**Web Development React with Firebase**

**Quick Hire project**

Submitted in Partial fulfilment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in

Computer Science and Engineering

# 6 Weeks Summer Industrial Training Report

Submitted by

**Ashish Kumar**

**2323455**



**SESSION 2023-27**

# I.K. GUJRAL PUNJAB TECHNICAL

# UNIVERSITY,

**JALANDHAR**

**CERTIFICATE OF COMPLETION**

**DECLARATION**

The project report entitled “Quick Hire” was submitted by me to I.K. Gujral Punjab Technical University for the degree of Bachelor in Technology Sem. IV. It is an original piece of work and has not been submitted to any other university for the award of any degree. I also undertake that any quotation or philosophy from the published or unpublished work of another person has been duly acknowledged in the work that I present in the project report.

**Place: I.K. Gujral Punjab Technical University**

**Signed by**

**Ashish Kumar**

**ACKNOWLEDGEMENT**

I have accumulated a large number of debts in preparing this project. While a brief acknowledgment here in no way writes them off, it is a small courtesy whose sentiments are sincere. I would like to extend my sincere thanks to all the people who helped me in different ways with the development of this project report. Without their continuous support and guidance, the completion of my project would have been impossible.

I also wish to express my most sincere thanks to my supervisor Ms Janki Singh for her invaluable guidance, advice, support, and encouragement. I will carry out her guidance throughout my life. I shall be falling behind in my duties if I do not place on record my sincere thanks to all those writers and authors from whose writings I have benefited.

I would like to express my special gratitude and thanks to O7 Services Staff for giving me such attention and time.

In the end, I would also like to mention that this project would not have been possible but for the continuous support and guidance of my parents who gave me the strength and will to succeed. My thanks and appreciations also go to my friends in developing the project and the people who have willingly helped me out with their abilities.

Signed by

**Ashish Kumar**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Chapter | Contents | Page No. |
| 1 | **Introduction to the Industry / Institution** |  |
|  | 1.1 Nature of business of the Industry / Institution | 6 |
|  | 1.2 Different products / activities of the Industry / Institution | 7 |
| 2 | **Industrial Training** |  |
|  | 2.1 Industrial Training at a Glance | 8-10 |
|  | 2.2 Details on modules learned | 11-15 |
|  | 2.3 Details of Project work | 16-19 |
|  | 2.4 Hardware and Software Requirements | 20 |
| 3 | 3.1 Observations, including self-assessment report on contribution Results and Conclusions. | 21-24 |
| 4 | 4.1 Data Flow Diagrams | 25-30 |
|  | 4.2 Data Sheet | 31-34 |
| 5 | 5.1 Output | 35-40 |
|  | 5.2 Conclusion | 41 |

