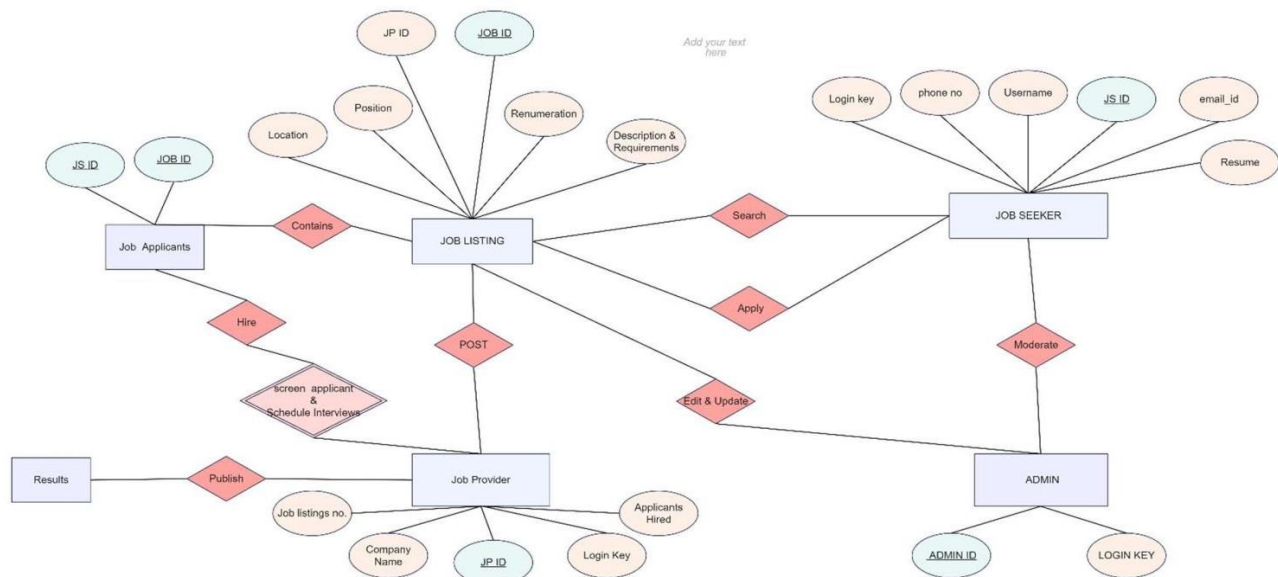


Fig. 3.1 ER Diagram

CHAPTER: 4

BACKEND DESIGN

4.1 Database Tables

When building a web application using the **ASP.NET Framework**, one of the essential parts of backend design is database modeling. This includes creating tables that define how the data will be stored, retrieved, and related to each other. In this example, we will focus on a **Job Portal** backend, which involves three primary modules: **Admin**, **Jobseeker**, and **Job Recruiter**.

Below is an example of how you might design the database tables for these modules using **ASP.NET Entity Framework**. Entity Framework (EF) allows you to define database tables as **C# classes**, and it automatically handles the database creation and querying.

1. Database Tables

a. Admin Table

- **Admin** table stores information about the administrators of the portal.

Columns:

- Admin_ID (Primary Key): A unique identifier for each admin.
- Admin_Name: Name of the admin.
- Admin_Email: Email of the admin.
- Admin_Password: Password for authentication.

b. Jobseeker Table

- **Jobseeker** table stores information about the job seekers.

Columns:

- Jobseeker_ID (Primary Key): A unique identifier for each jobseeker.
- Name: Name of the jobseeker.
- Email: Email address of the jobseeker.
- Skills: A comma-separated list of skills of the jobseeker.
- Resume: A file path or URL pointing to the resume of the jobseeker.

c. Job Recruiter Table

- **Recruiter** table stores information about the companies/recruiters who post jobs.

Columns:

- Recruiter_ID (Primary Key): A unique identifier for each recruiter.
- Company_Name: Name of the recruiting company.
- Email: Email address of the recruiter.
- Phone: Phone number of the recruiter.

CHAPTER : 5

FRONT-END DESIGN

Home page is the main page of the website .This page is tell about the information and look and feel of the website .This page is very important to make attractive. This page include all the features and functions of that website.The **Home Page** is the central entry point of the website and plays a critical role in guiding users to the platform's core features. It provides a **clear overview** of the site's purpose, sets the **visual tone**, and facilitates **easy navigation** for different user types.

