

INTRODUCTION ABOUT THE PROJECT

In today's technology-driven world, the job market has become highly competitive, with both job seekers and employers seeking efficient ways to connect. The rise of digital platforms has transformed recruitment, providing job seekers access to numerous opportunities while enabling employers to reach a larger, more diverse talent pool. However, despite the abundance of online job portals, many platforms struggle to offer the personalized, user-friendly experience that modern users demand. This is where **Career Horizon - Job Portal** comes in.

Career Horizon bridges the gap between talent and opportunity by offering a comprehensive, feature-rich platform for both job seekers and employers. Designed with a focus on usability, scalability, and performance, the platform provides a streamlined approach to job searching and recruitment. **Job seekers** can explore various job listings across industries such as IT, healthcare, education, marketing, and more, with personalized job recommendations, advanced search filters, and the ability to save jobs for later. **Employers** can post job openings, filter applications, and manage their recruitment process with ease.

The goal of **Career Horizon** is to create a one-stop solution that helps individuals achieve their career goals while assisting organizations in finding top talent efficiently. By leveraging modern technologies and best practices, the platform not only provides a seamless job search experience but also ensures access to high-quality opportunities that align with users' skills and aspirations.

As the job market becomes increasingly digital and remote work continues to rise, **Career Horizon** is built to meet these changing needs. Its responsive design, advanced filtering capabilities, and personalized features make it a powerful tool for users on both sides of the recruitment process.

From a technical perspective, the platform's development is rooted in modern web development principles, utilizing **HTML5**, **CSS3**, **JavaScript**, **PHP**, and **MySQL** to

create a scalable and secure environment. The project follows an **Agile development** approach, ensuring features are implemented iteratively with continuous feedback. The back-end architecture is built using the **Model-View-Controller (MVC)** design pattern, ensuring maintainability and ease of future enhancements.

By combining ease of use, performance, and a personalized user experience, **Career Horizon** seeks to stand out in the job portal market and become a trusted resource for both job seekers and employers. Whether an individual is looking for their first job, a career change, or a leadership role, **Career Horizon** offers the tools and resources needed to succeed in today's competitive job market.

OBJECTIVE AND SCOPE OF PROJECT

The Career Horizon Job Portal has the following objectives:

- To provide a wide range of job listings across multiple industries, ensuring a diverse selection for job seekers.
- To offer personalized job recommendations based on user profiles, preferences, and search history.
- To enable job seekers to save jobs they are interested in, creating a personalized "like" section for easy access.
- To allow employers to post job openings, manage applications, and connect with potential candidates seamlessly.
- To provide a secure and user-friendly platform for job seekers to create profiles, upload resumes, and apply for jobs.
- To enhance the efficiency of the recruitment process for employers by offering advanced filtering and search options.

The scope for developing the Career Horizon Job Portal includes:

- Managing user accounts, with personalized sections for job seekers to view their saved jobs and application history.
- Offering advanced search functionalities that allow users to filter jobs based on location, industry, salary, and job type.
- Providing a responsive design that ensures accessibility across various devices, including desktops, tablets, and smartphones.
- Integrating secure payment gateways for premium services offered to employers for advanced job postings and candidate search features.

- Implementing analytics tools for employers to track job posting performance and applicant trends.

INTRODUCTION OF HTML AND CSS

HTML (Hypertext Mark-up Language) and CSS (Cascading Style Sheets) are essential technologies that form the foundation of web development, enabling the creation of visually engaging and well-structured web pages. Much like Visual Basic in application development, HTML and CSS are integral in defining the structure and presentation of web content.

HTML (Hypertext Mark-up Language):

HTML serves as the fundamental framework of web pages, providing a structured mark-up language to define content and layout. Developers use HTML to create documents by utilizing tags that categorize and organize different elements on a webpage, such as headings, paragraphs, images, links, forms, and more.

CSS (Cascading Style Sheets):

CSS complements HTML by allowing developers to manage the visual styling of web pages. It introduces style rules that dictate how HTML elements should be displayed across various devices. By separating content from design, CSS streamlines the creation of consistent and visually appealing user interfaces.

Together, HTML and CSS provide the essential building blocks for crafting an engaging and user-friendly web experience. While HTML defines the structure, CSS enhances the design, offering a flexible and responsive layout. This synergy supports the principles of Rapid Application Development (RAD), simplifying the process of designing and styling web interfaces efficiently.

INTRODUCTION OF JAVASCRIPT

JavaScript is a versatile and powerful programming language primarily employed in web development. It is a key component of modern web browsers, empowering developers to create dynamic and interactive user interfaces. Often abbreviated as JS, JavaScript plays a pivotal role in enhancing web page functionality through client-side scripting.

Key Features of JavaScript:

Client-Side Scripting:

JavaScript is predominantly used for client-side scripting, meaning it executes directly within the user's web browser. This capability enables developers to craft dynamic content, validate forms, handle events, and manipulate the Document Object Model (DOM), allowing webpage content to be updated dynamically without the need for a page reload.

Object-Oriented:

JavaScript is an object-oriented language, supporting core concepts such as encapsulation, inheritance, and polymorphism. These features allow developers to organize and modularize their code effectively, facilitating better code management and scalability.

Interactivity and Dynamic Content:

JavaScript enhances web pages by enabling real-time responsiveness to user interactions. Developers can implement features like sliders, pop-ups, and form validations, while also facilitating dynamic content loading and updating, contributing to more engaging and user-friendly websites.

Event-Driven Programming:

JavaScript follows an event-driven programming model, wherein developers define functions that execute in response to specific events such as button clicks, mouseovers, or form submissions. This paradigm improves user experience by making web pages more interactive and responsive to user actions.

Cross-Browser Compatibility:

JavaScript is supported by all major web browsers, including Chrome, Firefox, Safari, and Edge, ensuring consistent functionality across diverse platforms and devices.

JavaScript is an indispensable technology in modern web development, frequently working alongside HTML and CSS to build interactive web applications. Over time, JavaScript has expanded beyond its initial use in web browsers, now also playing a significant role in server-side development (via Node.js), mobile app development, and other application domains.

INTRODUCTION OF PHP

PHP is a general-purpose scripting language primarily designed for web development. Initially created by Danish-Canadian programmer Rasmus Lerdorf in 1993 and officially released in 1995, PHP's reference implementation is now maintained by The PHP Group. Originally an acronym for "Personal Home Page," PHP now stands for the recursive term "PHP: Hypertext Preprocessor."

PHP code is typically processed on a web server through a PHP interpreter, which can be implemented as a module, daemon, or Common Gateway Interface (CGI) executable. The result of the interpreted PHP code—whether HTML, binary image data, or other types of output—forms part of the HTTP response. Various web template systems, content management systems, and frameworks are available to streamline and facilitate this process.

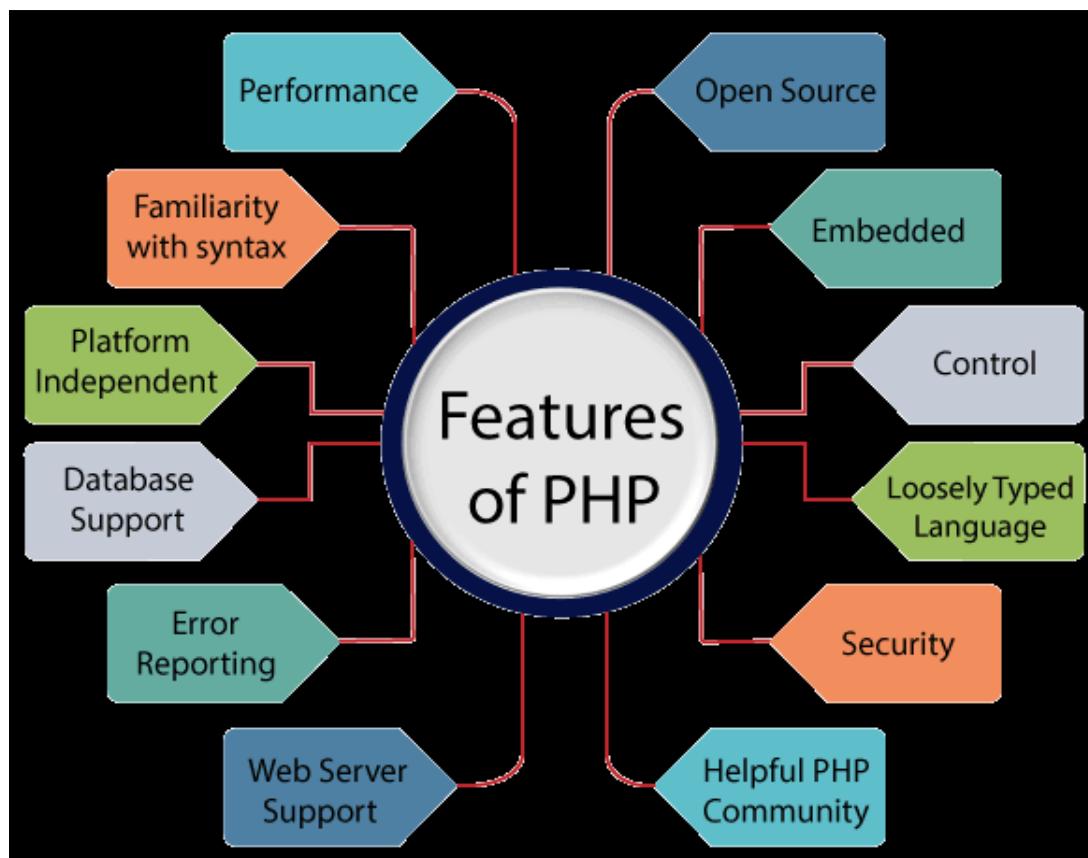
While PHP is most commonly associated with web development, it is also capable of handling non-web-based programming tasks such as creating standalone graphical applications and controlling robotic drones. PHP code can also be executed directly from the command line, expanding its versatility.

The standard PHP interpreter, powered by the Zend Engine, is open-source software released under the PHP License. PHP is widely portable and compatible with most web servers across various operating systems and platforms. Although PHP lacked a formal specification until 2014, its original implementation served as the de facto standard for other implementations to follow. Since 2014, efforts have been undertaken to create a formal specification for PHP.

According to W3Techs, as of January 2023, PHP powers 77.8% of websites with known server-side programming languages. However, it is concerning that a significant portion of PHP users continue to utilize unsupported versions such as PHP 7 and even PHP 5, both of which no longer receive security updates and are known to have critical vulnerabilities.

PHP remains a powerful and reliable language for server-side scripting, with applications ranging from form data handling and dynamic content generation to database management, session handling, cookie management, and email processing. Furthermore, PHP offers numerous hash functions for encrypting user data, ensuring its security and robustness as a server-side scripting language. These capabilities make PHP an indispensable tool for modern web development.

Features of PHP:



Performance:

PHP executes scripts faster than many other programming languages, such as JSP and ASP. This is due to PHP's ability to use its own memory management, which reduces the server's workload and loading times, ultimately improving processing speed and overall performance.

Open Source:

PHP is open-source software, meaning its source code is freely available. Developers can modify and develop PHP versions to suit their specific needs without any associated costs. All components of PHP are open for free download and use.

Familiarity with Syntax:

PHP has a straightforward and easily understandable syntax, making it accessible for programmers of varying skill levels. The simplicity of the syntax allows for efficient coding and quick adoption by developers.

Embedded:

PHP code can be seamlessly embedded within HTML tags and other scripts, enabling developers to mix PHP with standard HTML code effortlessly, enhancing the flexibility of web development.

Platform Independence:

PHP is platform-independent, meaning it can be deployed across multiple operating systems such as Windows, macOS, Linux, and UNIX. A PHP application developed on one operating system can be executed on another without modification.

Database Support:

PHP supports a wide range of databases, including MySQL, SQLite, ODBC, and others. This makes it a versatile language for applications requiring database interaction.

Error Reporting:

PHP provides predefined error reporting constants, such as E_ERROR, E_WARNING, E_STRICT, and E_PARSE, that allow developers to generate error notices or warnings during runtime. This feature is essential for debugging and improving code reliability.

Loosely Typed Language:

PHP is a loosely typed language, meaning that variables do not require explicit type declarations. The data type is automatically determined at runtime based on the value assigned to the variable, which simplifies development and reduces errors.

Web Server Support:

PHP is compatible with a wide array of web servers, including Apache, Netscape, and Microsoft IIS, allowing developers to choose their preferred server environment without compatibility concerns.

Security:

PHP provides robust security features designed to protect websites from threats and malicious attacks. Multiple layers of security are integrated into PHP, making it a reliable choice for developing secure websites.

Control:

PHP is known for its efficiency and control, as it allows developers to accomplish complex tasks with relatively few lines of code. Additionally, PHP offers extensive control over

website functionality, enabling developers to make quick adjustments and improvements when needed.

A Helpful PHP Community:

PHP benefits from a large and active community of developers who regularly contribute to documentation, tutorials, FAQs, and other helpful resources. The wealth of community-driven knowledge makes learning PHP easier and more accessible for both beginners and experienced developers.

INTRODUCTION OF MYSQL

What is a Database?

A database is a specialized application designed to store, manage, and organize data in a structured manner. It provides distinct APIs that allow users to create, access, manage, search, and replicate the data it contains. While other types of data storage systems, such as file systems or large hash tables in memory, can be used, they generally do not offer the same speed and ease of data retrieval and manipulation as a dedicated database system.

In modern applications, relational database management systems (RDBMS) are commonly used to handle large volumes of data. The term "relational" refers to the way data is organized into tables, where relationships between the data are established using primary keys and foreign keys.

Relational Database Management System (RDBMS)

An RDBMS is software that facilitates the implementation and management of relational databases. It enables the creation of databases with tables, columns, and indexes while ensuring the following functions:

- **Table Structure:** Allows the creation and organization of tables to store data.
- **Referential Integrity:** Maintains the integrity of relationships between tables using primary and foreign keys.
- **Index Management:** Automatically updates indexes to optimize data retrieval.
- **SQL Query Processing:** Interprets SQL queries and retrieves data by combining information from various tables.

RDBMS Terminology

Before exploring specific database systems like MySQL, it is important to understand some key terminology used in RDBMS:

- **Database:** A collection of related tables that store data.
- **Table:** A table is a collection of rows and columns, representing data in a structured format similar to a spreadsheet.
- **Column:** A column contains data of a specific type or category, such as customer names, addresses, or product prices.
- **Row:** A row (also known as a tuple, entry, or record) is a set of related data within a table, often representing an individual record.
- **Redundancy:** The practice of storing data multiple times to improve system performance. While redundancy can speed up access, it may introduce challenges in data consistency.
- **Primary Key:** A primary key is a unique identifier for each record in a table. It ensures that each row in a table is distinguishable from others, and no two rows can have the same primary key value.
- **Foreign Key:** A foreign key is used to create a relationship between two tables. It refers to the primary key of another table, ensuring referential integrity between the two tables.
- **Compound Key (Composite Key):** A compound key is a primary key that consists of two or more columns, used when a single column cannot uniquely identify records in a table.
- **Index:** An index in a database is a data structure that allows for faster retrieval of data, similar to an index at the back of a book. It speeds up query execution by reducing the amount of data that needs to be scanned.

- **Referential Integrity:** Referential integrity ensures that foreign key values in a table always point to valid, existing rows in another table, preventing orphaned records and maintaining data consistency.

What is SQL?

SQL (Structured Query Language) is the standard language used for managing and manipulating data in a relational database management system (RDBMS). It provides an interface for accessing, updating, deleting, and managing database structures and data. SQL is the foundation of interacting with databases, enabling users to perform a wide range of operations on the data they store.

SQL is divided into four main subsets:

1. **DDL (Data Definition Language):** DDL is used to define and manage the structure of a database. It allows the creation, alteration, and deletion of database objects such as tables, views, and schemas. Key commands include:
 - **CREATE:** To create new database objects.
 - **ALTER:** To modify existing database objects.
 - **DROP:** To delete database objects.
2. **DML (Data Manipulation Language):** DML deals with the manipulation of data within the database. It allows for adding, updating, deleting, and querying data. Common DML commands include:
 - **SELECT:** To retrieve data from one or more tables.
 - **INSERT:** To add new data into a table.
 - **UPDATE:** To modify existing data.
 - **DELETE:** To remove data from a table.