**CHAPTER-1 ABOUT THE INDUSTRY**

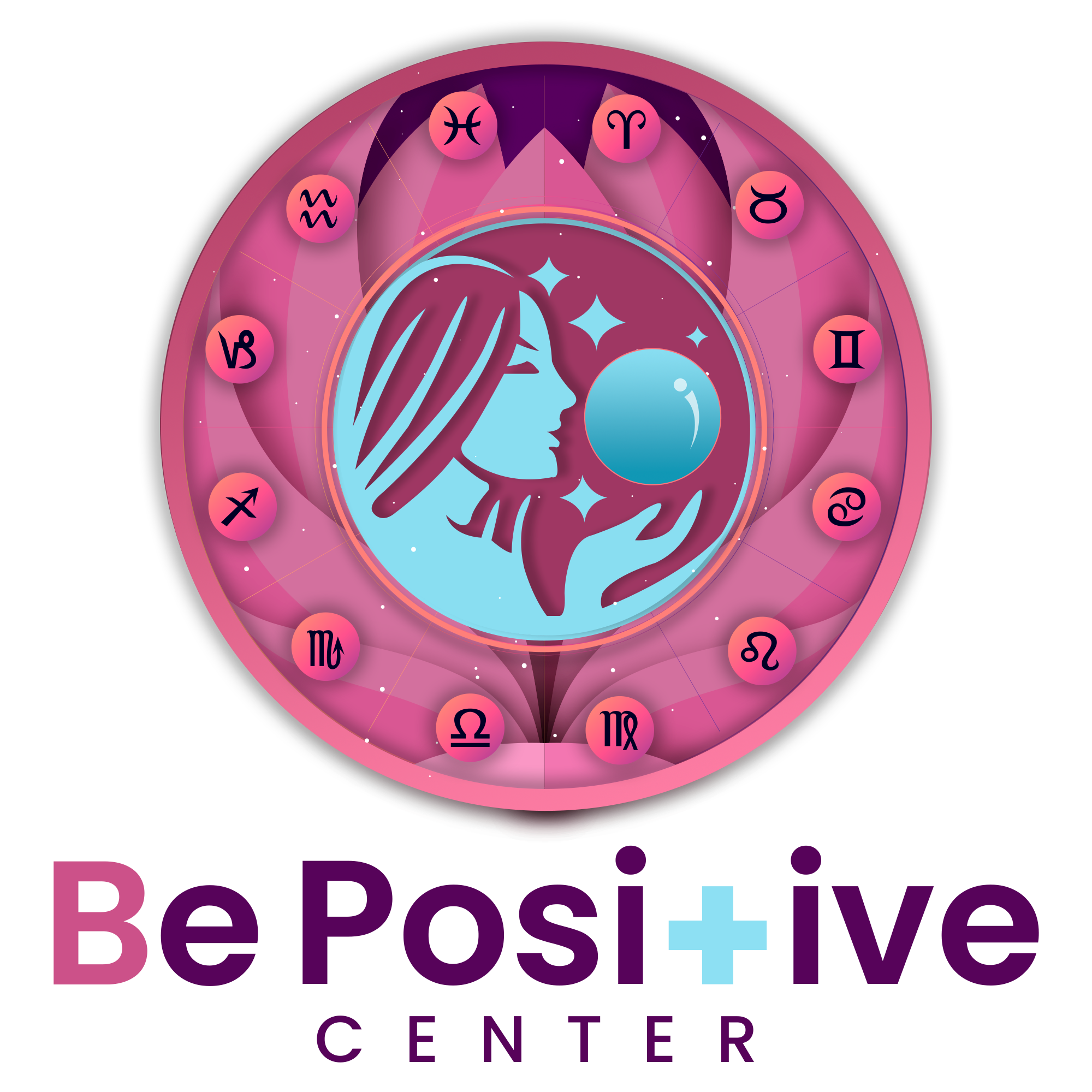


O7 Services is ISO 9001:2015 Certified company founded in 2015. They specialise in a variety of services including Web Development, Mobile Application Development, Custom Software Development, UI/UX Designing, Hosting services, Digital Marketing, Registration of Domain Names with various extensions, AMC & MMC Services, Bulk SMS and voice calls. O7 Services offers advanced IT solutions supporting the entire business cycle - from consulting to system development, deployment, quality assurance, and 24x7 support. With over 9 years of experience, the company aims to form long-lasting strategic partnerships with clients, offering affordable prices, timely delivery, and measurable business results. Their headquarter is located in Jalandhar with a branch office in Hoshiarpur.

Some of the products developed by O7 Services include Vehicle Tracking System, Invoice Software, School Management System, Hospital Management System, Parents - Teacher Communication App, Fee Management system, Task Management System, Online Food Ordering App, Security App, Admission system, Inventory Software, and Car Servicing App.

Additionally, O7 Services provides training programs including 6 Weeks/6 Months Industrial Training, Project-Based Training, Corporate Training, and Job-Oriented Courses Training, covering major IT trends such as Full Stack Development (MEAN/MERN), Flutter, Kotlin Android, Swift UI (iOS), Firebase, Python, Angular, React Js, Vue Js, Node Js, ASP.NET, .NET Core, PHP, Laravel, CodeIgniter, Software Testing, Cloud Computing, Blockchain, DevOps, Data Science, Artificial Intelligence, Machine Learning, UI/UX Designing, Digital Marketing, WordPress, Linux, CCNP, CCNA Security, Network Security, Cyber Security, MCSE, MCITP, Java, Spring, Hibernate, C/C++, Photoshop, Adobe Illustrator, Figma, CorelDraw and many more.

* +91- 8437365007, +91-181-5015007
* E-Mail: [enquiry@o7services.com](mailto:enquiry@o7services.com),[hr@o7services.com](mailto:hr@o7services.com)Website : [www.o7services.com](http://www.o7services.com)



## **CHAPTER-2 .1 Industrial Training at a Glance: Web Development**

### **Overview of the Field**

**Web development** remains a cornerstone of the digital era, enabling the creation, design, and deployment of interactive and dynamic web applications that power modern platforms and services. It involves both front-end (client-side) and back-end (cloud/server-side) development to deliver robust, user-friendly solutions.

My industrial training focused on mastering modern web development using a powerful combination of technologies — **React** for building dynamic and responsive user interfaces, and **Firebase** for implementing secure, scalable backend services like authentication, real-time databases, and cloud hosting.

**✅ Objectives of the Training**

**• Skill Enhancement:**

* Gain in-depth understanding of modern web development workflows and best practices.
* Develop strong proficiency in **React** for front-end development and **Firebase** for back-end integration.
* Build a solid foundation for creating responsive, scalable, and maintainable web applications.