Key Components and Functions of the Home Page:

1. Header Banner with Icons and Branding:

- At the top, a vibrant header with the title “**JOB SEARCH**” is prominently displayed.
- The banner features graphical icons and labels such as **Application, Career, Qualification, Talent, Search, Goal, Interview, Skills**, visually summarizing the job search journey.
- This section visually engages visitors and communicates the mission of the website — to help users navigate the employment process.

2. Navigation Bar:

- Just below the header is a green navigation bar with the following options:
- **JobSeeker Sign Up**
- **JobRecruiter Sign Up**
- **Sign In**
- This allows new users to register based on their role and existing users to log in.

3. User Login Form:

- Below the navigation bar, there is a **login form** that includes:
- **User Type Dropdown:** Allows the user to select from **ADMIN, JOBSEEKERS, or JOBRECRUITER**.
- **Email ID and Password Fields:** Standard credentials input for authentication.
- **Submit Button:** Triggers the login process based on the selected user type.

4. Footer Section:

- The footer contains copyright information:
“© 2024 Pallavi. All rights reserved.”
- It gives a professional finishing touch to the website and denotes authorship.

- **Functional Highlights:**

- **User Role Selection** ensures customized experiences for admins, jobseekers, and recruiters.
- The page is designed to be **intuitive and interactive**, helping users quickly understand where they fit and what actions to take next.
- **Responsive layout and organized elements** make the home page user-friendly and accessible.

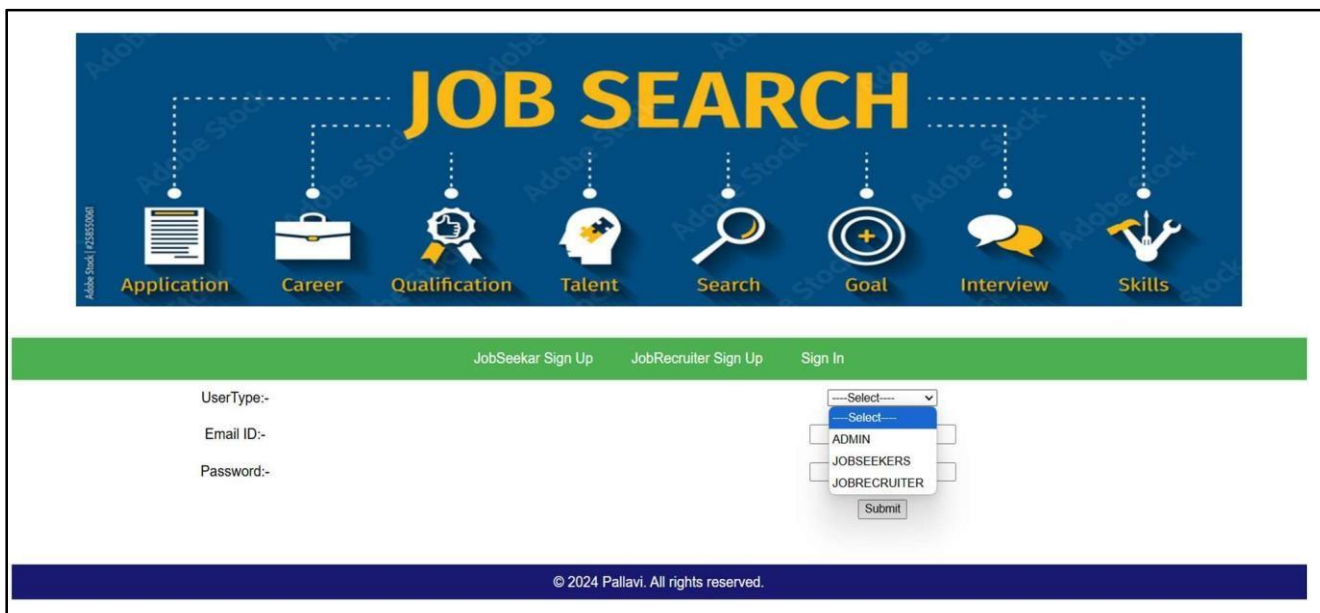


Fig 5.1 Home Page

The Login Function is a critical feature that enables authenticated access to the Job Search platform based on user roles. It ensures that users — whether Jobseekers, Job Recruiters, or Admins — can securely sign into the system and access functionalities tailored to their roles.

Purpose:

- To authenticate users using their email ID and password.
- To determine user roles (e.g., JOBSEEKERS, JOBRECRUITER, ADMIN) and redirect them to the respective dashboard or interface after successful login.

Key Components of the Login Page:

1. User Type Dropdown:

Allows the user to select their role:

- ADMIN
- JOBSEEKERS
- JOBRECRUITER

Helps in segregating user access and functionalities based on their selected role.