3. **DCL (Data Control Language):** DCL is used to control access to data within the database. It helps manage user permissions, allowing for secure access and data management. Main DCL commands include:
 - **GRANT:** To give a user specific access rights to database objects.
 - **REVOKE:** To remove previously granted permissions.
4. **TCL (Transaction Control Language):** TCL is used to manage transactions in a database. A transaction is a sequence of operations performed as a single unit. TCL ensures that transactions are handled properly, preserving data integrity. Key TCL commands include:
 - **COMMIT:** To save all changes made during the transaction.
 - **ROLLBACK:** To undo changes made during the transaction if an error occurs.
 - **SAVEPOINT:** To set a point within a transaction to which you can roll back.
 - **SET TRANSACTION:** To configure transaction properties.

MySQL

MySQL is a widely used relational database management system (RDBMS) developed and maintained by MySQL AB, a Swedish company. It is known for being fast, reliable, and easy to use, making it a popular choice for both small businesses and large enterprises. MySQL has several advantages that contribute to its growing popularity:

- **Open Source:** MySQL is released under an open-source license, meaning it is free to use and modify.
- **Powerful:** Despite being free, MySQL provides a range of features that rival some of the most expensive commercial database management systems. It can handle substantial databases and a variety of data processing tasks.

- **Standardized SQL Support:** MySQL uses a standard form of SQL, making it compatible with most database applications.
- **Cross-Platform Compatibility:** MySQL can run on various operating systems, including Windows, Linux, and macOS, and supports multiple programming languages like PHP, PERL, C, C++, and Java.
- **Speed:** MySQL is known for its quick performance, especially with large datasets. It is optimized for high-speed operations.
- **Scalability:** MySQL can handle large databases with ease. It supports up to 50 million rows per table, with the theoretical file size limit reaching 8 million terabytes.
- **PHP Integration:** MySQL is often used alongside PHP, making it a popular choice for web development.
- **Customization:** Being open source, MySQL can be customized and extended to fit specific needs. Developers can modify the software to suit their particular environment or requirements.

DEFINITION OF PROBLEM FOR CAREER HORIZON

The **Career Horizon** job portal faces several challenges related to its functionality, user experience, and backend operations. These problems span across multiple areas such as search efficiency, system performance, security, and user support, all of which impact the platform's overall effectiveness and satisfaction for both job seekers and employers. Below are the primary problems:

1. User Experience and Interface:

- Ensuring intuitive navigation for job seekers and employers.
- Addressing cluttered or complex design elements that could hinder users from efficiently utilizing the portal.

2. Job Search and Filter Functionality:

- Ensuring accurate, relevant, and fast search results based on user preferences.
- Optimizing filtering mechanisms to allow job seekers to easily refine results and find the best opportunities.

3. Job Listing Accuracy and Maintenance:

- Keeping job listings updated and free from errors to avoid confusion for job seekers.
- Implementing a streamlined process for employers to manage listings effectively and in real-time.

4. Database Management and Performance:

- Ensuring that the database can handle a large volume of job listings and user data without compromising performance.
- Optimizing queries to avoid slowdowns and maintain quick response times under high traffic conditions.

5. Security and Privacy:

- Protecting sensitive data like resumes, personal details, and application history.
- Ensuring compliance with data protection regulations (e.g., GDPR) to safeguard user privacy.

6. Payment and Billing for Premium Features:

- Addressing billing discrepancies or payment failures in premium feature access.
- Improving the process of managing payments for job listings, premium profiles, or extra visibility features.

7. Employer-Job Seeker Communication:

- Streamlining the messaging system to ensure efficient, secure, and private communication between employers and candidates.
- Addressing any issues with delayed or lost communication that could negatively impact hiring processes.

8. Mobile Compatibility and App Development:

- Ensuring a seamless mobile experience for users accessing the portal via smartphones or tablets.
- Developing a dedicated app that offers the full functionality of the website while maintaining performance.

9. System Downtime and Technical Issues:

- Minimizing downtime or service interruptions that may affect user access or data integrity.
- Optimizing backend architecture to handle high concurrent traffic during peak periods.

10. Customer Support:

- Providing fast, reliable, and efficient support for users facing issues with account management, job postings, or applications.
- Reducing response times and enhancing service standards to resolve issues promptly.

In sum, **Career Horizon** needs to address the above problems in a systematic and comprehensive manner to provide a smooth, secure, and effective job search and recruitment platform for users. These challenges require a balanced approach to enhance user experience, streamline operations, and ensure the scalability of the platform for future growth.

SYSTEM ANALYSIS

System analysis for the Career Horizon Job Portal involves understanding the requirements of both job seekers and employers. The system must provide a seamless user experience while ensuring data security and efficient management of job postings and applications. The system analysis phase includes gathering requirements through user interviews, analyzing similar platforms, and identifying key functionalities that will make Career Horizon stand out in the market. Here is the **System Analysis for Career Horizon - Job Portal** in points:

1. Problem Identification:

- Modern job portals often lack personalization and fail to provide an intuitive user experience for both job seekers and employers.
- Employers face difficulties managing applications efficiently, while job seekers struggle to find relevant job matches quickly.

2. User Requirements:

- **Job Seekers:** Need a platform that provides personalized job recommendations, the ability to save job listings, easy application management, and a responsive interface.
- **Employers:** Require features for posting job vacancies, filtering candidate applications, managing recruitment, and accessing a pool of potential candidates.

3. System Requirements:

- **Functional:**
 - User authentication and profile management.

- Job posting and search functionalities with advanced filtering options.
 - Like/Save feature for job seekers to bookmark jobs.
 - Application tracking for both employers and job seekers.
- **Non-Functional:**
 - Scalability to handle large volumes of users and data.
 - High security to protect user data, especially sensitive information like personal profiles and job applications.
 - Performance optimization to ensure fast search results and smooth navigation.

4. System Architecture:

- The system is based on a **Model-View-Controller (MVC) architecture**, separating the business logic from the user interface for better maintainability and scalability.
- The front-end is developed using **HTML5, CSS3, and JavaScript**, providing a responsive and user-friendly experience across devices.
- The back-end is developed using **PHP** and **MySQL**, enabling secure data storage, retrieval, and management.

5. Data Flow:

- The system manages user input (like job searches or job applications) through secure forms and interfaces, processes the data via PHP, and retrieves or stores the required information in the MySQL database.
- The **MVC pattern** ensures that data handling is done independently from the user interface, enhancing system modularity and scalability.

6. User Interface and Experience:

- The system emphasizes ease of use, with intuitive navigation, responsive design, and minimal steps for both job seekers and employers to complete their tasks.
- **AJAX** is used to update data asynchronously, providing a smoother user experience without the need for full page reloads.

7. Security Considerations:

- The system is designed to protect user data with **encrypted passwords** and **prepared SQL statements** to avoid SQL injection attacks.
- **SSL** is recommended for secure data transmission between users and the server.

SYSTEM REQUIREMENTS

Minimum Hardware Requirements for Development

- **CPU:** Intel Core i3 or equivalent (modern processors provide better performance).
- **Memory (RAM):** 4 GB (2 GB is minimal and may not be sufficient for development tasks).
- **Hard Disk:** 512 GB (256 GB is minimal; additional storage is recommended for development flexibility).
- **System Type:** 32-bit or 64-bit Operating System (64-bit is preferred for handling more memory and compatibility with modern tools).

Minimum Software Requirements for Development

- **Operating System:**
 - **Windows:** Windows 10 or 11.
 - **Linux:** A recent version (e.g., Ubuntu 20.04 LTS or later).
- **Database:** MySQL 5.7 or later.
- **Server:** Apache (using XAMPP or a similar package for local development).
- **Scripting Language:** PHP 7.4 or later.
- **Frontend Technologies:** HTML5, CSS3, JavaScript (including jQuery and AJAX).
- **IDE:** Visual Studio Code or PHPStorm.
- **Screen Resolution:** 1920x1080 or higher (for better visibility and ease of development; 1024x768 is minimal).

Minimum Hardware and Software Requirements for Running the Website

On PCs (Desktops and Laptops)

- **CPU:** Intel Core i3 or equivalent (sufficient for running typical web applications).
- **Memory (RAM):** 4 GB or higher (more memory will improve performance, especially with higher traffic).
- **Hard Disk:** 256 GB or higher (adequate for the website and data storage).
- **System Type:** 32-bit or 64-bit Operating System (64-bit is preferred for modern software compatibility and better performance).
- **Operating System:**
 - **Windows:** Windows 7 or later (Windows 10 or 11 preferred).
 - **Linux:** Ubuntu 16.04 LTS or later (e.g., Ubuntu 20.04 LTS or later).
- **Web Browser:** Modern browsers supporting HTML5, CSS3, and JavaScript (e.g., Google Chrome, Mozilla Firefox, Safari).

On Mobile Devices (Phones and Tablets)

- **CPU:** Quad-core processor (e.g., Snapdragon 425 or equivalent) for smoother performance.
- **Memory (RAM):** 2 GB or higher (1 GB might be too limited for modern web applications).
- **Storage:** 16 GB or higher (for storing app data and caching content).
- **Operating System:**
 - **Android:** Version 8.0 (Oreo) or later.
 - **iOS:** Version 13 or later.

- **Browser:**
 - **Android:** Google Chrome or Mozilla Firefox (latest versions).
 - **iOS:** Safari (latest version) or Google Chrome (latest version).
- **Screen Resolution:** 720x1280 pixels or higher (HD resolution for adequate display of content).
- **Internet Connection:** Reliable 4G or higher network speed (to ensure smooth access and interaction with the website).

SYSTEM DESIGN AND CODING

INTRODUCTION:

Software design is a critical phase in the software engineering process, laying the foundation for all subsequent development activities. It transcends development paradigms and application areas, serving as a common thread in the creation of every engineered system or product. Design is initiated after system requirements have been specified and analyzed. This step forms the core of the development phase and precedes coding and testing. The primary goal of software design is to create a model or representation of the system that will eventually be implemented.

The central focus of design is quality. Through the design process, the quality of the final software product is ensured by translating customer requirements into a structured framework that can be translated into code. A robust design minimizes the risk of building unstable systems, which may collapse under small changes, become difficult to test, or fail to meet quality standards until the very end.

In the design phase, detailed representations of data structures, program structures, and procedural steps are refined, reviewed, and documented. From both technical and project management perspectives, system design plays a pivotal role in guiding the software engineering process toward its successful completion.

INPUT DESIGN:

Input design is the process of converting user-oriented input into a computer-readable format. As a key component of the overall system design, input design must be executed with meticulous attention to detail. This is because the collection of input data often represents one of the most resource-intensive aspects of the system development process. Effective input design focuses on optimizing the methods of input collection while ensuring the highest possible level of accuracy and usability.

The primary objectives of input design are as follows:

1. **Cost-Effectiveness:** Create an input process that is both efficient and economical.
2. **Accuracy:** Achieve the highest possible accuracy in input data to minimize errors and inconsistencies.
3. **User-Friendliness:** Ensure that the input process is intuitive, understandable, and straightforward for the users, facilitating seamless interaction with the system.

Input Data Considerations:

When designing input data, the goal is to create a process that is easy, logical, and error-free. The data entry operators must be clear on the allocated space for each field, the correct sequence of fields, and any required formatting. A well-designed input form provides these instructions explicitly to avoid ambiguity.

The input process typically involves the following stages:

- **Data Recording:** Capturing the raw data from users or external systems.
- **Data Transcription:** Converting handwritten or verbal data into digital format.
- **Data Conversion:** Converting data into a format that the system can process.
- **Data Verification:** Checking the accuracy of input data against predefined criteria.
- **Data Control:** Ensuring the integrity and security of the data throughout the process.
- **Data Transmission:** Transmitting data to the system for processing.
- **Data Correction:** Correcting errors in the input data after initial capture.

A fundamental aspect of input design is to select the data capture methods and devices that reduce the number of stages, thereby decreasing the chances of errors and reducing the overall cost. Input types can be classified as:

- **External Input:** Data sourced from external entities such as users or other systems.
- **Internal Input:** Data generated internally by the system itself.
- **Operational Input:** Data required for system operations and process execution.
- **Computerized Input:** Input that is processed by the system automatically without direct human intervention.
- **Interactive Input:** Input provided by users in real-time through direct interaction with the system.

OUTPUT DESIGN:

Output design refers to the process of determining how the results of processing will be communicated to the users. Outputs play a crucial role in conveying meaningful information to end-users, whether it's for immediate decision-making or for creating permanent records for future reference. Designing effective output involves organizing the presentation of information in a manner that is clear, actionable, and accessible.

The outputs of a computer system are typically categorized as follows:

1. **External Outputs:** Information that is sent outside the system to users, other systems, or external entities.
2. **Internal Outputs:** Information generated within the system for internal use by the software or organization.
3. **Operational Outputs:** Outputs produced as part of routine system operations and used for system monitoring or control.

4. **Interactive Outputs:** Outputs that respond dynamically to user queries or inputs in real-time.
5. **Turnaround Outputs:** Outputs that are used by the system for processing and then returned for further action, such as feedback forms or report responses.

The key to designing effective outputs is ensuring that the right information is presented to the appropriate users in the right format, at the right time. Outputs should be structured to provide clear and concise communication that facilitates decision-making, troubleshooting, and system operation.

The design of screens and interfaces plays a critical role in making the outputs not only informative but also interactive. Users should be able to navigate through the system, request data, and fulfill their needs by querying the system effectively. This process is vital for ensuring that users can access the data they require in an intuitive and seamless manner.

CONCLUSION:

In the software engineering process, system design, including both input and output design, is foundational for building quality, reliable, and user-friendly systems. Input design ensures that data is captured accurately and efficiently, while output design guarantees that users receive the information they need in a well-structured and understandable manner. These steps are critical for the success of any software system, as they directly impact both the functionality and usability of the final product. By focusing on these design principles, developers can ensure that the system meets user needs, operates efficiently, and remains scalable and adaptable for future enhancements.

DATABASE DESIGN

TABLE 1: Users

	<input type="button" value="T"/>	<input type="button" value="userid"/>	<input type="button" value="name"/>	<input type="button" value="email"/>	<input type="button" value="phone"/>	<input type="button" value="password"/>	<input type="button" value="created_at"/>
	<input type="checkbox"/> <input type="button" value="Edit"/>	<input type="checkbox"/> <input type="button" value="Copy"/>	<input type="checkbox"/> <input type="button" value="Delete"/>	6 Prats	temp@gmail.com	1234567891	\$2y\$10\$/5A1C1b/SxgV2kzzgS5l.XRChhYwkphQgiPJHxzsm... 2025-01-09 12:02:16
	<input type="checkbox"/> <input type="button" value="Edit"/>	<input type="checkbox"/> <input type="button" value="Copy"/>	<input type="checkbox"/> <input type="button" value="Delete"/>	7 Admin	Admin@gmail.com	\$2y\$10\$0lkWME5lwPbREq27V2Z.ilpEkm71SuF.3Kx4H9yLe... 2025-01-09 15:07:39	
	<input type="checkbox"/> <input type="button" value="Edit"/>	<input type="checkbox"/> <input type="button" value="Copy"/>	<input type="checkbox"/> <input type="button" value="Delete"/>	9 Pratyush	pratyush9392@gmail.com	9302672392	\$2y\$10\$oFlqeJts15l2oj8uj9ska.3aKmbFifb5dreYZ.BMOPi... 2025-01-09 15:49:52

TABLE 2:

<input type="checkbox"/>	<input type="button" value="company_id"/>	<input type="button" value="company_name"/>	<input type="button" value="logo_url"/>	<input type="button" value="company_profile"/>	<input type="button" value="featured"/>	<input type="button" value="job_no"/>
ete	2	Google Inc.	images/google.png	Google Inc. is a multinational technology company ...	1	5
ete	3	Microsoft Corporation	images/MScorp.jpeg	Microsoft Corporation is a multinational technolog...	1	3
ete	4	Tech Innovations	images/Tech_innovations.jpeg	Tech Innovations refer to groundbreaking advanceme...	4	4
ete	5	SecureNet Technologies	images/securenetvad_cover.jpeg	SecureNet Technologies specializes in cutting-edge...	2	4
ete	6	InnoData Analytics	images/inno.png	InnoData Analytics is a data-driven company specia...	1	4
ete	7	SkyCloud Technologies	images/Sky Cloud Technologies.jpg	SkyCloud Technologies is a leading provider of clo...	3	2
ete	8	DesignSprint Innovations	images/design.png	DesignSprint Innovations is a UX/UI design and pro...	5	8