**• Practical Application:**

* Apply theoretical knowledge through hands-on projects, solving real-world web development challenges.
* Strengthen debugging, optimization, and deployment skills in a live-like development environment.
* Follow industry best practices for clean, efficient, and secure code.

**• Industry-Relevant Expertise:**

* Stay up to date with the latest trends in modern web frameworks and cloud services.
* Understand the complete development lifecycle, from design and development to testing and deployment using modern tools.
* Build a professional portfolio showcasing practical projects that demonstrate expertise with React and Firebase.

**✅ Training Highlights**

**• Project-Based Learning:**

* Developed multiple projects simulating real-world scenarios, including user authentication, dynamic dashboards, and responsive layouts.
* Worked collaboratively with team members, enhancing project planning, teamwork, and communication skills.
* Delivered individual and group projects, ranging from simple interfaces to fully integrated web applications like **QuickHired**.

**• Comprehensive Coverage:**

* Gained in-depth knowledge of front-end technologies (**HTML**, **CSS**, **JavaScript**, and **React**) for creating engaging, component-based user interfaces.
* Learned to integrate the frontend with **Firebase**, implementing secure authentication, real-time databases, and cloud hosting.
* Explored the synergy between modern front-end frameworks and cloud services for building production-ready, scalable applications.

**• Tool Proficiency:**

* **React:** Utilized for creating reusable components, managing dynamic data with hooks and context, and building responsive UIs.
* **Firebase Authentication:** Implemented secure user login, registration, and role management.
* **Firebase Firestore:** Used for storing and syncing real-time data, including user profiles, job postings, and applications.
* **HTML & CSS:** Structured and styled web pages with responsive layouts for various devices.
* **JavaScript:** Added interactivity and handled dynamic functionality across components.

**✅ Conclusion**

The industrial training in **modern web development** has equipped me with the practical experience and comprehensive skills required to build high-quality, dynamic, and scalable web applications. By mastering **React** and **Firebase**, I have gained valuable expertise in creating responsive user interfaces, secure authentication systems, and real-time data interactions — all of which are essential for developing platforms like **Quick Hired**.

This training has laid a strong foundation for continuous learning and professional growth in the fast-evolving web development landscape, ensuring that I am well-prepared to contribute effectively to modern projects and adapt to emerging technologies in the tech industry.

## **CHAPTER 2.2 : MODULES LEARNED**

### **2.1 HTML**

HTML, or HyperText Markup Language, is the foundation of web pages.

* **HyperText**: Refers to the links that connect web pages.
* **Markup Language**: Consists of text surrounded by tags to define structure.

HTML defines the structure of web pages and the content displayed on them. It is essential to save HTML files with the .html extension.



### **2.2 CSS**

CSS, or Cascading Style Sheets, simplifies the process of making web pages presentable. It allows for styling web pages independently of the HTML content.

* **Benefits of CSS**:
  + **Saves Time**: Reuse the same CSS sheet across multiple HTML pages.
  + **Easy Maintenance**: Global changes can be made by modifying the CSS file, updating all related elements.
  + **Search Engine Friendly**: Clean coding techniques in CSS improve readability for search engines.
  + **Superior Styles**: Provides a wider array of styling options compared to HTML attributes.
  + **Offline Browsing**: Enables web applications to be stored locally for offline viewing.



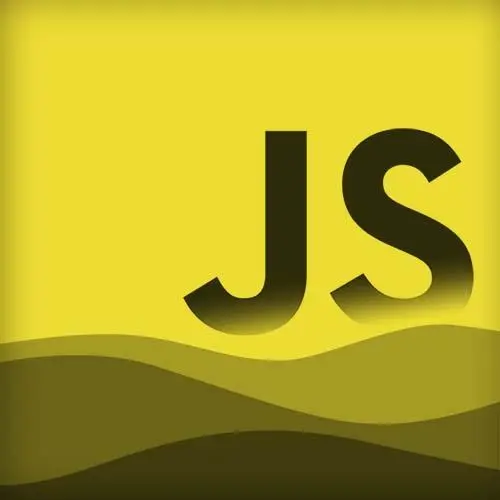
### **2.3 JavaScript**

JavaScript is a versatile, lightweight, and interpreted scripting language used primarily for web development. It can be employed for both client-side and server-side development.

* **Client-Side**: Manages browser and DOM interactions, handling user events and dynamic content.
* **Server-Side**: Supports server operations, database interactions, and file manipulations. Popular server-side framework: Node.js.