2. Email ID Field:

- Input field for users to enter their registered email address.
- Used as a unique identifier for user login.

3. Password Field:

- Secured input for users to enter their password.
- Used to validate user credentials against the stored password in the database

(often hashed for security).

4. Submit Button:

- Triggers the authentication process.
- If credentials are correct, the user is redirected to their respective dashboard.
- If credentials are incorrect, an error message is displayed, prompting users to retry.



The screenshot displays a web application interface for job searching. At the top, a blue banner features the text "JOB SEARCH" in large yellow letters. Below this banner is a row of eight icons with labels: "Application" (document), "Career" (briefcase), "Qualification" (diploma), "Talent" (head with gear), "Search" (magnifying glass), "Goal" (target), "Interview" (speech bubbles), and "Skills" (wrench and screwdriver). Below the icons is a green navigation bar with links: "JobSeeker Sign Up", "JobRecruiter Sign Up", and "Sign In". The main content area contains a login form with labels "UserType:-", "Email ID:-", and "Password:-". The "UserType" dropdown is set to "JOBSEEKERS". The "Email ID" field contains "pp@gmail.com" and the "Password" field contains "ppp". A "Submit" button is located below the password field. At the bottom, a dark blue footer contains the text "© 2024 Pallavi. All rights reserved."

Fig 5.2 LOGIN

[Home](#)
[ChangePassword](#)
[ShowJobs](#)
[Logout](#)

Recruiter Name	Job Profile	Work Mode	Gender	Recruiter Qualification	Recruiter Experince	offer salary	Vacancy	comment
google	.Net Developer	Work From Office	both	B.Tech,MCA,BCA	0 year -6year	Rs. 10000 - Rs.50000	2	hello
google	Testing	work From Home	Female	B.Tech,MCA	2 year -3year	Rs. 20000 - Rs.30000	2	hi
google	Php Developer	Work From Office	Female	M.Tech,B.Tech	2 year -3year	Rs. 10000 - Rs.30000	2	hello
google	Sales	Work From Office	Female	Diploma,Polytecnic	0 year -3year	Rs. 30000 - Rs.50000	2	need

© 2024 Pallavi. All rights reserved.

Fig 5.3 JOBSEEKER

[Home](#)
[JobPost](#)
[ShowJobPost](#)
[ChangePassword](#)
[Logout](#)

Recruiter Name	Job Profile	Work Mode	Gender	Recruiter Qualification	Recruiter Experince	offer salary	Vacancy	comment		
google	.Net Developer	Work From Office	both	B.Tech,MCA,BCA	0 year -6year	Rs. 10000 - Rs.50000	2	hello	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
google	Testing	work From Home	Female	B.Tech,MCA	2 year -3year	Rs. 20000 - Rs.30000	2	hi	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
google	Php Developer	Work From Office	Female	M.Tech,B.Tech	2 year -3year	Rs. 10000 - Rs.30000	2	hello	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
google	Sales	Work From Office	Female	Diploma,Polytecnic	0 year -3year	Rs. 30000 - Rs.50000	2	need	<input type="button" value="Delete"/>	<input type="button" value="Edit"/>

© 2024 Pallavi. All rights reserved.

Fig 5.5 RECURITER PAGE



Fig 5.6 ADMIN

The screenshot displays the MANGERECUITERPAGE interface. It features the same "JOB SEARCH" header and navigation menu as Fig 5.6. The main content area is a table with the following data:

Recruiter ID	Name	Email	Password	Company URL	State ID	City ID	Address	HR Name	Mobile Number	Image Path	Comments	Inserted Date	Status
3	TCS	TCS@GMAIL.COM	tcs	WWW.TCS.COM	2	18	Lucknow,Kanpur,Uttar Pradesh	Ansh	9643864369		hi	2024-10-05	0
4	TCS	TCS@GMAIL.COM	123TCS	WWW.TCS.COM	2	3	Lucknow,Lucknow,Uttar Pradesh	Ansh	8764237598		hi	2024-10-06	0
5	TCS	TCS@GMAIL.COM	T123	WWW.TCS.COM	2	17	Lucknow,Lucknow,Uttar Pradesh	Ansh	8764237598		hi	2024-10-06	0
6	TCS	TCS@GMAIL.COM	tit	WWW.TCS.COM	5	12	gujrat 23,Surat,Gujrat	swati	9643864369		hello	2024-12-01	0

At the bottom, a dark blue footer bar contains the copyright notice "© 2024 Pallavi. All rights reserved."