TABLE 3:

		jobid	title	keywords	logo_url	location	salary	shift	requirement	qualification	skills	company_id	posted_date
<input type="checkbox"/>	  	4	Software Engineer	Cloud Computing, Software Architecture, API Devolo...	images/google.png	Bengaluru, Karnataka	12,00,000 - 18,00,000	Full-Time	Education Graduate Age: 24+ Language: Hindi, En.	NULL	Java, Python, SQL	2	2024-08-06
<input type="checkbox"/>	  	5	Data Scientist	Data Visualization, Big Data, Statistics	images/MScorp.jpeg	Hyderabad, Telangana	14,00,000 - 20,00,000	Full-Time	Education Graduate Age: 24+ Language: Hindi, En.	Bachelor's Preferred Machine Learning, Python, R	Machine Learning, Python, R	3	2024-05-09
<input type="checkbox"/>	  	6	Software Engineer	Cloud Computing, Software Architecture, API Devolo...	images/google.png	Bengaluru, Karnataka	₹12,00,000 - ₹18,00,000	Full-Time	Education Graduate Age: 24+ Language: Hindi, En.	NULL	Java, Python, SQL	2	2024-08-06
<input type="checkbox"/>	  	7	Data Scientist	Data Visualization, Big Data, Statistics	images/MScorp.jpeg	Hyderabad, Telangana	₹14,00,000 - ₹20,00,000	Full-Time	Education Graduate Age: 24+ Language: Hindi, En.	Bachelor's Preferred Machine Learning, 1 year (R...)	Machine Learning, Python, R	3	2024-05-09
<input type="checkbox"/>	  	8	Software Engineer	Software Development, Java Developer, Backend Engi...	images/Tech_innovations.jpeg	Bangalore, Karnataka	₹7,00,000 - ₹15,00,000	Day shift	Experience with software development lifecycle (SD...	Bachelor's degree in Computer Science, IT, or rela...	Java, Python, C++, Full-stack development, SQL, No...	4	2025-01-07
<input type="checkbox"/>	  	10	Cybersecurity Analyst	Cybersecurity, Ethical Hacking, Information Securi...	images/securenetvad_cover.jpeg	Hyderabad, Telangana	₹6,00,000 - ₹12,00,000	Rotational shift	Strong analytical skills, understanding of OWASP s...	Bachelor's degree in Cybersecurity, IT, or related...	Network security, Threat detection, Firewalls, Pen...	5	2024-10-14

TABLE 4: Job Applications

app_id	user_id	job_id	applied_at	first_name	last_name	address	phone	email	city	state	postal_code	resume
1	6	5	2025-01-26 22:57:48	Pratyush	Yadu	Avanti Vihar, basant corner, a-14	9302672397	pratyush9392@gmail.com	NULL	NULL	NULL	uploads/resumes/679670949591e6.88849373.pdf
2	6	5	2025-01-26 23:27:37	Pratyush	Yadu	Avanti Vihar, basant corner, a-14	9302672397	pratyush9392@gmail.com	NULL	NULL	NULL	uploads/resumes/67967791c52e02.64391236.pdf
4	9	4	2025-01-27 11:51:49	Pratyush	Yadu	Avanti Vihar, basant corner, a-14	9302672397	pratyush9392@gmail.com	NULL	NULL	NULL	uploads/resumes/679725fd57c9d6.29826456.pdf

TABLE 5: Liked_jobs

		liked_id	user_id	job_id	liked_at
lit	Copy Delete	1	9	4	2025-01-13 23:29:50
lit	Copy Delete	2	9	5	2025-01-14 02:07:42
lit	Copy Delete	3	6	5	2025-01-26 18:01:25
lit	Copy Delete	4	6	4	2025-01-26 18:02:13
lit	Copy Delete	5	6	6	2025-01-26 18:42:25

TABLE 6: Review And rating

		review_id	user_id	rating	comment	reviewed_at
oy	Delete	6	9	5	Noice	2025-01-09 20:01:42
oy	Delete	8	6	5	hi	2025-01-09 20:18:15

DATA FLOW DIAGRAM

It is a graphical tool used to describe and analyze the flow of data though a system. It focuses on the data flowing into the system between process of in & out of data stores.

DFD's are of two types:

1. Physical DFD:

The physical DFD is a model of current system & is use to ensure that the current system has been clearly understood.

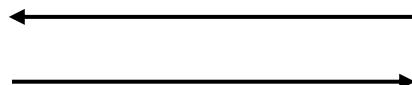
2. Logical DFD:

Logical DFD are the model of proposed system. They should clearly show the requirements on which the new system should built.

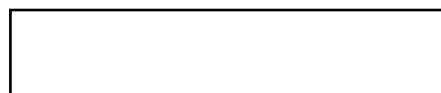
Notation Used In DFD:

There are four simple notations are used to complete DFD's

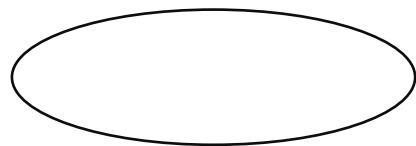
- Dataflow:



- External Entity:



- Process:

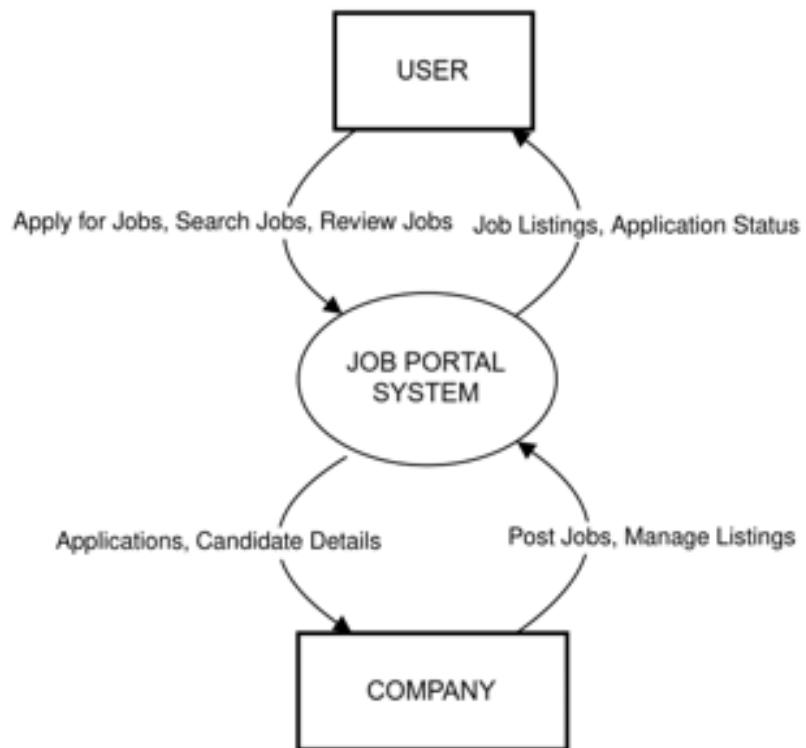


- Data store:

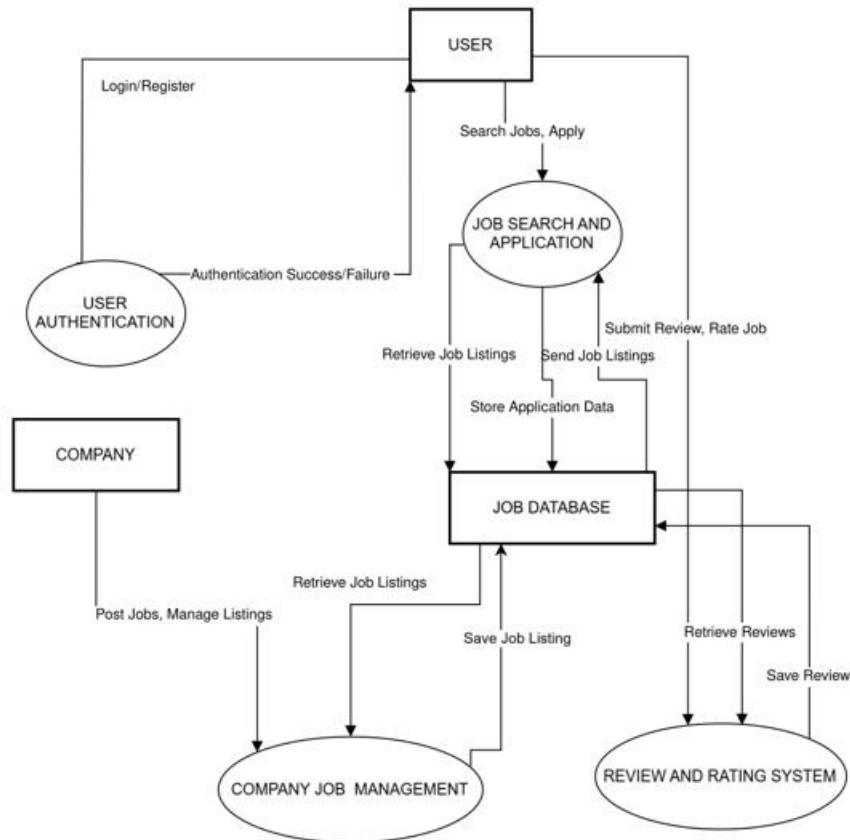


DFD FOR THE SYSTEM

Level 0:



Level 1:



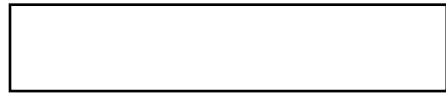
ENTITY RELATIONSHIP DIAGRAM

Entity Relationship Diagram (ERD) can express overall logical structure of a database graphically.

The components of E-R Diagram are:

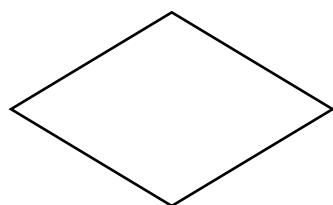
Entity:

Entity is a thing or object in a Real world that is Distinguishable from all other objects.



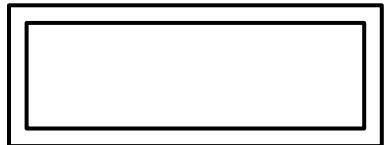
Relationship:

It is an association among several entities.

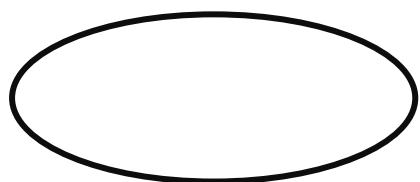


Weak Entity:

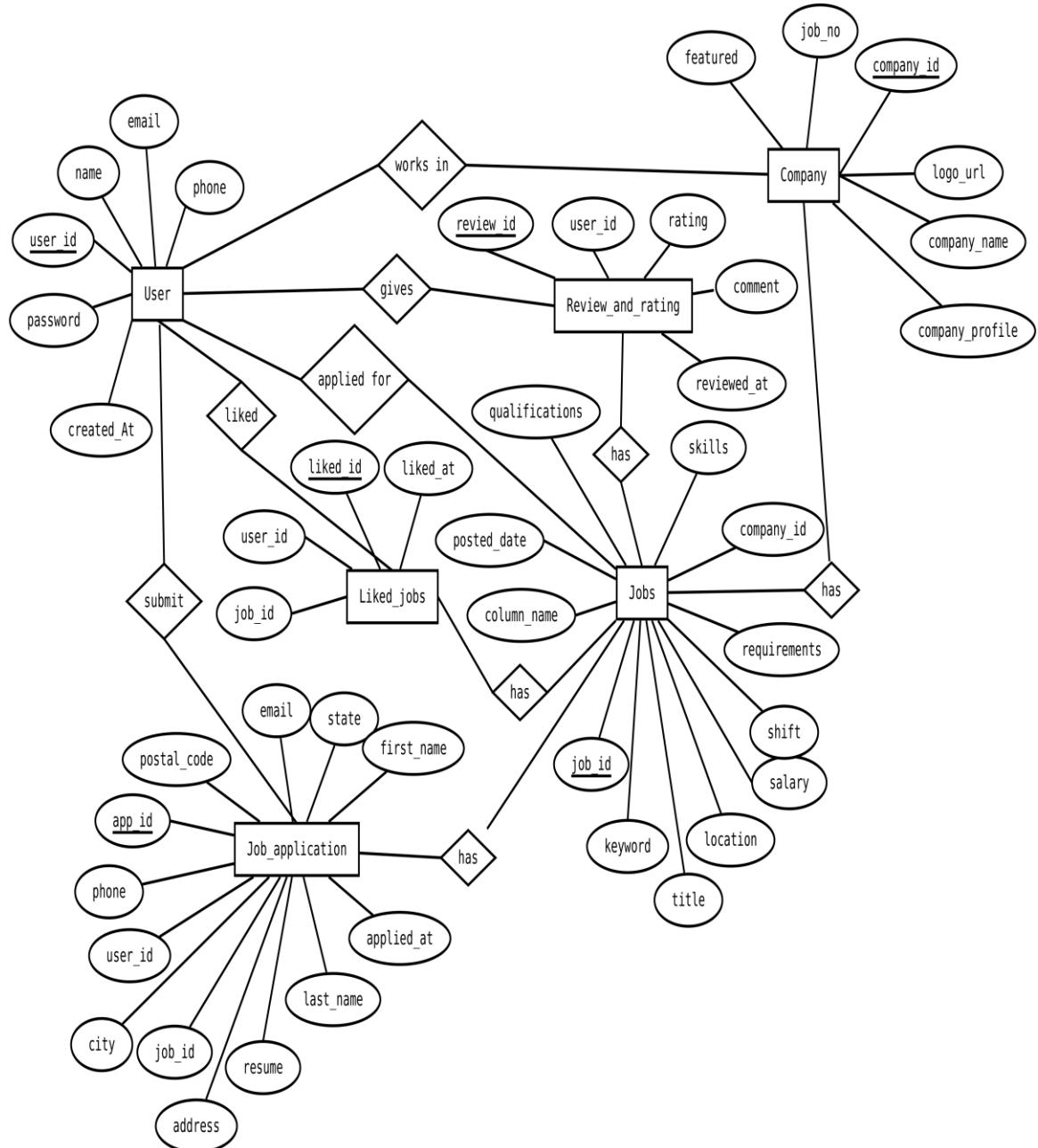
Weak entity is an entity that does not have any primary key.

**Attributes:**

A property or characteristic of an entity. Often shown as an oval or circle.