**Fig 5.7 ADMIN
MANGERECUITERPAGE**

CHAPTER:6

ADVANTAGES AND LIMITATION

ADVANTAGES

1. Convenience:

- **Centralized Access:** Job seekers can easily browse a wide variety of job listings from multiple employers in one unified platform, eliminating the need to visit individual company websites.
- **One-Stop Solution:** Employers can connect with thousands of potential candidates without investing heavily in physical recruitment drives or third-party hiring agencies.
- **User-Friendly Interface:** The login and navigation system is simple and intuitive, making it easy for users of all technical levels to use the platform efficiently.

2. Time-Saving:

- **Quick Application Process:** Candidates can apply for several job openings using saved resumes and cover letters, minimizing repetitive tasks.
- **Automated Shortlisting:** Employers can set criteria such as keywords, qualifications, or experience levels to automatically shortlist the most suitable candidates.
- **Instant Notifications:** Both job seekers and recruiters can receive instant alerts and updates on application status, interview schedules, or new job postings.

3. Customization:

- **Advanced Filters:** Users can filter results based on job role, experience, location, salary range, company type, employment type (full-time, part-time, freelance), and more.
- **Profile Personalization:** Candidates can build a professional profile showcasing their skills, achievements, portfolio, and work preferences, while recruiters can personalize job postings to target ideal candidates.

4. Networking Opportunities:

- **Professional Connections:** Platforms with social features (like LinkedIn) enable users to connect with recruiters, industry leaders, and fellow professionals to grow their network.
- **Endorsements & Recommendations:** Job seekers can receive skill endorsements and testimonials from colleagues or former employers, boosting

credibility and visibility.

- **Industry Engagement:** Many portals allow users to participate in forums, attend virtual job fairs, and join groups based on their interests or profession.

5. Accessibility and Reach:

- **Global Opportunities:** Candidates can apply for jobs not just locally but also in different regions or countries, expanding their career options.
- **Mobile-Friendly:** Many platforms offer responsive web designs or mobile apps, allowing users to search and apply for jobs on the go.

6. Cost-Effective:

- **Free Access for Job Seekers:** Most job portals are free for job seekers, providing immense value without any investment.
- **Affordable Hiring for Employers:** Employers can post job listings and search for candidates at a fraction of the cost of traditional hiring methods.

LIMITATIONS

1. High Competition:

- **Crowded Marketplace:** Job portals attract a large number of applicants for every listed position, often resulting in hundreds of applications for a single role.
- **Low Visibility for New Candidates:** Fresh graduates or candidates without prior experience may struggle to get noticed among experienced applicants.
- **Automated Rejections:** Many systems use algorithms to filter candidates, which can result in worthy applicants being rejected automatically due to keyword mismatches or formatting errors.

2. Lack of Personalization:

- **Impersonal Communication:** Standard resume and cover letter formats limit the ability to express passion, creativity, or fit for company culture.
- **Limited Employer Insight:** Employers may find it difficult to assess soft skills, communication abilities, or work ethics solely through digital profiles and documents.
- **One-Size-Fits-All Approach:** Some platforms provide generic recommendations that may not align with the user's long-term career goals or preferences.

3. Information Overload:

- **Too Many Listings:** Job seekers may become overwhelmed by the sheer volume of job listings, making it difficult to identify the most relevant opportunities.
- **Duplicate Listings:** Sometimes the same job is posted multiple times by different recruiters, creating confusion and redundancy.

4. Risk of Scams and Fraudulent Listings:

- **Fake Employers:** Some job portals are prone to scam postings or fake

- employers seeking personal information from unsuspecting applicants.
- **Phishing Risks:** Users may receive fake emails or messages pretending to be from legitimate recruiters, leading to data breaches.

5. Limited Feedback and Communication:

- **No Response Culture:** Often, candidates never hear back from employers, even after multiple applications, which can be demotivating.
- **Delayed Updates:** Notifications regarding application status, interview calls, or rejections may not always be timely or reliable.

CHAPTER 7

10.1 Future Scope

The scope of the project includes that what all future enhancements can be done in this system to make it more feasible to us:-

Future Scope of Job Portal: Key Points

1. AI-Powered Job Matching

- Implement smart matching algorithms to connect job seekers with the most suitable job postings.
- Use AI to analyze resumes and job descriptions to increase accuracy in matching.
- Add a **career suggestion engine** based on user skills and interests.

2. Social Media and Professional Network Integration

- Integrate **LinkedIn or GitHub login** for easy profile creation and professional history imports.
- Add options for users to **share job listings on WhatsApp, Twitter, and Facebook**, boosting visibility.
- Enable recruiters to view **social profiles and project portfolios**.