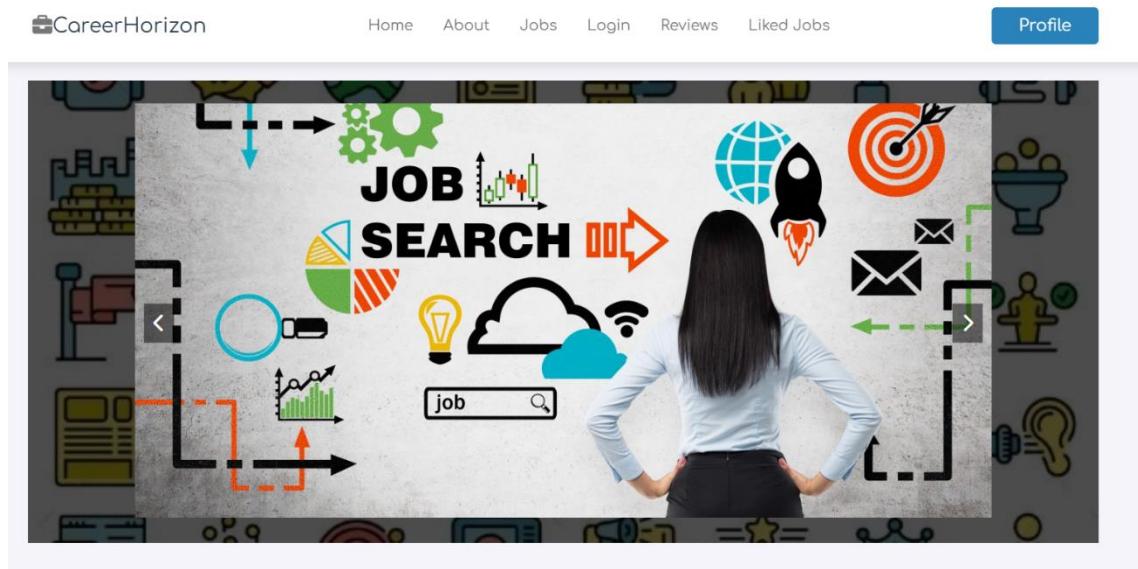


ENTITY RELATIONSHIP DIAGRAM



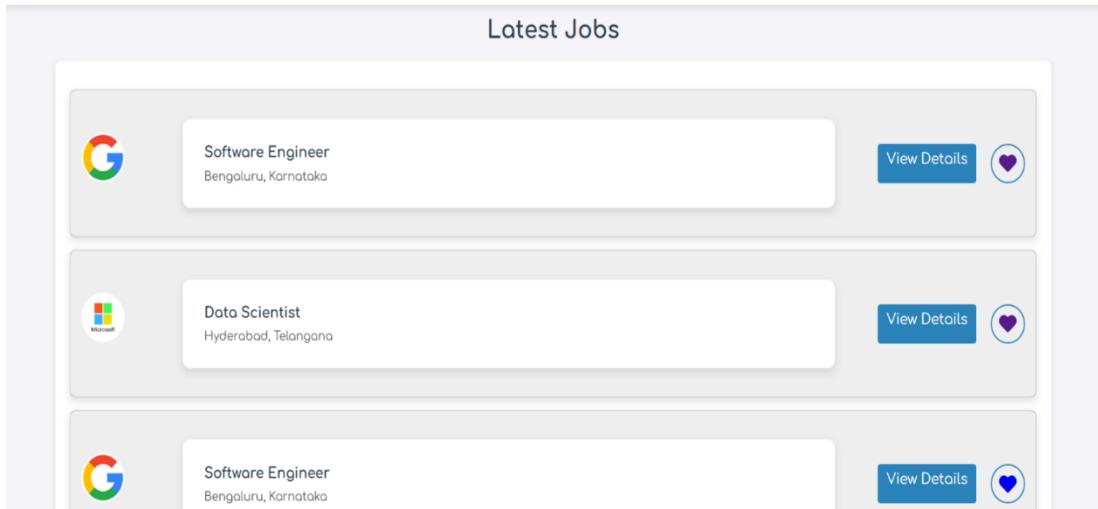
INPUT FORMS, PAGES AND CODING

Home Page:



Job Category

 Software Development 2200 jobs	 Data Analysis 1500 jobs	 Graphic Design 1200 jobs	 Digital Marketing 1800 jobs
 Teaching & Education 2000 jobs	 Healthcare Services 2500 jobs	 Banking & Finance 1700 jobs	 Sales & Marketing 1400 jobs
 Human Resources 1100 jobs	 Legal Services 900 jobs	 Customer Service 1200 jobs	 Media & Entertainment 1500 jobs



Code:

```

<?php
// Database connection
$conn = mysqli_connect("localhost", "root", "", "careerhorizon");

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Handle 'like' action
if (isset($_GET['like_job'])) {
    session_start();
    if (!isset($_SESSION['user_id'])) {
        // Redirect to login if the user is not logged in
        header("Location: login.php");
        exit();
    }
    $user_id = $_SESSION['user_id'];
    $job_id = intval($_GET['like_job']);

    // Check if the job is already liked
    $check_query = "SELECT * FROM liked_jobs WHERE user_id = $user_id AND job_id = $job_id";
    $check_result = mysqli_query($conn, $check_query);
    if (mysqli_num_rows($check_result) == 0) {
        // Insert into the liked_jobs table
        $like_query = "INSERT INTO liked_jobs (user_id, job_id) VALUES ($user_id, $job_id)";
        mysqli_query($conn, $like_query);
    }
}

```

```

        }
    }
// Get search inputs
$title = isset($_GET['title']) ? $_GET['title'] : '';
$location = isset($_GET['location']) ? $_GET['location'] : '';
// SQL query with filters
$query = "
    SELECT jobs.*, company.company_name
    FROM jobs
    LEFT JOIN company ON jobs.company_id = company.company_id
    WHERE 1";
if (!empty($title)) {
    $query .= " AND jobs.title LIKE '%" . mysqli_real_escape_string($conn, $title) . "%'";
}
if (!empty($location)) {
    $query .= " AND jobs.location LIKE '%" . mysqli_real_escape_string($conn, $location) . "%'";
}
$result = mysqli_query($conn, $query);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>

    <!-- Header section starting -->
    <header class="header">
        <section class="flex">
            <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
                <a href="jobs.php">Jobs</a>
                <a href="login.php">Login</a>
                <a href="reviews.php">Reviews</a>
                <a href="Liked_jobs.php">Liked jobs</a>
            </nav>
            <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
    </header>

```

```

        </section>
    </header>
    <script>
        document.getElementById('menu-btn').addEventListener('click',function() {
            document.querySelector('nav').classList.toggle('active');
        });
    </script>
    <!-- Header section ending -->
    <!-- Home section starting -->
    <section class="home">
        <div class="home-container">
            <div class="image-slider">
                <div class="slider-images">
                    
                    
                    
                </div>
                <button class="arrow-btn left-btn"><i class="fas fa-chevron-left"></i></button>
                    <button class="arrow-btn right-btn"><i class="fas fa-chevron-right"></i></button>
                </div>
            </div>
        </div>
    </section>
    <script>
        const images = ['./images/marquee1.png', './images/marquee2.jpeg',
        './images/marquee3.jpg'];
        const marqueeImage = document.getElementById('marquee-image');
        let currentImageIndex = 0;
        function updateImage(index) {
            if (index < 0) {
                currentImageIndex = images.length - 1;
            } else if (index >= images.length) {
                currentImageIndex = 0;
            } else {
                currentImageIndex = index;
            }
            marqueeImage.src = images[currentImageIndex];
        }
        document.getElementById('prev-btn').addEventListener('click', () => {
            updateImage(currentImageIndex - 1);
        });
        document.getElementById('next-btn').addEventListener('click', () => {
            updateImage(currentImageIndex + 1);
        });
    </script>
    <!-- Home section ending -->

```

```

<!-- Additional content here -->
</body>
</html>
</a>
<a href="#" class="box">
    <i class="fas fa-gavel"></i>
    <div>
        <h3>Legal Services</h3>
        <span>900 jobs</span>
    </div>
</a>
<a href="#" class="box">
    <i class="fas fa-headset"></i>
    <div>
        <h3>Customer Service</h3>
        <span>1200 jobs</span>
    </div>
</a>
<a href="#" class="box">
    <i class="fas fa-film"></i>
    <div>
        <h3>Media & Entertainment</h3>
        <span>1500 jobs</span>
    </div>
</a>
</div>
</section>
<!-- category section ending -->
<!-- job section starting -->
<h1 class="heading">Latest Jobs</h1>
<div class="container">
<?php
    if (mysqli_num_rows($result) > 0) {
        while ($row = mysqli_fetch_assoc($result)) {
            $logo_url = isset($row['logo_url']) && !empty($row['logo_url']) ? $row['logo_url'] : 'default-logo.png';
            echo '<div class="job-card">';
            echo '<div class="company-info">';
            echo '';
            echo '</div>';
            echo '<div class="job-details">';
            echo '<h3>' . htmlspecialchars($row['title']) . '</h3>';
            echo '<p>' . htmlspecialchars($row['location']) . '</p>';
            echo '</div>';
            echo '<div class="button-group">';

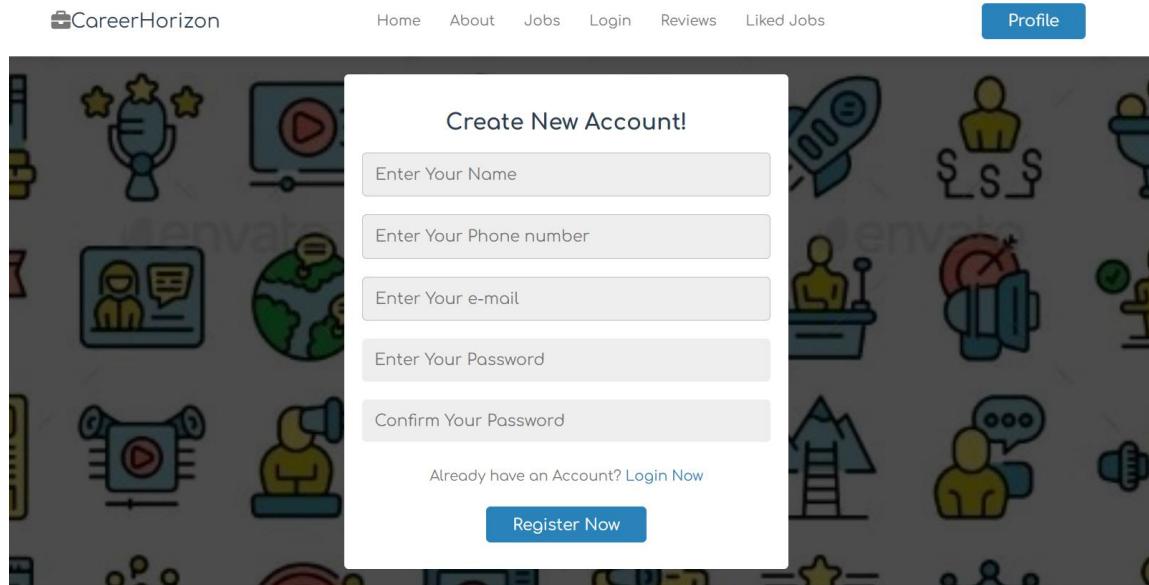
```

```

        echo '<a href="view_jobs_fixed.php?jobid=' . $row['jobid'] . '" class="view-details-btn">View Details</a>';
        echo '<a href="jobs.php?like_job=' . $row['jobid'] . '" class="heart-icon fas fa-heart"></a>';
        echo '</div>';
        echo '</div>';
    }
} else {
    echo '<p>No jobs found matching your criteria.</p>';
}
?>
<a href="jobs.php">
    <button class="btn">View All Jobs</button>
</a>
</div>
<!-- Footer section starting -->
<footer class="footer">
    <section class="grid">
        <div class="box">
            <h3>Quick links</h3>
            <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
            <a href="About.php"><i class="fas fa-angle-right"></i>About</a>
            <a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
            <a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
            <a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
        </div>
        <div class="box">
            <h3>Extra links</h3>
            <a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
            <a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
            <a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
        </div>
        <div class="box">
            <h3>Follow us</h3>
            <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
            <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
            <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
            <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
            <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
            <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
        </div>
    </section>
    <div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
        | All rights reserved
    </div>
</footer> <!-- Footer section ending --></body></html>

```

Register Form:



Code:

```
<?php
// Start session for error reporting
ini_set('display_errors', 1);
error_reporting(E_ALL);
// Include database connection file
include('Dbconnection.php');
if($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {
    // Get form data
    $name = $_POST['name'];
    $phone = $_POST['phone'];
    $email = $_POST['email'];
    $pass = $_POST['pass'];
    $confirm_pass = $_POST['c_pass'];
    // Validate input fields
    if (empty($name) || empty($phone) || empty($email) || empty($pass) ||
empty($confirm_pass)) {
        echo "All fields are required.";
        exit;
    }
    if ($pass !== $confirm_pass) {
        echo "Passwords do not match.";
```

```

        exit;
    }
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "Invalid email format.";
        exit;
    }
    // Hash the password
    $hashed_password = password_hash($pass, PASSWORD_DEFAULT);
    // Prepare SQL query to insert user data
    $sql = "INSERT INTO users (name, phone, email, password) VALUES (?, ?, ?, ?)";
    if ($stmt = $conn->prepare($sql)) {
        $stmt->bind_param("ssss", $name, $phone, $email, $hashed_password);
        if ($stmt->execute()) {
            echo "Registration successful!";
        } else {
            echo "Error: " . $stmt->error;
        }
        $stmt->close();
    } else {
        echo "Error preparing statement: " . $conn->error;
    }
    $conn->close();
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <!-- Header section starting -->
    <header class="header">
        <section class="flex">
            <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
                <a href="jobs.php">Jobs</a>
                <a href="login.php">Login</a>
                <a href="reviews.php">Reviews</a>
            </nav>
        </section>
    </header>

```

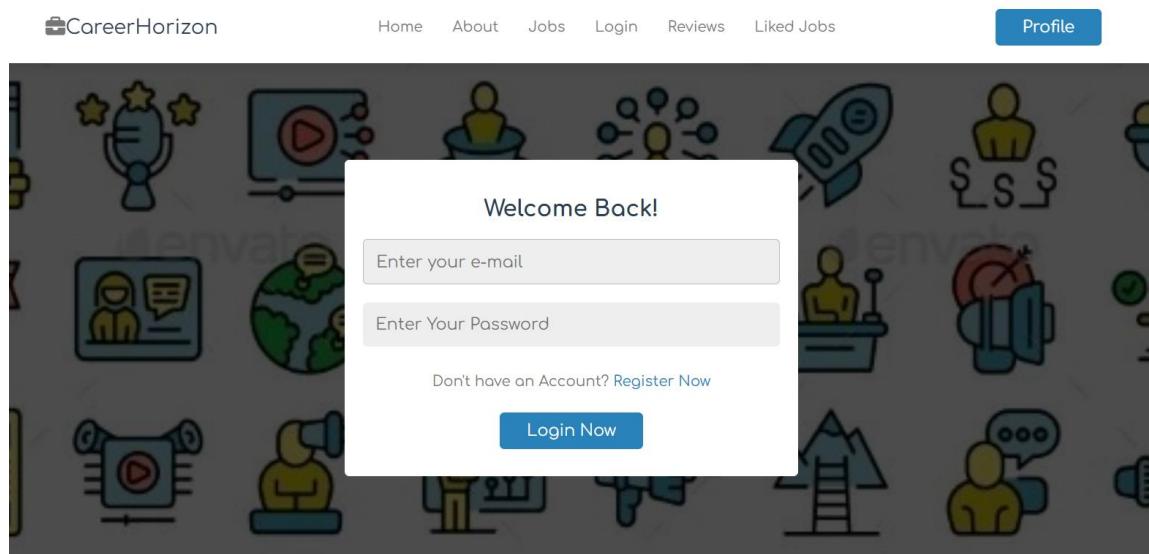
```

<!-- <a href="Admin.php">Admin</a> -->
    <a href="Liked_jobs.php">Liked jobs</a>
</nav>
    <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
</section>
</header>
<script>
    document.getElementById('menu-btn').addEventListener('click', function() {
        document.querySelector('nav').classList.toggle('active');
    });
</script>
<!-- Header section ending -->
<!-- Account creation form starting -->
<div class="account-form-container">
    <section class="account-form">
        <form action="" method="post">
            <h3>Create new account!</h3>
            <input type="text" required name="name" maxlength="20" placeholder="Enter Your Name" class="input">
            <input type="text" required name="phone" maxlength="20" placeholder="Enter Your Phone number" class="input">
            <input type="email" required name="email" maxlength="50" placeholder="Enter Your e-mail" class="input">
            <input type="password" required name="pass" maxlength="20" placeholder="Enter Your Password" class="input">
            <input type="password" required name="c_pass" maxlength="20" placeholder="Confirm Your Password" class="input">
            <p>Already have an Account? <a href="login.php">Login Now</a></p>
            <input type="submit" value="Register now" name="submit" class="btn">
        </form>
    </section>
</div>
<!-- Account creation form ending -->
<!-- Footer section starting -->
<footer class="footer">
    <section class="grid">
        <div class="box">
            <h3>Quick links</h3>
            <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
            <a href="About.php"><i class="fas fa-angle-right"></i>About</a>
            <a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
            <a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
            <a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
        </div>
        <div class="box">
            <h3>Extra links</h3>

```

```
<a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
<a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
<a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
</div>
<div class="box">
    <h3>Follow us</h3>
    <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
    <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
    <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
    <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
    <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
    <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>
</section>
<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
    | All rights reserved
</div>
</footer>
<!-- Footer section ending -->
</body>
</html>
```

Login Form:



Code:

```
<?php
session_start();
include('Dbconnection.php'); // Include database connection
// Initialize variables
$error_message = "";
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['submit'])) {
    // Get the user input
    $email = trim($_POST['email']);
    $password = trim($_POST['pass']);
    // Validate input
    if (empty($email) || empty($password)) {
        $error_message = "Both email and password are required!";
    } else {
        // Check if email and password match the admin credentials
        if ($email == "Admin@gmail.com" && $password == "admin") {
            // Store admin details in session
            $_SESSION['user_email'] = $email;
            $_SESSION['user_name'] = "Admin"; // Can be modified as per requirement
            // Redirect to Admin.php
        }
    }
}
echo "<script>
```

```

        alert('Login successful!');
        window.location.href = 'Admin.php';
        </script>";
    exit();
} else {
    // Prepare SQL query to fetch the user by email (for non-admin users)
    $sql = "SELECT userid, name, email, phone, password FROM users WHERE email
= ?";
    if ($stmt = $conn->prepare($sql)) {
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $stmt->store_result();
        if ($stmt->num_rows == 1) {
            $stmt->bind_result($user_id, $user_name, $user_email, $user_phone,
$hashed_password);
            $stmt->fetch();
            // Verify password
            if (password_verify($password, $hashed_password)) {
                // Store user details in session
                $_SESSION['user_id'] = $user_id;
                $_SESSION['user_name'] = $user_name;
                $_SESSION['user_email'] = $user_email;
                $_SESSION['user_phone'] = $user_phone;
                // Redirect to home.php with an alert message
                echo "<script>
                    alert('Login successful!');
                    window.location.href = 'home.php';
                </script>";
                exit();
            } else {
                $error_message = "Invalid password.";
            }
        } else {
            $error_message = "No account found with that email.";
        }
        $stmt->close();
    } else {
        $error_message = "Error: " . $conn->error;
    }
}
$conn->close();
}
?>
<!DOCTYPE html>
<html lang="en">

```

```

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <!-- Header section starting -->
    <header class="header">
        <section class="flex">
            <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
                <a href="jobs.php">Jobs</a>
                <a href="login.php">Login</a>
                <a href="reviews.php">Reviews</a>
                <!-- <a href="Admin.php">Admin</a> -->
                <a href="Liked_jobs.php">Liked jobs</a>
            </nav>
            <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
        </section>
    </header>
    <script>
        document.getElementById('menu-btn').addEventListener('click', function() {
            document.querySelector('nav').classList.toggle('active');
        });
    </script>
    <!-- Account section starts -->
    <div class="account-form-container">
        <section class="account-form">
            <form action="login.php" method="post">
                <h3>Welcome back!</h3>
                <?php
                    // Display the error message if exists
                    if (!empty($error_message)) {
                        echo "<div style='color: red; font-weight: bold; margin-bottom: 10px;'>$error_message</div>";
                    }
                ?>
                <input type="email" required name="email" maxlength="50" placeholder="Enter your e-mail" class="input">
        </section>
    </div>

```

```

        <input type="password" required name="pass" maxlength="20"
placeholder="Enter Your Password" class="input">
        <p>Don't have an Account? <a href="register.php">Register Now</a></p>
        <input type="submit" value="Login now" name="submit" class="btn">
    </form>
</section>
</div>
<!-- Account section ends -->
<!-- Footer section starting -->
<footer class="footer">
<section class="grid">
<div class="box">
    <h3>Quick links</h3>
    <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
    <a href="About.php"><i class="fas fa-angle-right"></i>About</a>
    <a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
    <a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
    <a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
</div>
<div class="box">
    <h3>Extra links</h3>
    <a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
    <a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
    <a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
</div>
<div class="box">
    <h3>Follow us</h3>
    <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
    <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
    <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
    <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
    <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
    <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>
</section>
<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
| All rights reserved
</div>
</footer>
<!-- Footer section ending -->
</body>
</html>

```

Admin Dashboard:

The screenshot shows a clean, modern dashboard interface. At the top center, it says "Admin Dashboard". Below that is a row of four cards: "Registered Users" (3), "Newly Posted Jobs" (14), "Expired Jobs" (6), and "User Reviews" (2). Underneath these cards is a section titled "Recent Activities" which contains a table with three rows of data.

Application ID	Job Title	Applicant	Applied At
4	Software Engineer	Pratyush Yadu	2025-01-27 11:51:49
2	Data Scientist	Pratyush Yadu	2025-01-26 23:27:37
1	Data Scientist	Pratyush Yadu	2025-01-26 22:57:48

Code:

```
<?php
// Include database connection
include 'Dbconnection.php';
// Fetch total registered users
$user_query = "SELECT COUNT(*) AS total_users FROM users";
$user_result = $conn->query($user_query);
$total_users = ($user_result->num_rows > 0) ? $user_result->fetch_assoc()['total_users'] : 0;
// Fetch total user reviews
$reviews_query = "SELECT COUNT(*) AS total_reviews FROM review_rating";
$reviews_result = $conn->query($reviews_query);
$total_reviews = ($reviews_result->num_rows > 0) ? $reviews_result->fetch_assoc()['total_reviews'] : 0;
// Fetch recent activities (latest job applications)
$activity_query =
    SELECT job_application.app_id, jobs.title AS job_title,
           CONCAT(job_application.first_name, ' ', job_application.last_name) AS applicant_name,
           job_application.applied_at
    FROM job_application
    JOIN jobs ON job_application.job_id = jobs.jobid
    ORDER BY job_application.applied_at DESC
    LIMIT 5";
$activity_result = $conn->query($activity_query);
?>
```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Panel</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <div class="admin-panel-container">
        <div class="admin-panel-header">
            <h1>Admin Dashboard</h1>
        </div>
        <div class="dashboard-section">
            <div class="dashboard-card">
                <h2>Registered Users</h2>
                <p><?php echo $total_users; ?></p>
            </div>
            <div class="dashboard-card">
                <h2>Newly Posted Jobs</h2>
                <p>14</p> <!-- Static Value -->
            </div>
            <div class="dashboard-card">
                <h2>Expired Jobs</h2>
                <p>6</p> <!-- Static Value -->
            </div>
            <div class="dashboard-card">
                <h2>User Reviews</h2>
                <p><?php echo $total_reviews; ?></p>
            </div>
        </div>
        <div class="data-tables-section">
            <h2>Recent Activities</h2>
            <table class="data-table">
                <thead>
                    <tr>
                        <th>Application ID</th>
                        <th>Job Title</th>
                        <th>Applicant</th>
                        <th>Applied At</th>
                    </tr>
                </thead>
                <tbody>
                    <?php if ($activity_result->num_rows > 0): ?>
                    <?php while ($row = $activity_result->fetch_assoc()): ?>
                        <tr>

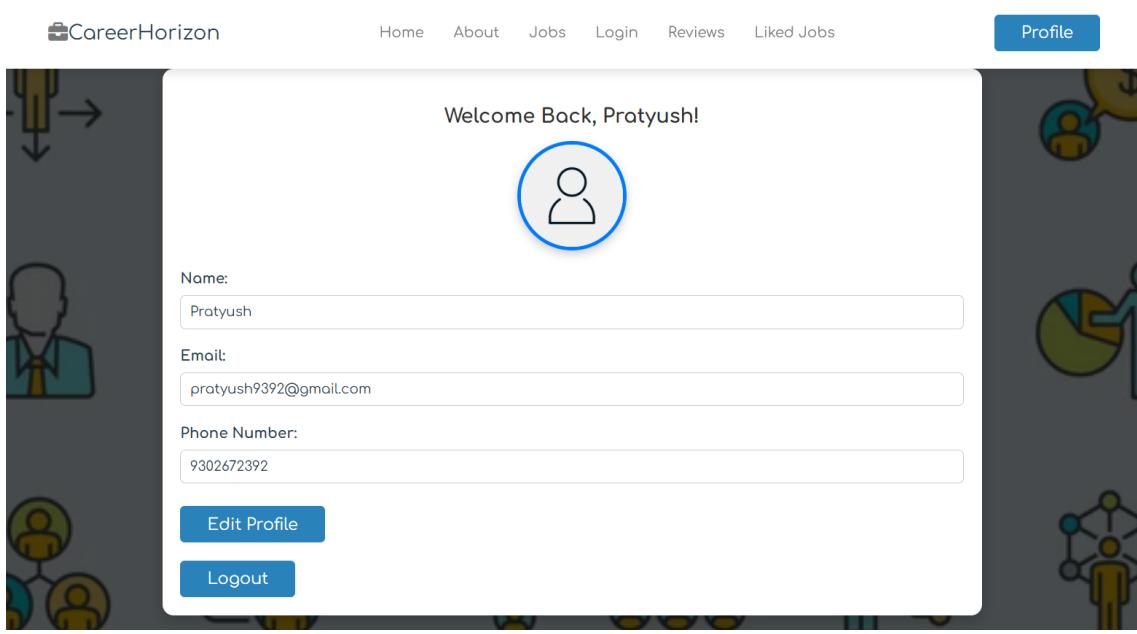
```

```

<td><?php echo htmlspecialchars($row['app_id']); ?></td>
<td><?php echo htmlspecialchars($row['job_title']); ?></td>
<td><?php echo htmlspecialchars($row['applicant_name']); ?></td>
<td><?php echo htmlspecialchars($row['applied_at']); ?></td>
</tr>
<?php endwhile; ?>
<?php else: ?>
<tr>
    <td colspan="4">No recent activities</td>
</tr>
<?php endif; ?>
</tbody>
</table>
</div>
</div>
<script>
function updateDashboard() {
    fetch('fetch_dashboard_data.php')
        .then(response => response.json())
        .then(data => {
            document.getElementById('totalUsers').innerText = data.total_users;
            document.getElementById('totalReviews').innerText = data.total_reviews;
            document.getElementById('recentActivities').innerHTML = data.recent_activities;
        })
        .catch(error => console.error('Error:', error));
}
// Refresh data every 5 seconds
setInterval(updateDashboard, 5000);
</script>
</body>
</html>

```

Profile Page:



Code:

```
<?php
session_start();
// Check if the user is logged in
if (!isset($_SESSION['user_id'])) {
    // Redirect to login page if not logged in
    header("Location: login.php");
    exit();
}
// Database connection
$servername = "localhost";
$username = "root";
$password = "";
$database = "careerhorizon";
$conn = new mysqli($servername, $username, $password, $database);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

```

// Handle logout logic
if (isset($_POST['logout'])) {
    // Destroy the session
    session_unset();
    session_destroy();
    // Redirect to login page
    header("Location: login.php");
    exit();
}
// Handle profile update
$message = ""; // Initialize message
if (isset($_POST['save_changes']))
{
    $user_id = $_SESSION['user_id'];
    $updated_name = $conn->real_escape_string($_POST['name']);
    $updated_email = $conn->real_escape_string($_POST['email']);
    $updated_phone = $conn->real_escape_string($_POST['phone']);
    // Update query
    $sql = "UPDATE users SET name = '$updated_name', email = '$updated_email', phone =
'$updated_phone' WHERE userid = '$user_id'";
    if ($conn->query($sql) === TRUE) {
        // Update session variables
        $_SESSION['user_name'] = $updated_name;
        $_SESSION['user_email'] = $updated_email;
        $_SESSION['user_phone'] = $updated_phone;
        $message = "Profile updated successfully!";
    } else {
        $message = "Error updating profile: " . $conn->error;
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Profile Page</title>
    <link rel="stylesheet" href="../css/styleproj.css">
    <script>
        // Enable editing when Edit Profile is clicked
        function enableEditing() {
            document.getElementById('name').removeAttribute('readonly');
            document.getElementById('email').removeAttribute('readonly');
            document.getElementById('phone').removeAttribute('readonly');
            document.getElementById('save-btn').style.display = 'inline-block'; // Show Save
Changes button

```

```

        document.getElementById('edit-btn').style.display = 'none'; // Hide Edit Profile
button
    }
</script>
</head>
<body class="ddd">
<header class="header">
    <section class="flex">
        <a href="#" class="logo">
            <i class="fas fa-briefcase"></i>CareerHorizon
        </a>
        <nav class="navbar">
            <a href="home.php">Home</a>
            <a href="about.php">About</a>
            <a href="jobs.php">Jobs</a>
            <a href="login.php">Login</a>
            <a href="reviews.php">Reviews</a>
            <a href="Liked_jobs.php">Liked jobs</a>
        </nav>
        <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
    </section>
</header>
<script>
    document.getElementById('menu-btn').addEventListener('click', function() {
        document.querySelector('nav').classList.toggle('active');
    });
</script>
<section class="profile-container">
    <h1 class="heading">Welcome <?php echo
htmlspecialchars($_SESSION['user_name']); ?></h1>
    <div id="profile-info">
        <div class="profile-pic">
            
        </div>
        <form action="profile.php" method="POST">
            <div class="user-info">
                <p>Name:</p>
                <input type="text" name="name" id="name" value="<?php echo
htmlspecialchars($_SESSION['user_name']); ?>" class="editable-input" readonly>
                <p>Email:</p>
                <input type="email" name="email" id="email" value="<?php echo
htmlspecialchars($_SESSION['user_email']); ?>" class="editable-input" readonly>
                <p>Phone Number:</p>
                <input type="tel" name="phone" id="phone" value="<?php echo
htmlspecialchars($_SESSION['user_phone']); ?>" class="editable-input" readonly>
            </div>

```

```
<div class="flex-btn">
    <!-- Edit Profile Button -->
    <button type="button" id="edit-btn" class="btn"
        onclick="enableEditing()">Edit Profile</button>
    <!-- Save Changes Button (Initially Hidden) -->
    <button type="submit" name="save_changes" id="save-btn" class="btn"
        style="display: none;">Save Changes</button>
</div>
</form>
<!-- Logout Button -->
<form action="profile.php" method="POST" style="margin-top: 10px;">
    <button type="submit" name="logout" class="btn">Logout</button>
</form>
</div>
<!-- Display messages -->
<?php if (!empty($message)): ?>
    <p style="color: green; text-align: center;"><?php echo
        htmlspecialchars($message); ?></p>
    <?php endif; ?>
</section>
</body>
</html>
```

About Page:

The screenshot shows the 'About' page of the Career Horizon website. At the top, there's a navigation bar with links for Home, About, Jobs, Login, Reviews, Liked Jobs, and a blue 'Profile' button. Below the navigation, there's a welcome message: "Welcome to Career Horizon—your gateway to a world of opportunities. We connect ambitious job seekers with the best companies, offering unparalleled opportunities to advance careers and secure lucrative packages. Whether you're a seasoned professional or just starting out, our platform is designed to help you utilize your skills to the fullest and achieve a fulfilling career." Under this message, there are three sections: 'Our Mission', 'Why Choose Career Horizon?', and '1. Diverse Job Listings'. Each section has a brief description and a corresponding image.

Welcome to Career Horizon—your gateway to a world of opportunities. We connect ambitious job seekers with the best companies, offering unparalleled opportunities to advance careers and secure lucrative packages. Whether you're a seasoned professional or just starting out, our platform is designed to help you utilize your skills to the fullest and achieve a fulfilling career.

Our Mission

At Career Horizon, we believe that everyone deserves a chance to pursue their dream job. Our mission is to provide a comprehensive platform that bridges the gap between talent and opportunity, helping individuals find jobs that align with their skills, values, and career aspirations.

Why Choose Career Horizon?

1. Diverse Job Listings

We offer a wide variety of job openings across multiple industries, ensuring that there's something for everyone. From tech giants to innovative startups, our listings are sourced from the best of the best.

2. Competitive Salaries

We understand that compensation is a crucial factor in job satisfaction. That's why we focus on roles that offer competitive packages, empowering you to achieve financial success while doing what you love.