3. Global Reach and Language Support

- Expand the platform to support **international job listings** with currency and time zone conversions.
- Provide **multilingual support** for broader accessibility.
- Include **visa guidance sections** for users applying to jobs abroad.

4. Enhanced Remote Job Features

- Add advanced filters for **remote, hybrid, and freelance** opportunities.
- Organize **virtual job fairs and live career webinars** within the portal.
- Build a **remote readiness toolkit** for users to prepare for remote work.

5. Smart Communication & Interview Tools

- Allow **chat functionality** between job seekers and recruiters.
- Include **automated email updates** on application status.
- Add **interview scheduling with calendar sync** and interview reminders.

6. Gamification & Motivation

- Introduce **badges or points** for job seekers based on activity (e.g., applying, completing profiles).
- Reward users for referring friends or completing learning modules.
- Display **top active users** and success stories to motivate others.

7. Learning & Upskilling Features

- Suggest relevant **certification courses or training** based on job search behavior.
- Embed quick **coding or aptitude tests** to help users prepare and prove their skills.
- Display **learning progress and badges** on user profiles.

8. Company Insights & Reviews

- Add a section for **real employee reviews** to help applicants evaluate companies.
- Display **company ratings**, benefits, and workplace culture highlights.
- Enable **company comparison tools** for better decision-making.

CHAPTER 8

CONCLUSION

The future scope of job portals presents numerous opportunities for growth and enhancement, making them even more effective and user-friendly for both job seekers and employers. By integrating advanced technologies like AI, blockchain, and mobile apps, job portals can significantly improve job matching, data security, and user engagement. Personalization through real-time alerts, skill development, and diversity initiatives will help candidates find the most relevant opportunities while fostering inclusive hiring practices.

Additionally, with the rise of remote work, international job listings, and virtual tools, job portals will be able to connect a global workforce with employers in more efficient and dynamic ways. By focusing on user-centric features such as video resumes, personalized job recommendations, and gamified experiences, job portals will create a more engaging and productive job search experience.

Artificial Intelligence can further enhance resume parsing, predictive job matching, and sentiment analysis of cover letters and interviews, enabling more accurate recruitment decisions. Integration with e-learning platforms can also allow users to upskill in real time, aligning candidate capabilities with current market demands.

Job portals may evolve into holistic career platforms offering career counseling, mentorship matching, and mental health support, making the user experience more comprehensive. Advanced analytics and dashboards can provide employers with deep insights into candidate behavior, application trends, and hiring funnels, enabling data-driven recruitment strategies.

As the job market continues to evolve, the continuous enhancement of these platforms will play a vital role in shaping the future of recruitment, making the hiring process smoother, more transparent, and more accessible to everyone. In summary, the future of job portals looks promising, with many innovative opportunities to explore that will benefit both job seekers and employers alike.

APPENDIX 1 CODING

1.1 HOME PAGE

```
<%@ Page Title="" Language="C#"
    MasterPageFile="~/Default.Master" AutoEventWireup="true"
    CodeBehind="SignIn.aspx.cs" Inherits="Job_Portal.SignIn"
%>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2"
    ContentPlaceHolderID="ContentPlaceHolder1"
    runat="server">
<table>
<tr>
<td>UserType:-</td>
<td><asp:DropDownList ID="ddlusertype" runat="server">
<asp:listitem Text="----Select ----" Value="0"></asp:listitem>
<asp:listitem Text="ADMIN" Value="1"></asp:listitem>
<asp:listitem Text="JOBSEEKERS" Value="2"></asp:listitem>
<asp:listitem Text="JOBRECRUITER" Value="3"></asp:listitem>
</asp:DropDownList>
</td>
</tr><tr>
<td>Email ID:-</td>
<td><asp:TextBox ID="txtemail" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td>Password:-</td>
<td><asp:TextBox ID="txtpassword" runat="server"></asp:TextBox></td>
</tr>
<tr>
<td></td>
<td><asp:ButtonID="signin"runat="server"Text="Submit"
OnClick="signin_Click"/></td>
</tr>
<tr>
<td></td><td><asp:LabelID="lblmsg"runat="server"ForeColor="Red"
FontBold="true"></asp:Label></td></tr></table>
```

ADMIN PAGE

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Admin.Master"
AutoEventWireup="true" CodeBehind="AdminHome.aspx.cs"
Inherits="Job_Portal.Admin.AdminHome" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
<style>
body {
font-family: Arial, sans-serif; margin: 0;
padding: 0;
background-color: #f4f4f9;
}
.header {
background-color: #4CAF50; color: white;
text-align: center; padding: 10px 0;
}
.container {
text-align: center; padding: 50px;
}
h1 {
font-size: 3rem; color: #333;
}
.footer {
position: absolute; bottom: 0;
width: 100%;
background-color: #4CAF50; color: white;
text-align: center; padding: 10px 0;
}</style> </asp:Content>
```

JOBSEEKERS PAGE

```
<% @ Page Title="" Language="C#" MasterPageFile="~/JobSeeker.Master"
AutoEventWireup="true" CodeBehind="JobSeekersHome.aspx.cs"
Inherits="Job_Portal.JobSeekers.JobSeekersHome" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
<table>
<tr>
<td>Name :</td>
<td><asp:Label ID="lblname" runat="server"></asp:Label></td>
<tr>
<td>Email :</td>
<td>
<ASP:Label ID="lblemail" runat="server"></ASP:Label></td>
<td>Password :</td>
<td>
<asp:Label ID="lblpassword" runat="server"></asp:Label></td>
</tr>
<tr>
<td>Mobile Number :</td>
<td>
<asp:Label ID="lblmobile" runat="server"></asp:Label></td>
<td>Date of Birth :</td>
<td>
<asp:Label ID="lbldateofbirth" runat="server"></asp:Label></td>
</tr>
<tr>
<td>Gender :</td>
<td>
<asp:Label ID="lblgender" runat="server"></asp:Label>
</tr>
<tr>
<td>Job Profile :</td>
<td>
<asp:Label ID="lbljobprofile" runat="server"></asp:Label></td>
```

<td>Job Experince :</td>


```

<td>
<asp:Label ID="lblexp" runat="server"></asp:Label></td>
</tr>
<tr>
<td>Qualification :</td>
<td>
<asp:Label ID="lblqualification" runat="server"></asp:Label></td>
<td>State :</td>
<td>
<asp:Label ID="lblstate" runat="server"></asp:Label></td>
</tr>
<tr>
<td>City :</td>
<td>
<asp:Label ID="lblcity" runat="server"></asp:Label></td>
<td>Address :</td>
<td>
<asp:Label ID="lbladdress" runat="server"></asp:Label></td>
</tr>
<tr>
<td>Photo Upload :</td>
<td>
<asp:Label ID="lblphoto" runat="server" /></td>
<td>Resume Upload :</td>
<td>
<asp:Label ID="lblresume" runat="server" /></td>
</tr>
<tr>
<td>Security Question :</td>
<td>
<asp:Label ID="blecurityquestion" runat="server"></asp:Label></td>
<td>Answer :</td>
<td>
<asp:Label ID="blanswer" runat="server"></asp:Label></td>
</tr>
<tr><td>InsertedDate :</td><td>
<asp:Label ID="lbldateinsert" runat="server"></asp:Label></td>
<td>Status</td>
<td><asp:Label ID="lblstatus" runat="server"></asp:Label></td>

```

JOBRECRUITER PAGE

```
<% @ Page Title="" Language="C#" MasterPageFile="~/JobRecruiter.Master"
AutoEventWireup="true"
CodeBehind="~/JobRecruiterModule/JobRecruiterHome.aspx.cs"
Inherits="Job_Portal.JobRecruiter.JobRecruiterHome" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
<table>
<tr>
<td>Name :</td>
<td>
<asp:Label ID="lblname" runat="server"></asp:Label></td>
<td>URL :</td>
<td>
<asp:Label ID="lblurl" runat="server"></asp:Label></td>
</tr>
<tr>
<td>Email :</td>
<td>
<asp:Label ID="lblemail" runat="server"></asp:Label></td>
<td>Password :</td>
<td>
<asp:Label ID="lblpassword" runat="server"></asp:Label></td>
</tr>
<tr>
<td>HR :</td>
<td>
<asp:Label ID="lblhr" runat="server"></asp:Label></td>
<td>Mobile Number :</td>
<td>
<asp:Label ID="lblmobile" runat="server"></asp:Label></td>
</tr>
<tr>
<td>State :</td>
<td>
<asp:Label ID="lblstate" runat="server"></asp:Label></td>
<td>City :</td>
<td>
<asp:Label ID="lblcity" runat="server"></asp:Label></td>
</tr>
<tr>
<td>Address :</td>
<td>
```