3. Skill Utilization

Our platform prioritizes jobs that match your skills, ensuring you can contribute effectively and continue growing in your field. We aim to place you in positions where your talents are recognized and rewarded.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <!-- Header section starting -->
    <header class="header">
        <section class="flex">
            <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
```

```

<a href="jobs.php">Jobs</a>
<a href="login.php">Login</a>
<a href="reviews.php">Reviews</a>
<a href="Liked_jobs.php">Liked jobs</a>
</nav>
<a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
</section>
</header>
<script>
    document.getElementById('menu-btn').addEventListener('click', function() {
        document.querySelector('nav').classList.toggle('active');
    });
</script>
<!-- Header section ending -->
<!-- About section starting -->
<div class="section-title">about us</div>
<section class="about">
    
    <div class="box">
        <p>Welcome to Career Horizon—your gateway to a world of opportunities. We connect ambitious job seekers with the best companies, offering unparalleled opportunities to advance careers and secure lucrative packages. Whether you're a seasoned professional or just starting out, our platform is designed to help you utilize your skills to the fullest and achieve a fulfilling career.</p>
        <h3>Our Mission</h3>
        <p>At Career Horizon, we believe that everyone deserves a chance to pursue their dream job. Our mission is to provide a comprehensive platform that bridges the gap between talent and opportunity, helping individuals find jobs that align with their skills, values, and career aspirations.</p>
        <h3>Why Choose Career Horizon?</h3>
        <h4>1. Diverse Job Listings</h4>
        <p>We offer a wide variety of job openings across multiple industries, ensuring that there's something for everyone. From tech giants to innovative startups, our listings are sourced from the best of the best.</p>
        <h4>2. Competitive Salaries</h4>
        <p>We understand that compensation is a crucial factor in job satisfaction. That's why we focus on roles that offer competitive packages, empowering you to achieve financial success while doing what you love.</p>
        <h4>3. Skill Utilization</h4>
        <p>Our platform prioritizes jobs that match your skills, ensuring you can contribute effectively and continue growing in your field. We aim to place you in positions where your talents are recognized and rewarded.</p>
        <h4>4. Career Growth</h4>
        <p>Career Horizon is not just about finding a job—it's about building a career. We provide resources and guidance to help you navigate your career path, from resume tips to interview prep and beyond.</p>

```

<h4>5. User-Friendly Experience</h4>

<p>Our website is designed with you in mind. With intuitive navigation and robust search functionalities, finding your dream job has never been easier. Our user-friendly interface ensures a seamless experience from job search to application.</p>

<h3>Our Vision</h3>

<p>We envision a world where every individual has access to their ideal job, where career success is within reach for all. Career Horizon strives to be a catalyst for this change by continuously innovating and expanding our services to meet the evolving needs of job seekers.</p>

<h3>Join Us</h3>

<p>Embark on your career journey with Career Horizon today. Sign up to explore thousands of job listings, connect with top employers, and take the first step toward achieving your career goals. Whether you're looking for a part-time gig, a full-time position, or a leadership role, Career Horizon is here to support you every step of the way.</p>

<h3>Get Started</h3>

<p>Your dream job awaits. Dive into the world of opportunities with Career Horizon and take control of your future. With our platform, the horizon is bright, and your career is limitless. Join us today and let's build a future where your potential knows no bounds.</p>

</div>

</section>

<!-- About section ending -->

<!-- Footer section starting -->

<footer class="footer">

<section class="grid">

<div class="box">

<h3>Quick links</h3>

<i class="fas fa-angle-right"></i>Home

<i class="fas fa-angle-right"></i>About

<i class="fas fa-angle-right"></i>Jobs

<i class="fas fa-angle-right"></i>All Reviews

<i class="fas fa-angle-right"></i>Liked Jobs

</div>

<div class="box">

<h3>Extra links</h3>

<i class="fas fa-angle-right"></i>Profile

<i class="fas fa-angle-right"></i>Login

<i class="fas fa-angle-right"></i>Register

</div>

<div class="box">

<h3>Follow us</h3>

<i class="fab fa-facebook-f"></i>Facebook

<i class="fab fa-twitter"></i>Twitter

<i class="fab fa-instagram"></i>Instagram

<i class="fab fa-linkedin"></i>LinkedIn

```
<a href="#"><i class="fab fa-youtube"></i>Youtube</a>
<a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>
</section>
<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
| All rights reserved
</div>
</footer>
<!-- Footer section ending -->
</body>
</html>
```

Job Searching Form:

The screenshot shows a web application for job searching. At the top, there is a navigation bar with links for Home, About, Jobs, Login, Reviews, Liked Jobs, and a Profile button. Below the navigation bar, the main content area is titled "Available Jobs". It features two job listings. The first listing is for a "Software Engineer" position at "Bengaluru, Karnataka", associated with the Google logo. The second listing is for a "Data Scientist" position at "Hyderabad, Telangana", associated with the Microsoft logo. Each listing includes a "View Details" button and a purple heart icon for favoriting.

Code:

```
<?php
// Database connection
$conn = mysqli_connect("localhost", "root", "", "careerhorizon");
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// Handle 'like' action
if (isset($_GET['like_job'])) {
    session_start();
    if (!isset($_SESSION['user_id'])) {
        // Redirect to login if the user is not logged in
        header("Location: login.php");
        exit();
    }
    $user_id = $_SESSION['user_id'];
    $job_id = intval($_GET['like_job']);
    // Check if the job is already liked
    $check_query = "SELECT * FROM liked_jobs WHERE user_id = $user_id AND job_id = $job_id";
    $check_result = mysqli_query($conn, $check_query);
    if (mysqli_num_rows($check_result) == 0) {
        // Insert into the liked_jobs table
    }
}
```

```

        $like_query = "INSERT INTO liked_jobs (user_id, job_id) VALUES ($user_id,
$job_id)";
        mysqli_query($conn, $like_query);
    }
}

// Get search inputs
$title = isset($_GET['title']) ? $_GET['title'] : '';
.setLocation = isset($_GET['location']) ? $_GET['location'] : '';
// SQL query with filters
$query = "
    SELECT jobs.*, company.company_name
    FROM jobs
    LEFT JOIN company ON jobs.company_id = company.company_id
    WHERE 1";
if (!empty($title)) {
    $query .= " AND jobs.title LIKE '%" . mysqli_real_escape_string($conn, $title) . "%'";
}
if (!empty($location)) {
    $query .= " AND jobs.location LIKE '%" . mysqli_real_escape_string($conn, $location) . "%'";
}
$result = mysqli_query($conn, $query);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>All Jobs</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <header class="header">
        <section class="flex">
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
                <a href="jobs.php">Jobs</a>
                <a href="login.php">Login</a>
                <a href="reviews.php">Reviews</a>
                <a href="Liked_jobs.php">Liked jobs</a>
            </nav>
            <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
        </section>
    </header>

```

```

        </section>
    </header>
    <div class="container">
        <h1>Available Jobs</h1>
        <div class="search-bar">
            <form method="GET" action="">
                <input type="text" name="title" placeholder="Enter job title..." value="<?php
echo htmlspecialchars($title); ?>">
                <input type="text" name="location" placeholder="Enter location..." value="<?php echo htmlspecialchars($location); ?>">
                <button type="submit" class="search-btn">Search</button>
            </form>
        </div>
        <?php
if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $logo_url = isset($row['logo_url']) && !empty($row['logo_url']) ? $row['logo_url'] : 'default-logo.png';
        echo '<div class="job-card">';
        echo '<div class="company-info">';
        echo '';
        echo '</div>';
        echo '<div class="job-details">';
        echo '<h3>' . htmlspecialchars($row['title']) . '</h3>';
        echo '<p>' . htmlspecialchars($row['location']) . '</p>';
        echo '</div>';
        echo '<div class="button-group">';
        echo '<a href="view_jobs_fixed.php?jobid=' . $row['jobid'] . '" class="view-details-btn">View Details</a>';
        echo '<a href="jobs.php?like_job=' . $row['jobid'] . '" class="heart-icon fas fa-heart"></a>';
        echo '</div>';
        echo '</div>';
    }
} else {
    echo '<p>No jobs found matching your criteria.</p>';
}
?>
</div>
<?php mysqli_close($conn); ?>
<!-- Footer section starting -->
<footer class="footer">
<section class="grid">
<div class="box">
    <h3>Quick links</h3>

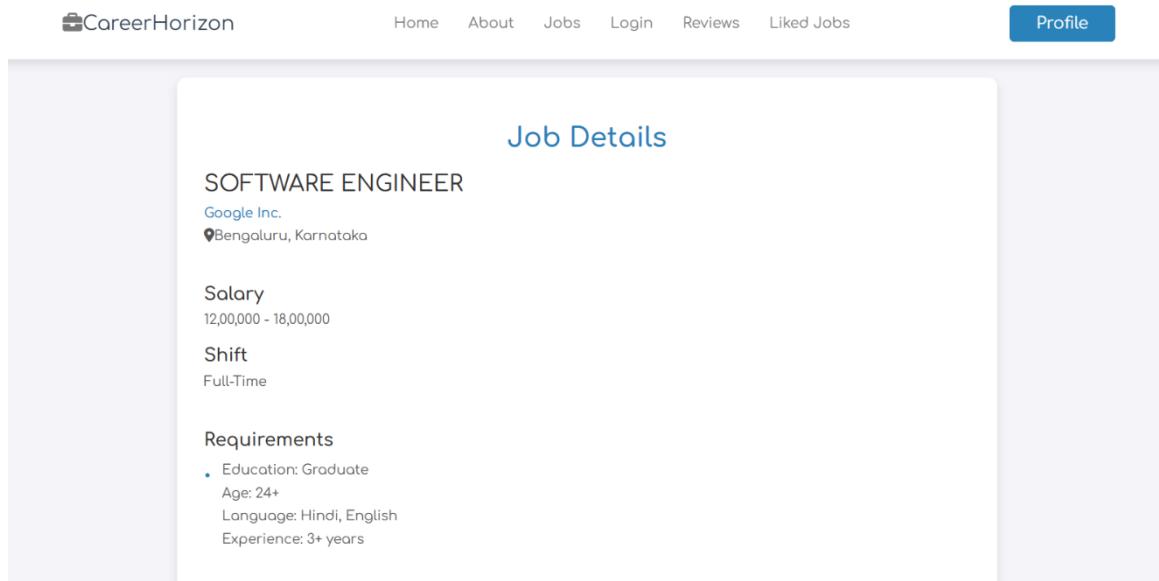
```

```

<a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
<a href="About.php"><i class="fas fa-angle-right"></i>About</a>
<a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
<a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
<a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
</div>
<div class="box">
<h3>Extra links</h3>
<a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
<a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
<a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
</div>
<div class="box">
<h3>Follow us</h3>
<a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
<a href="#"><i class="fab fa-twitter"></i>Twitter</a>
<a href="#"><i class="fab fa-instagram"></i>Instagram</a>
<a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
<a href="#"><i class="fab fa-youtube"></i>Youtube</a>
<a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>
</section>
<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
| All rights reserved
</div>
</footer>
<!-- Footer section ending -->
</body>
</html>

```

Job Details Page And Apply Form:



The screenshot shows a job details page for a Software Engineer position at Google Inc. in Bengaluru, Karnataka. The job pays between 12,00,000 - 18,00,000 and is full-time. Requirements include a graduate degree, age 24+, Hindi and English language skills, and 3+ years of experience.

Job Details

SOFTWARE ENGINEER

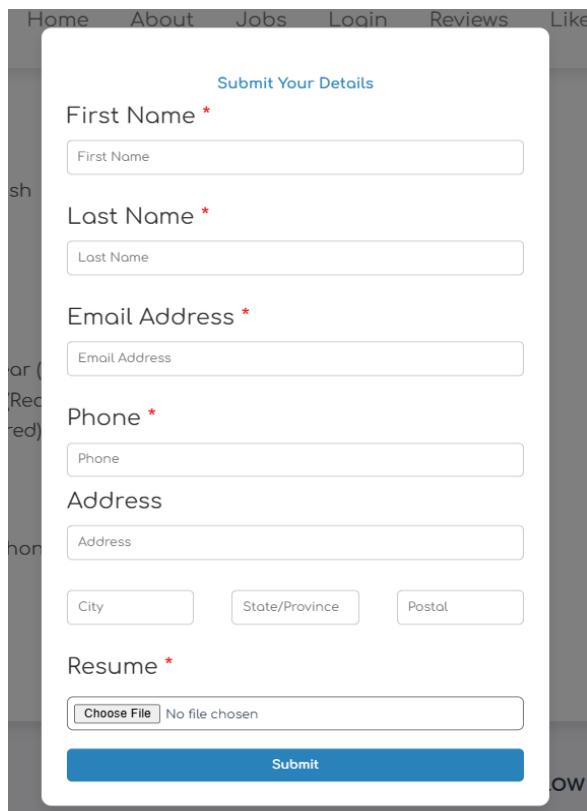
Google Inc.
Bengaluru, Karnataka

Salary
12,00,000 - 18,00,000

Shift
Full-Time

Requirements

- Education: Graduate
- Age: 24+
- Language: Hindi, English
- Experience: 3+ years



The screenshot shows an apply form overlay titled "Submit Your Details". It includes fields for First Name, Last Name, Email Address, Phone, Address, City, State/Province, Postal, and Resume, each with a required asterisk. A "Submit" button is at the bottom.

Submit Your Details

First Name *

Last Name *

Email Address *

Phone *

Address

City State/Province Postal

Resume *

No file chosen

Code:

```
<?php
// Include the database connection file
include('DbConnection.php');
// Start session to track user login status
session_start();
// Check if the jobid is provided and valid
if (isset($_GET['jobid']) && is_numeric($_GET['jobid'])) {
    $jobid = intval($_GET['jobid']); // Convert to integer
    // Ensure jobid starts from 4
    if ($jobid < 4) {
        echo "<h3>Error: Job ID must start from 4 or higher. Please go back and try again.</h3>";
        exit;
    }
    // Fetch job details from the database
    $query =
        SELECT
            jobs.*,
            company.company_name
        FROM
            jobs
        JOIN
            company
        ON
            jobs.company_id = company.company_id
        WHERE
            jobs.jobid = ?
    ";
    // Prepare and execute the query
    $stmt = $conn->prepare($query);
    $stmt->bind_param("i", $jobid); // Bind the job ID as integer
    $stmt->execute();
    $result = $stmt->get_result();
    $job = $result->fetch_assoc();
    // Check if the job exists
    if (!$job) {
        echo "<h3>Error: Job not found. Please go back and try again.</h3>";
        exit;
    }
} else {
    echo "<h3>Error: Job ID not provided or invalid. Please go back and try again.</h3>";
    exit;
}
// Handle saving the job to the user's saved jobs list
```

```

if($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['save_job'])) {
    if(isset($_SESSION['user_id'])) {
        $user_id = $_SESSION['user_id']; // Get the logged-in user's ID
        $job_id = $jobid; // Job ID to save
        // Check if the job is already saved
        $check_query = "SELECT * FROM liked_jobs WHERE user_id = ? AND job_id = ?";
        $check_stmt = $conn->prepare($check_query);
        $check_stmt->bind_param("ii", $user_id, $job_id);
        $check_stmt->execute();
        $check_result = $check_stmt->get_result();
        if($check_result->num_rows > 0) {
            echo "<script>alert('You have already saved this job.');//</script>";
        } else {
            // Save the job to the liked_jobs table
            $save_query = "INSERT INTO liked_jobs (user_id, job_id) VALUES (?, ?)";
            $save_stmt = $conn->prepare($save_query);
            $save_stmt->bind_param("ii", $user_id, $job_id);
            if($save_stmt->execute()) {
                echo "<script>alert('Job saved successfully.');//</script>";
            } else {
                echo "<script>alert('Error saving job. Please try again.');//</script>";
            }
        }
    } else {
        echo "<script>alert('You need to log in to save jobs.');//</script>";
    }
}
// Handle the job application form submission
if      ($_SERVER['REQUEST_METHOD']      ===      'POST'      &&
isset($_POST['submit_application'])) {
    if(isset($_SESSION['user_id'])) {
        $user_id = $_SESSION['user_id'];
        $job_id = $jobid;
        // Collect form data and sanitize
        $first_name = mysqli_real_escape_string($conn, $_POST['first_name']);
        $last_name = mysqli_real_escape_string($conn, $_POST['last_name']);
        $email = mysqli_real_escape_string($conn, $_POST['email']);
        $phone = mysqli_real_escape_string($conn, $_POST['phone']);
        $address = mysqli_real_escape_string($conn, $_POST['address']);
        // Handle the file upload
        if(isset($_FILES['resume'])) {
            $resume = $_FILES['resume'];
            $resume_name = $resume['name'];
            $resume_tmp_name = $resume['tmp_name'];
            $resume_size = $resume['size'];
        }
    }
}

```

```

$resume_error = $resume['error'];
// Validate file size (limit to 2MB for example)
if ($resume_size > 2 * 1024 * 1024) {
    echo "<script>alert('File size is too large. Please upload a resume under 2MB.');//</script>";
}
// Validate file type (PDF only)
$file_ext = strtolower(pathinfo($resume_name, PATHINFO_EXTENSION));
if ($file_ext != 'pdf') {
    echo "<script>alert('Please upload a valid PDF resume.');//</script>";
} else {
    // Generate a unique file name and move the uploaded file
    $resume_new_name = uniqid("", true) . '.' . $file_ext;
    $resume_dest = 'uploads/resumes/' . $resume_new_name;
    if (move_uploaded_file($resume_tmp_name, $resume_dest)) {
        // Insert application data into the database
        $insert_query =
            "INSERT INTO Job_application (user_id, job_id, first_name, last_name,
email, phone, address, resume)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
        $stmt = $conn->prepare($insert_query);
        $stmt->bind_param("iisssss", $user_id, $job_id, $first_name, $last_name,
$email, $phone, $address, $resume_dest);
        if ($stmt->execute()) {
            echo "<script>
                alert('Application submitted successfully.');
                window.location.href = 'jobs.php';
            </script>";
        } else {
            echo "<script>alert('Error saving job application. Please try again.');//</script>";
        }
    } else {
        echo "<script>alert('Error uploading resume. Please try again.');//</script>";
    }
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>View Job Details</title>
<link rel="stylesheet" href="../css/styleproj.css">
</head>
<body class="tree">
<!-- Header section starts -->
<header class="header">
    <section class="flex">
        <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
        <a href="#" class="logo">
            <i class="fas fa-briefcase"></i>CareerHorizon
        </a>
        <nav class="navbar">
            <a href="home.php">Home</a>
            <a href="about.php">About</a>
            <a href="jobs.php">Jobs</a>
            <a href="login.php">Login</a>
            <a href="reviews.php">Reviews</a>
            <a href="liked_jobs.php">Liked jobs</a>
        </nav>
        <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
    </section>
</header>
<script>
    document.getElementById('menu-btn').addEventListener('click', function() {
        document.querySelector('nav').classList.toggle('active');
    });
</script>
<!-- View job section starts -->
<section class="job-details">
    <h1 class="heading">Job Details</h1>
    <div class="details">
        <div class="job-info">
            <h3><?php echo htmlspecialchars($job['title']); ?></h3>
            <a href="view_company.php?company_id=<?php echo htmlspecialchars($job['company_id']); ?>">
                <?php echo htmlspecialchars($job['company_name']); ?>
            </a>
            <p><i class="fas fa-map-marker-alt"></i><?php echo htmlspecialchars($job['location']); ?></p>
        </div>
        <div class="basic-details">
            <h3>Salary</h3>
            <p><?php echo htmlspecialchars($job['salary']); ?></p>
            <h3>Shift</h3>
            <p><?php echo htmlspecialchars($job['shift']); ?></p>
        </div>
    </div>
</section>

```