```
<asp:Label ID="lbladdress" runat="server"></asp:Label></td></table>
```

CSS

```
@import
url('https://fonts.googleapis.com/css?family=Open+Sans:300i,400,600,700,800');
@import url(../icons/font-awesome/css/font-awesome.min.css);
@import url(../icons/simple-line-icons/css/simple-line-icons.css); @import
url(../icons/weather-icons/css/weather-icons.min.css); @import
url(../icons/linea-icons/linea.css);
@import url(../icons/themify-icons/themify-icons.css); @import url(../icons/flag-
icon-css/flag-icon.min.css);
@import url(../icons/material-design-iconic-
font/css/materialdesignicons.min.css); @import url(spinner.css);
@import url(animate.css);
.preloader { width: 100%;
height: 100%;
top: 0; position: fixed; z-index: 99999;
background: #fff;
}
.preloader .cssload-speeding-wheel { position: absolute;
top: calc(46.5%); left: calc(46.5%);
}
* {
outline: none;
}
body { background: #fff;
font-family: 'Open Sans', sans-serif; margin: 0;
overflow-x: hidden; color: #02756a;
}
html {
position: relative; min-height: 100%; background: #ffffff; a:focus,
```

```

a:hover {
text-decoration: none;
}
a.link {
color: #455a64;
}
a.link:focus, a.link:hover { color: #1976d2;
}
.img-responsive,
.carousel.vertical .carousel-inner > .item > img,
.carousel.vertical .carousel-inner > .item > a > img { width: 100%;
height: auto;
display: inline-block;
}
.img-rounded { border-radius: 4px;
}
.mdi-set,
.mdi:before {
line-height: initial;
}
h1,
h2,
h3,
h4,
h5,
h6 {
color: #455a64; font-weight: 400;
}
h1 {
line-height: 40px; font-size: 36px;
}
h2 { font-size: 24px;
}

```

```

h3 {
line-height: 30px; font-size: 21px;
}
h4 {
line-height: 22px; font-size: 18px;
}
h5 {
line-height: 18px; font-size: 16px; font-weight: 400;
}
h6 {
line-height: 16px; font-size: 14px; font-weight: 400;
.message-list .checkbox-wrapper-mail label:active { background: #87949b;
}
.mail-list a {
font-family: "Roboto", sans-serif; vertical-align: middle;
color: #797979; padding: 10px 15px; display: block;
}
@media (max-width: 648px) {
.inbox-leftbar { width: 100%;
}.inbox-rightbar { margin-left: 0; }}
@media (max-width: 520px) {
.message-list li .col-mail-1 { width: 150px;}bottom: 110px;
right: 40px;
color: #FFF;border-radius: 50px;
text-align: center;}.my-float2 { font-size: 24px; margin-top: 18px;}

```

```

a.float2 + div.label-
container2 { visibility:
hidden;
opacity: 0;
transition: visibility 3s, opacity 0.5s
ease;} a.float2:hover + div.label-
container2 { visibility: visible;
opacity: 1;}#hideMe {
-moz-animation: cssAnimation 1s ease-in 2s forwards;
-webkit-animation: cssAnimation 1s ease-in 2s forwards;
/* Safari and Chrome */
-o-animation: cssAnimation 1s ease-in 2s forwards;
/* Opera */
animation: cssAnimation 1s ease-in 2s forwards;
-webkit-animation-fill-mode:
forwards; animation-fill-mode:
forwards;
} @keyframes
cssAnimation { from {left:
0px;}
to {left: 1900px;}} @-webkit-keyframes
cssAnimation { from {left: 0px;}
to {left: 1900px;}
}

```

DATABASE

```
create database
dbjobportal use
dbjobportal
create table admin(
AdminId int primary key
identity, AdminName
varchar(50), AdminEmail
varchar(50),
AdminPassword varchar
(50), InsertedDate date,
Status int default 0
)
insert into admin
(AdminName,AdminEmail,AdminPassword,InsertedDate)values('admin','admin','a
dmin',GETDATE());
select*from admin
create procedure
proc_admin @adminid int,
@adminNewPassword
varchar(50)=null,
@adminCurrentPassword
varchar(50)=null as
begin
update admin set AdminPassword=@adminNewPassword
where AdminId=@adminid and
AdminPassword=@adminCurrentPassword end
Create table
tblquestions (
QuestionId int primary key
identity, QuestionName
varchar(50)
)
insert into tblquestions(QuestionName) values('what is childhood name ?')
insert into tblquestions(QuestionName) values('what is your favorite food
name ?') insert into tblquestions(QuestionName) values('what is your first
teacher name ?') create table tblJobProfile
(
```