```
</div>
<ul>
    <h3>Requirements</h3>
    <li><?php echo nl2br(htmlspecialchars($job['requirement'])); ?></li>
</ul>
<ul>
    <h3>Qualifications</h3>
    <li><?php echo nl2br(htmlspecialchars($job['qualification'])); ?></li>
</ul>
<ul>
<h3>Skills</h3>
    <li><?php echo nl2br(htmlspecialchars($job['skills'])); ?></li>
</ul>
</div>
<form id="apply-form" method="post" class="flex-btn">
    <button type="button" class="btn" id="apply-btn">Apply Now</button>
    <button type="submit" class="save" name="save_job">
        <i class="far fa-heart"></i>
        <span>Save job</span>
    </button>
</form>
</section>
<!-- Apply form modal starts --&gt;
&lt;div class="overlay" id="overlay"&gt;&lt;/div&gt;
&lt;section class="form-container hidden" id="form-section"&gt;
    &lt;h1 class="form-heading"&gt;Submit Your Details&lt;/h1&gt;
    &lt;form method="POST" action="view_jobs_fixed.php?jobid=&lt;?php echo $jobid; ?&gt;"&gt;
        &lt;div class="form-group"&gt;
            &lt;label for="first-name"&gt;First Name &lt;span class="required"&gt;*&lt;/span&gt;&lt;/label&gt;
            &lt;input type="text" name="first_name" id="first-name" placeholder="First Name" required&gt;
        &lt;/div&gt;
        &lt;div class="form-group"&gt;
            &lt;label for="last-name"&gt;Last Name &lt;span class="required"&gt;*&lt;/span&gt;&lt;/label&gt;
            &lt;input type="text" name="last_name" id="last-name" placeholder="Last Name" required&gt;
        &lt;/div&gt;
        &lt;div class="form-group"&gt;
            &lt;label for="email"&gt;Email Address &lt;span class="required"&gt;*&lt;/span&gt;&lt;/label&gt;
            &lt;input type="email" name="email" id="email" placeholder="Email Address" required&gt;
        &lt;/div&gt;
        &lt;div class="form-group"&gt;
            &lt;label for="phone"&gt;Phone &lt;span class="required"&gt;*&lt;/span&gt;&lt;/label&gt;
            &lt;input type="tel" name="phone" id="phone" placeholder="Phone" required&gt;
        &lt;/div&gt;
    &lt;/form&gt;
&lt;/section&gt;</pre>
```

```

        </div>
    <div class="form-group">
        <label for="address">Address</label>
        <input type="text" name="address" id="address" placeholder="Address">
        <div class="address-fields">
            <input type="text" name="city" placeholder="City">
            <input type="text" name="state" placeholder="State/Province">
            <input type="text" name="postal" placeholder="Postal">
        </div>
    </div>
    <div class="form-group">
        <label for="resume">Resume <span class="required">*</span></label>
        <input type="file" name="resume" id="resume-upload" accept=".pdf" required>
    </div>
    <button type="submit" name="submit_application" class="submit-button">Submit</button>
</form>
</section>
<script>
    // Open the form when clicking the Apply button
    document.getElementById('apply-btn').addEventListener('click', function () {
        const formSection = document.getElementById('form-section');
        const overlay = document.getElementById('overlay');
        // Show form and overlay
        formSection.classList.add('visible');
        overlay.classList.add('visible');
    });
    // Close the form when clicking outside it or on the overlay
    document.addEventListener('click', function (event) {
        const formSection = document.getElementById('form-section');
        const overlay = document.getElementById('overlay');
        // Close if clicked outside the form or on the overlay
        if (
            !formSection.contains(event.target) && // Click is outside the form
            !event.target.closest('#apply-btn') // Click is not the Apply button
        ) {
            formSection.classList.remove('visible');
            overlay.classList.remove('visible');
        }
    });
    // Trigger file upload on attach resume button click
    document.getElementById("attach-resume").addEventListener("click", function (event)
{
    event.preventDefault();
    document.getElementById("resume-upload").click();
});

```

```

// Handle resume upload and preview
document.getElementById("resume-upload").addEventListener("change", function () {
  const file = this.files[0];
  if (file && file.type === "application/pdf") {
    // Hide embedded PDF preview and display a link instead
    const pdfPreview = document.getElementById("pdf-preview");
    pdfPreview.style.display = "none"; // Hide PDF embed
    // Create a link to the PDF file for download
    const resumeLink = document.getElementById("resume-link");
    resumeLink.style.display = "block"; // Show the link
    resumeLink.href = URL.createObjectURL(file);
    resumeLink.textContent = "PDF uploaded: " + file.name; // Show file name
  } else {
    alert("Please upload a valid PDF file.");
  }
});
</script>
<!-- Footer section starting -->
<footer class="footer">
<section class="grid">
<div class="box">
  <h3>Quick links</h3>
  <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
  <a href="About.php"><i class="fas fa-angle-right"></i>About</a>
  <a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
  <a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
  <a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
</div>
<div class="box">
  <h3>Extra links</h3>
  <a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
  <a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
  <a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
</div>
<div class="box">
  <h3>Follow us</h3>
  <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
  <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
  <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
  <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
  <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
  <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>
</section>
<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
| All rights reserved

```

```
</div>
</footer>
<!-- Footer section ending -->
</body>
</html>
```

Company Details Page:

The screenshot shows a web application interface. At the top, there is a navigation bar with the logo 'CareerHorizon' and links for Home, About, Jobs, Login, Reviews, Liked Jobs, and a blue 'Profile' button. Below the navigation bar, the main content area has a title 'Company Details'. In the center, there is a logo placeholder for 'Google Inc.' featuring the Google 'G' logo. Below the logo, the company name 'Google Inc.' is displayed. To the left of the logo, under the heading 'About Company', there is a brief description of Google Inc. followed by a bullet point indicating '5 jobs posted'.

Code:

```
<?php
include 'Dbconnection.php';
include 'fetching.php';
// Get company_id from URL or set a default value
$company_id = isset($_GET['company_id']) ? (int)$_GET['company_id'] : 0;
if ($company_id <= 0) {
    die("Invalid or missing company ID.");
}
// Prepare SQL query to fetch company details
$query = "SELECT * FROM company WHERE company_id = ?";
$stmt = $conn->prepare($query);
$stmt->bind_param("i", $company_id);
$stmt->execute();
$result = $stmt->get_result();
// Check if any company data was returned
if ($result && $row = $result->fetch_assoc()) {
    $logo_url = $row['logo_url']; // Relative path stored in database
```

```

$company_name = $row['company_name'];
$company_profile = $row['company_profile'];
$job_no = $row['job_no'];
} else {
    echo "Company with ID $company_id not found!";
    exit;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Company Details</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
<header class="header">
    <section class="flex">
        <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
        <a href="#" class="logo">
            <i class="fas fa-briefcase"></i>CareerHorizon
        </a>
        <nav class="navbar">
            <a href="home.php">Home</a>
            <a href="about.php">About</a>
            <a href="jobs.php">Jobs</a>
            <a href="login.php">Login</a>
            <a href="reviews.php">Reviews</a>
            <!-- <a href="Admin.php">Admin</a> -->
            <a href="Liked_jobs.php">Liked jobs</a>
        </nav>
        <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
    </section>
</header>
<script>
    document.getElementById('menu-btn').addEventListener('click', function() {
        document.querySelector('nav').classList.toggle('active');
    });
</script>
<section class="view-company">
    <h1 class="heading">Company Details</h1>
    <div class="details">
        <div class="info">
            <!-- Update image path to include relative URL and domain -->

```

```

        ">
        <h3><?php echo htmlspecialchars($company_name); ?></h3>
        </div>
        <div class="description">
            <h3>About Company</h3>
            <p><?php echo nl2br(htmlspecialchars($company_profile)); ?></p>
            <ul>
                <li><?php echo htmlspecialchars($job_no); ?> jobs posted</li>
            </ul>
            </div>
        </div>
    </section>
    <!-- Footer section starting -->
    <footer class="footer">
        <section class="grid">
            <div class="box">
                <h3>Quick links</h3>
                <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
                <a href="About.php"><i class="fas fa-angle-right"></i>About</a>
                <a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
                <a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
                <a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
            </div>
            <div class="box">
                <h3>Extra links</h3>
                <a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
                <a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
                <a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
            </div>
            <div class="box">
                <h3>Follow us</h3>
                <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
                <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
                <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
                <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
                <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
                <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
            </div>
        </section>
        <div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
            | All rights reserved
        </div>
    </footer>
    <!-- Footer section ending -->
</body></html>

```

Liked Jobs Page:

The screenshot shows the 'Liked Jobs' section of the CareerHorizon website. At the top, there is a navigation bar with links for Home, About, Jobs, Login, Reviews, and Liked Jobs. A 'Profile' button is also visible. Below the navigation bar, the title 'Liked Jobs' is centered. The main content area displays two job listings in cards. The first card features a Google logo icon, the title 'Software Engineer', and the location 'Bengaluru, Karnataka'. It includes a 'View Details' button. The second card features a Microsoft logo icon, the title 'Data Scientist', and the location 'Hyderabad, Telangana'. It also includes a 'View Details' button.

Code:

```
<?php
session_start();
if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit();
}
$conn = mysqli_connect("localhost", "root", "", "careerhorizon");
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$user_id = $_SESSION['user_id'];
$query = "
    SELECT jobs.*, company.company_name
    FROM liked_jobs
    INNER JOIN jobs ON liked_jobs.job_id = jobs.jobid
    LEFT JOIN company ON jobs.company_id = company.company_id
"
```

```

        WHERE liked_jobs.user_id = $user_id";
$result = mysqli_query($conn, $query);
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Liked Jobs</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <header class="header">
        <section class="flex">
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
                <a href="jobs.php">Jobs</a>
                <a href="login.php">Login</a>
                <a href="reviews.php">Reviews</a>
                <a href="Liked_jobs.php">Liked jobs</a>
            </nav>
            <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
        </section>
    </header>
    <div class="container">
        <h1>Liked Jobs</h1>
        <?php

```

```

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $logo_url = isset($row['logo_url']) && !empty($row['logo_url']) ? $row['logo_url'] : 'default-logo.png';
        echo '<div class="job-card">';
        echo '<div class="company-info">';
        echo '';
        echo '</div>';
        echo '<div class="job-details">';
        echo '<h3>' . htmlspecialchars($row['title']) . '</h3>';
        echo '<p>' . htmlspecialchars($row['location']) . '</p>';
        echo '</div>';
        echo '<div class="button-group">';
        echo '<a href="view_jobs_fixed.php?jobid=' . $row['jobid'] . '" class="view-details-btn">View Details</a>';
        echo '</div>';
        echo '</div>';
    }
} else {
    echo '<p>You have not liked any jobs yet.</p>';
}
?>
</div>
<?php mysqli_close($conn); ?>
<!-- Footer section starting -->
<footer class="footer">
<section class="grid">
<div class="box">
    <h3>Quick links</h3>
    <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>

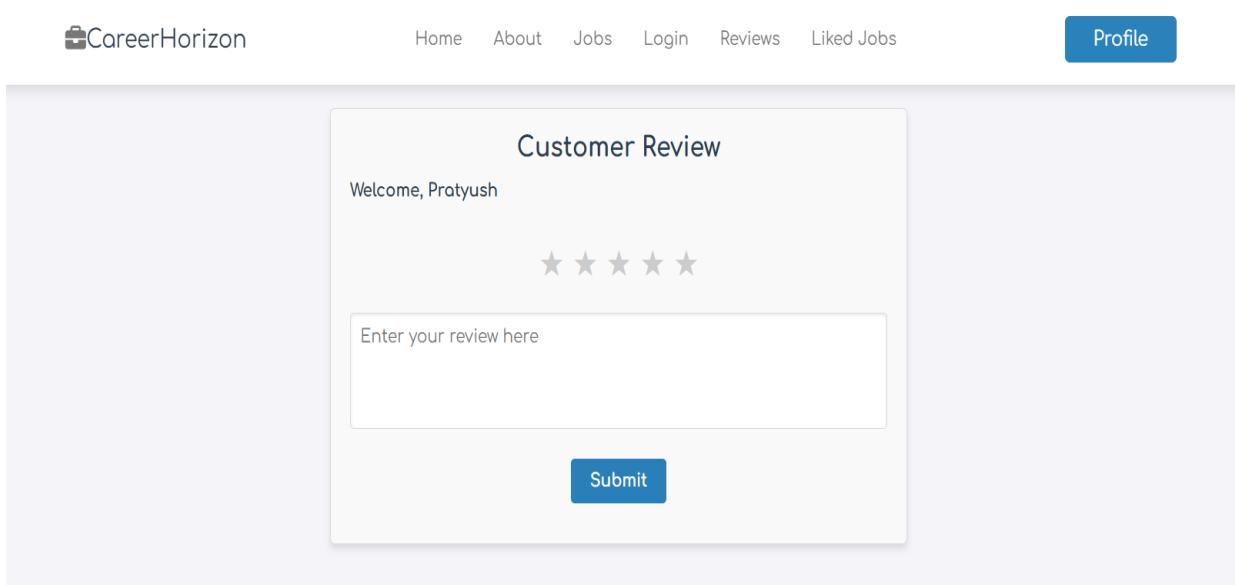
```

```

<a href="About.php"><i class="fas fa-angle-right"></i>About</a>
<a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
<a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
<a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
</div>
<div class="box">
    <h3>Extra links</h3>
    <a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
    <a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
    <a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
</div>
<div class="box">
    <h3>Follow us</h3>
    <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
    <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
    <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
    <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
    <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
    <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>
</section>
<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
    | All rights reserved
</div>
</footer>
<!-- Footer section ending -->
</body>
</html>

```

Review And Rating Page:



Code:

```
<?php
session_start();
include('Dbconnection.php');
// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    echo "You must be logged in to submit a review.";
    exit;
}
// Handle the form submission
if (isset($_POST['submit'])) {
    // Sanitize input to prevent SQL injection
    $user_id = $_SESSION['user_id']; // Assuming user ID is stored in session
    $rating = $_POST['rating'];    // Rating value
    $comment = mysqli_real_escape_string($conn, $_POST['review']); // Review comment
```

```

// Validate rating range
if ($rating < 1 || $rating > 5) {
    echo "Invalid rating. Please select a rating between 1 and 5.";
    exit;
}

// Check if the user has already submitted a review
$user_review_check_query = "SELECT * FROM review_rating WHERE user_id = ?";
$stmt_check = $conn->prepare($user_review_check_query);
$stmt_check->bind_param("i", $user_id);
$stmt_check->execute();
$result_check = $stmt_check->get_result();
if ($result_check && $result_check->num_rows > 0) {
    echo "You have already submitted a review.";
    exit;
}

// Insert the new review
$insert_query = "INSERT INTO review_rating (user_id, rating, comment) VALUES (?, ?, ?)";
$stmt_insert = $conn->prepare($insert_query);
$stmt_insert->bind_param("iis", $user_id, $rating, $comment);

if ($stmt_insert->execute()) {
    echo "Review submitted successfully.";
} else {
    echo "Error: " . $stmt_insert->error;
}

// Fetch reviews
$sql_fetch_reviews = "SELECT r.rating, r.comment, r.reviewed_at, u.name AS user_name
FROM review_rating r
JOIN users u ON r.user_id = u.userid"

```

```

        ORDER BY r.reviewed_at DESC";
$result_reviews = $conn->query($sql_fetch_reviews);
if (!$result_reviews) {
    echo "Error fetching reviews: " . $conn->error;
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Reviews</title>
    <link rel="stylesheet" href="../css/styleproj.css">
</head>
<body>
    <!-- Header section -->
    <header class="header">
        <section class="flex">
            <div id="menu-btn" class="fa-solid fa-bars-staggered"></div>
            <a href="#" class="logo">
                <i class="fas fa-briefcase"></i>CareerHorizon
            </a>
            <nav class="navbar">
                <a href="home.php">Home</a>
                <a href="about.php">About</a>
                <a href="jobs.php">Jobs</a>
                <a href="login.php">Login</a>
                <a href="reviews.php">Reviews</a>
                <!-- <a href="Admin.php">Admin</a> -->
                <a href="Liked_jobs.php">Liked jobs</a>
            </nav>
    </header>

```

```

        <a href="profile.php" class="btn" style="margin-top: 0;">Profile</a>
    </section>
</header>
<script>
    document.getElementById('menu-btn').addEventListener('click',function() {
        document.querySelector('nav').classList.toggle('active');
    });
</script>
<!-- Review section -->
<div class="review-section">
    <h2>Customer Review</h2>
    <h3>Welcome, <?php echo htmlspecialchars($_SESSION['user_name']); ?></h3>
    <form action="" method="POST">
        <!-- Star Rating Section -->
        <div class="rating">
            <input type="radio" name="rating" id="star5" value="5" required><label
for="star5" title="5 stars">#9733;</label>
            <input type="radio" name="rating" id="star4" value="4" required><label
for="star4" title="4 stars">#9733;</label>
            <input type="radio" name="rating" id="star3" value="3" required><label
for="star3" title="3 stars">#9733;</label>
            <input type="radio" name="rating" id="star2" value="2" required><label
for="star2" title="2 stars">#9733;</label>
            <input type="radio" name="rating" id="star1" value="1" required><label
for="star1" title="1 star">#9733;</label>
        </div>
        <textarea name="review" placeholder="Enter your review here"
required></textarea>
        <input type="submit" value="Submit" name="submit" class="btn">
    </form>
</div>

```

```

<section class="review">
<div class="review-display">
    <h3>Customer Reviews</h3>
    <?php
        // Display reviews
        if ($result_reviews && $result_reviews->num_rows > 0) {
            while ($row = $result_reviews->fetch_assoc()) {
                echo '<div class="review">';
                echo '<strong>' . htmlspecialchars($row['user_name']) . '</strong><br>';
                echo 'Rating: ' . str_repeat('★', $row['rating']) . '<br>';
                echo '<p>' . nl2br(htmlspecialchars($row['comment'])) . '</p>';
                echo '<small>Reviewed on: ' . $row['reviewed_at'] . '</small>';
                echo '</div><hr>';
            }
        } else {
            echo "No reviews yet.";
        }
    ?>
</div>
</section>
<!-- Footer section starting -->
<footer class="footer">
<section class="grid">
<div class="box">
    <h3>Quick links</h3>
    <a href="Home.php"><i class="fas fa-angle-right"></i>Home</a>
    <a href="About.php"><i class="fas fa-angle-right"></i>About</a>
    <a href="Jobs.php"><i class="fas fa-angle-right"></i>Jobs</a>
    <a href="reviews.php"><i class="fas fa-angle-right"></i>All Reviews</a>
    <a href="Liked_jobs.php"><i class="fas fa-angle-right"></i>Liked Jobs</a>
</div>

```

```
<div class="box">
    <h3>Extra links</h3>
    <a href="profile.php"> <i class="fas fa-angle-right"></i>Profile</a>
    <a href="Login.php"><i class="fas fa-angle-right"></i>Login</a>
    <a href="Register.php"><i class="fas fa-angle-right"></i>Register</a>
</div>

<div class="box">
    <h3>Follow us</h3>
    <a href="#"><i class="fab fa-facebook-f"></i>Facebook</a>
    <a href="#"><i class="fab fa-twitter"></i>Twitter</a>
    <a href="#"><i class="fab fa-instagram"></i>Instagram</a>
    <a href="#"><i class="fab fa-linkedin"></i>Linkdin</a>
    <a href="#"><i class="fab fa-youtube"></i>Youtube</a>
    <a href="#"><i class="fab fa-whatsapp"></i>Whatsapp</a>
</div>

</section>

<div class="credit">&copy; Copyright @ 2024 by <span>Pratyush Yadu</span>
    | All rights reserved
</div>

</footer>

<!-- Footer section ending -->

</body>
</html>
```

OUTPUT PAGES:

Register Form:

Create New Account!

Pratyush

123456789

pratyush9392@gmail.com

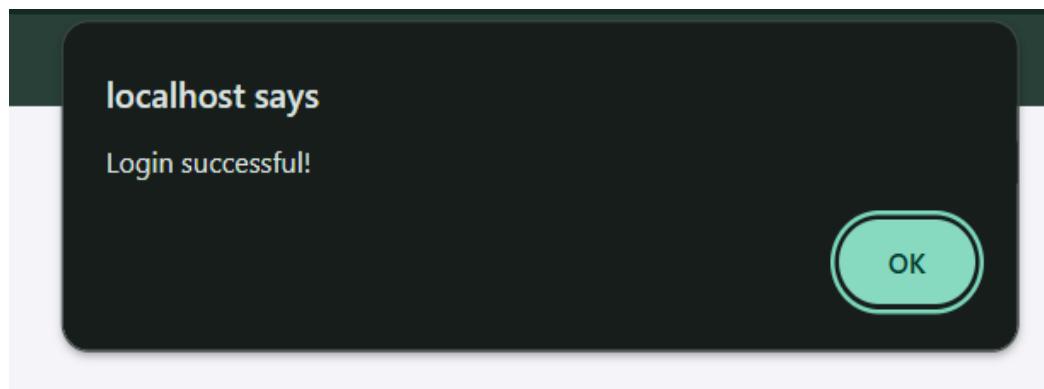
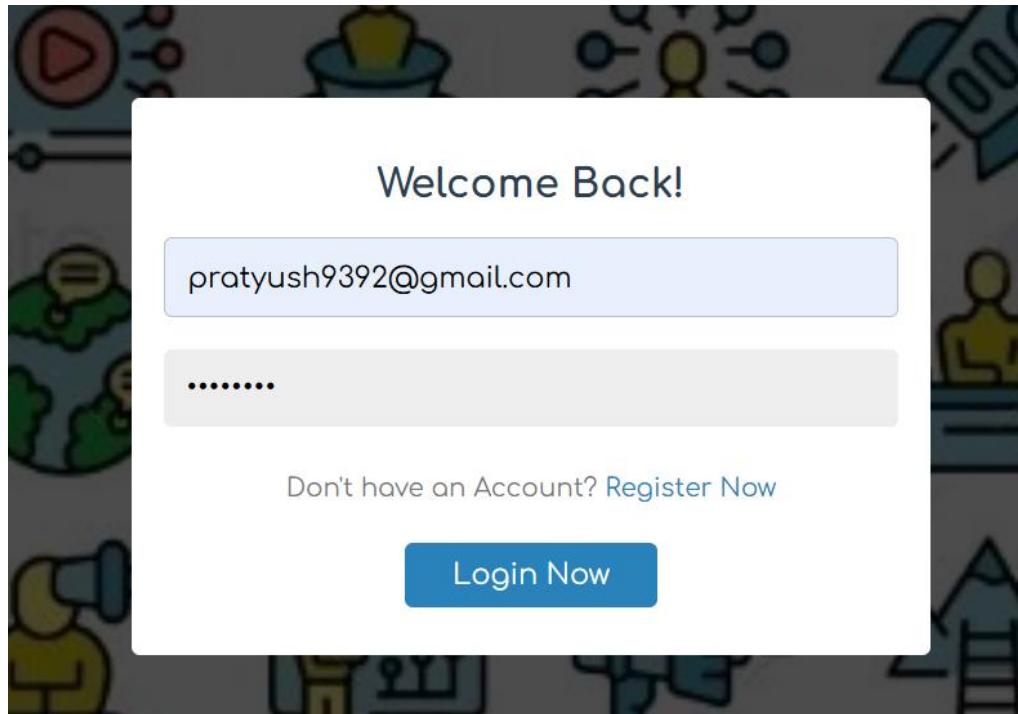
.....

.....

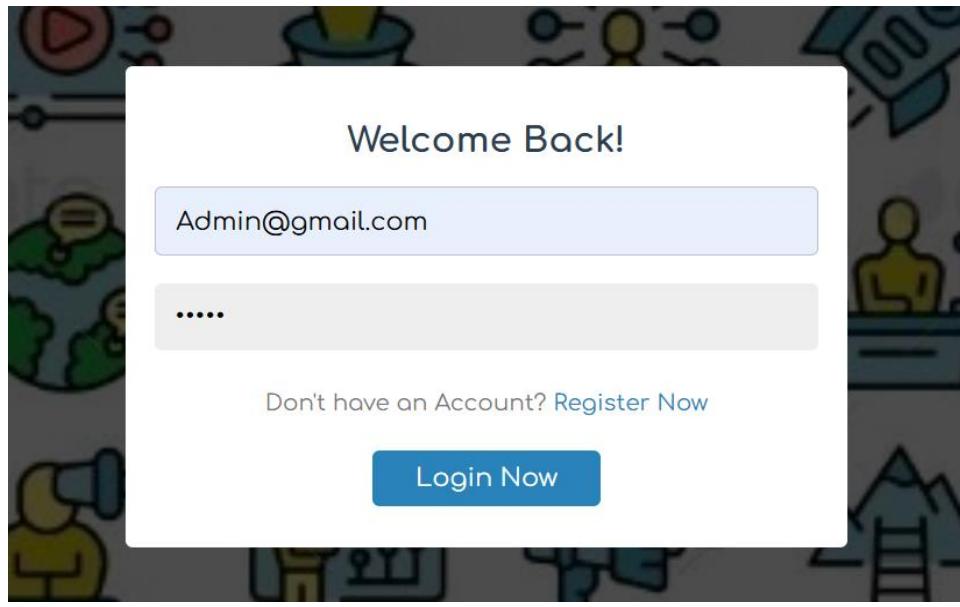
Already have an Account? [Login Now](#)

Register Now

Login Page:



Admin Login Page:



Job Searching Form:

A screenshot of a job search application. At the top, there is a navigation bar with a logo, "CareerHorizon", and links for "Home", "About", "Jobs", "Login", "Reviews", "Liked Jobs", and a "Profile" button. Below the navigation is a search section titled "Available Jobs" with three input fields: "Software", "Bengaluru", and a "Search" button. A job listing for a "Software Engineer" in "Bengaluru, Karnataka" is shown, featuring a "View Details" button and a heart icon. The overall design is clean and modern.

Apply Form:

Submit Your Details

First Name *

Last Name *

Email Address *

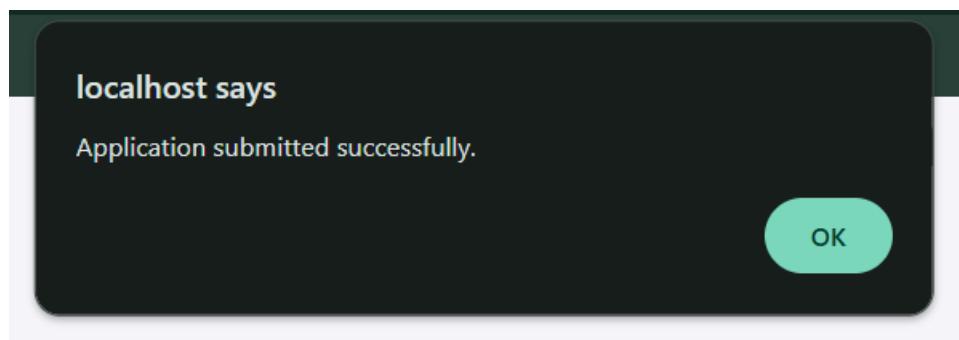
Phone *

Address

Raipur Chhattisgarh 492006

Resume *

pratyush_erd.pdf



Review And Rating Page:

Customer Review

Welcome, Pratyush

★★★★★

Noice

Submit

Customer Reviews

Prats

Rating: ★★★★

hii

Reviewed on: 2025-01-09 20:18:15

Pratyush

Rating: ★★★★

Noice

Reviewed on: 2025-01-09 20:01:42

LIMITATIONS

1. Lack of Multi-Language Support:

The website currently operates solely in English, which limits its usability for non-English-speaking users. This restriction significantly reduces the platform's reach and effectiveness in diverse linguistic regions. Users who are more comfortable in other languages may find it challenging to navigate and fully engage with the site, potentially excluding a large segment of the global job market.

2. Absence of Direct Communication Features:

The platform does not offer a direct chat or messaging feature for real-time communication between job seekers and employers. This limitation impedes immediate interaction, potentially causing delays in exchanging information and responding to queries. The lack of direct communication options may impact the efficiency of the hiring process and limit personalized interactions between users.

3. Basic Recommendation Engine:

The current recommendation engine is in its basic form, providing job suggestions based on user profiles and search histories but without advanced algorithms. As a result, the recommendations may lack accuracy and fail to cater to users' specific needs and preferences. This can lead to less relevant job matches and a suboptimal experience for users seeking precise job opportunities.

Future Scope of Career Horizon

1. AI-Driven Job Matching and Intelligent Recommendations:

Career Horizon can integrate artificial intelligence (AI) and machine learning algorithms to enhance job matching based on user profiles, skills, experience, and preferences. AI-driven recommendations can personalize job suggestions, improving candidate-employer alignment and hiring efficiency.

2. Advanced Search and Filtering Capabilities:

Implementing advanced search functionalities, including geolocation-based job searches, voice-enabled search, and refined filtering options (such as salary range, company ratings, remote work availability, and industry-specific roles), will enhance user experience and job discoverability.

3. Integration of Skill-Based Learning and Certifications:

Collaborating with e-learning platforms like Coursera, Udemy, or LinkedIn Learning can provide users with access to relevant courses and certifications. Incorporating skill assessment tools within the platform can help candidates validate their competencies and improve their employability.

4. Expansion into Freelance and Gig Economy Markets:

The platform can extend its services to support freelance, contract, and gig-based job opportunities, catering to the growing demand for flexible work arrangements. This expansion will allow professionals to find short-term projects and employers to source specialized talent efficiently.

5. AI-Powered Chatbots and Virtual Career Assistance:

Implementing AI-driven chatbots can enhance user engagement by providing 24/7 assistance for job seekers and employers. Virtual career assistants can offer resume-building suggestions, interview preparation guidance, and real-time support for queries related to job applications and recruitment.

CONCLUSION

The **Career Horizon - Job Portal** project successfully addresses the key objectives of providing a comprehensive and user-friendly platform for job seekers and employers. By offering a wide range of job listings, personalized job recommendations, and the ability to save liked jobs, the platform enhances the job search experience for users. For employers, the platform simplifies the process of posting jobs and managing applications, making recruitment more efficient.

The scope of the project includes creating a scalable system that can handle the growing demands of users, with a focus on secure user authentication, job search functionalities, and a responsive design. The **MVC architecture** ensures that the system is modular and maintainable, while the **MySQL database** allows for efficient management of job listings and user profiles. The project also adheres to basic security principles to protect user data and prevent unauthorized access.

In conclusion, **Career Horizon** provides an efficient, scalable, and secure solution for connecting job seekers with top companies. It meets the platform's goals by ensuring ease of use, responsive design, and streamlined workflows for both job seekers and employers. With future enhancements and regular updates, **Career Horizon** is poised to become a valuable tool in the competitive job market.

BIBLIOGRAPHY

The Bibliography contains reference to all the documents that were referred for the creation and successful completion of the project. It contains the names of the referred software engineering documents, visual basic technical and standards.

1. Book: Pankaj Jalote's Software Engineering.
2. Book: The Joy of PHP Programming: A Beginner's Guide by Alan Forbes.
3. Software Engineering: A Practitioner's Approach by Essman Roger, Tata McGraw Hill.
4. <https://fontawesome.com>
5. <https://flowbite.com>
6. <https://fonts.google.com>
7. <https://getbootstrap.com>
8. <https://cdnjs.com>
9. <https://nekocalc.com>
10. <https://plantuml.com>
11. <https://app.diagrams.net>
12. <https://www.youtube.com>