```

JobProfileId int primary key
identity, JobProfileName
varchar(50)
insert into tblJobProfile(JobProfileName) values('.Net Developer')
insert into tblJobProfile(JobProfileName) values('Java
Developer') insert into tblJobProfile(JobProfileName)
values('Php Developer') insert into
tblJobProfile(JobProfileName) values('Testing')
insert into tblJobProfile(JobProfileName) values('HR
') insert into tblJobProfile(JobProfileName)
values('Sales') create table tblQualifications
(
QualificationId int primary key
identity, QualificationName
varchar(50)
)
insert into tblQualifications(QualificationName)
values('M.Tech') insert into
tblQualifications(QualificationName) values('B.Tech') insert
into tblQualifications(QualificationName) values('MCA')
insert into tblQualifications(QualificationName)
values('BCA') insert into tblQualifications(QualificationName)
values('Diploma') insert into
tblQualifications(QualificationName) values('Polytecnic')
create table tblState
(
StateId int primary key
identity, StateName
varchar(50)
)
insert into tblState(StateName) values('Delhi')
insert into tblState(StateName) values('Uttar
Pradesh') insert into tblState(StateName)
values('Bihar')
insert into tblState(StateName)
values('Punjab') insert into
tblState(StateName) values('Gujrat') insert
into tblState(StateName) values('Rajasthan')
create table tblCity

```

```

(
CityId int primary key
identity, StateId int,
CityName varchar(50)
)
insert into tblCity(StateId,CityName) values(1,'Laxmi nagar') insert into
tblCity(StateId,CityName)
values(1,'janakpuri') insert into
tblCity(StateId,CityName) values(2,'Lucknow') insert
into tblCity(StateId,CityName) values(2,'Kanpur')
insert into tblCity(StateId,CityName)
values(2,'Varanasi') insert into
tblCity(StateId,CityName) values(3,'Patna')
insert into tblCity(StateId,CityName) values(3,'Motihari')
insert into tblCity(StateId,CityName)
values(4,'Amritsar') insert into
tblCity(StateId,CityName) values(4,'Ludhiana') insert
into tblCity(StateId,CityName)
values(5,'Ahemadabad') insert into
tblCity(StateId,CityName) values(5,'Surat')
insert into tblCity(StateId,CityName)
values(6,'Jaipur') insert into
tblCity(StateId,CityName) values(6,'Jaisalmer')
select*from tblJobSeeker
create table
tblJobSeeker (
JobSeekerId int primary key
identity, JobSeekerName
varchar(50), JobSeekerEmail
varchar(50),
JobSeekerPassword
varchar(50),
JobSeekerGender int,
JobSeekerJobProfile int,
JobSeekerQualification int,
JobSeekerState int,
JobSeekerCity int,
JobSeekerAddress
varchar(50),

```

```

JobSeekerQuestion int,
JobSeekerAnswer
varchar(50),
JobSeekerDOB date,
JobSeekerMobile bigint,
JobSeekerExp int,
JobSeekerImage
varchar(100),
JobSeekerResume
varchar(100),
JobSeekerComments varchar(2000),
JobSeekerInsertedDate date,
JobSeekerStatus int default 0
)
alter procedure
Proc_JobSeeker @action
varchar(50)=null,
@JobSeekerId int=0,
@JobSeekerName
varchar(50)=null,
@JobSeekerEmail
varchar(50)=null,
@JobSeekerPassword
varchar(50)=null,
@JobSeekerNewPassword
varchar(50)=null, @JobSeekerGender
int=0, @JobSeekerJobProfile int=0,
@JobSeekerQualification int=0,
@JobSeekerState int=0,
@JobSeekerCity int=0,
@JobSeekerAddress
varchar(50)=null,
@JobSeekerQuestion int=0,
@JobSeekerAnswer
varchar(50)=null,
@JobSeekerDOB date=null,
@JobSeekerMobile bigint=0,
@JobSeekerExp int=0,
@JobSeekerImage

```

```

    varchar(100)=null,
    @JobSeekerResume
    varchar(100)=null,
    @JobSeekerComments
    varchar(2000)=null AS
BEGIN
create table tblJobRecruiter
(JobRecruiterId int primary key
identity, JobRecruiterName
varchar(50), JobRecruiterEmail
varchar(50),
JobRecruiterPassword
varchar(50), JobRecruiterURL
varchar(50), JobRecruiterState int,
JobRecruiterCity int,
JobRecruiterAddress
varchar(50), JobRecruiterHR
varchar(50),
JobRecruiterMobile bigint,
JobRecruiterImage
varchar(100),
JobRecruiterComments
varchar(2000),
JobRecruiterInsertedDate date,
select * from
tblJobPost
truncate table
tblJobPost

```

REFERNCES

1. <https://www.linkedin.co>
2. <https://www.naukri.com/>