JavaScript can be added to HTML in two ways:

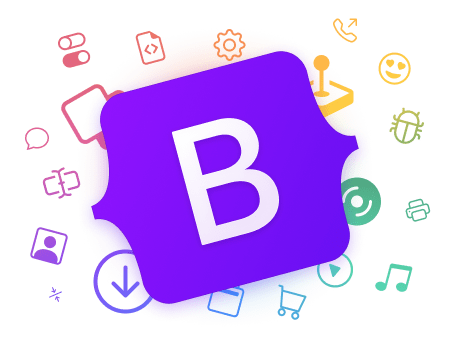
* **Internal JS**: Placed within the <script> tag inside the HTML file.
* **External JS**: Written in separate .js files and linked within the HTML file.



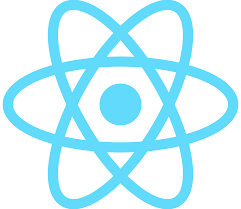
### **2.4Bootstrap**

Bootstrap is an open-source toolkit for developing responsive web applications. It is the most popular framework for creating mobile-first websites.

* **Benefits of Bootstrap**:
  + Speeds up web development with reusable components.
  + Ensures cross-browser compatibility and responsive design.
  + Free to use and well-documented, available at [getbootstrap.com](https://getbootstrap.com).
* **Usage**:
  + Include Bootstrap via CDN or download and use it locally.



**2.5 React**



**React** is a popular open-source JavaScript library used for building fast, interactive, and dynamic user interfaces, especially single-page applications (SPAs). It was developed by Facebook and is widely used in modern web development.

**• Components of React:**

* **Components:** Reusable, self-contained building blocks for creating UI sections.
* **JSX:** A syntax extension that allows writing HTML-like code within JavaScript, making UI development more intuitive.
* **State and Props:** Core concepts for managing data and dynamic rendering within components.
* **Virtual DOM:** A lightweight representation of the real DOM that enables efficient updates and rendering.

**React** enables developers to build complex, responsive interfaces for web applications like **QuickHired**, ensuring an engaging and seamless user experience for job seekers, recruiters, and admins.

**2.6 Firebase**



**Firebase** is a comprehensive Backend-as-a-Service (**BaaS**) platform developed by Google. It provides a suite of cloud-based tools and services that simplify backend development and help developers build secure, scalable web applications.

**• Key Firebase Components:**

* **Firebase Authentication:** Handles user sign-up, login, and secure authentication workflows for job seekers, recruiters, and admins.
* **Cloud Firestore:** A flexible, scalable NoSQL cloud database for storing and syncing user profiles, job listings, and applications in real-time.
* **Firebase Hosting:** Fast and secure web hosting for deploying modern web apps built with React.
* **Firebase Security Rules:** Ensure that only authorized users can access or modify sensitive data.
* **Cloud Functions:** Enable backend logic like sending notifications or handling complex queries without managing your own servers.

### **Chapter 2.3 Project Work**

### **1.1 INTRODUCTION**

### In the modern employment landscape, online job-seeking platforms stand as essential tools for connecting talent with opportunity and driving career growth. This project aims to enhance recruitment services by developing a dynamic and efficient job-seeking system tailored to the evolving needs of both job seekers and employers.

### **1.2 PROJECT DESCRIPTION**

### **Modules of Project:** Quick Hired consists of multiple modules working together to deliver a seamless job-seeking and recruitment experience.

### **1. ADMIN**

### The Admin Module provides centralized control over the entire platform. The admin manages recruiters, job seekers, job postings, and overall system settings.

### **Admin Login**: Secure access to manage user accounts, recruiter accounts,and platform settings.

### **User Management:** Add, update, or remove job seeker profiles.

### **Recruiter Management**: Approve, update, or remove recruiter accounts.

### **Job Posting Management:** Review, approve, or remove job listings.

### **Query Management**: Oversee and resolve queries raised by job seekers or recruiters.

### **2. RECRUITER**

### The Recruiter Module enables employers or recruiters to post jobs, manage postings, and interact with potential candidates.

### **Recruiter Registration/Login**: Recruiters register and log in securely to manage their postings.

### **Job Posting Creation**: Recruiters can create new job listings with detailed descriptions and requirements.

### **Job Management**: Edit, update, or remove existing job postings.

### **Application Management**: View and shortlist candidate applications for each job.

### **Query Management**: Raise queries or requests to the admin if needed**.**

### **Profile Management**: View and manage recruiter account details.

### **Logout:** Securely log out of the platform.

### **3. JOB SEEKER**

### The Job Seeker Module provides candidates with tools to search for jobs, manage applications, and update their profiles.

### **Job Seeker Registration**: New users can register on the platform.

### **Job Seeker Login**: Secure login to access personal dashboards and saved applications.

### **Job Browsing & Application**: Search and filter job listings, view details, and submit applications online.

### **Profile Management**: Manage personal details, resumes, and application history.

### **Query Management:** Raise queries or support requests to the admin.

### **Logout**: Securely log out of the platform.

### **1.3 PROBLEM DEFINITION**

### Despite their continued relevance, many existing **job-seeking and recruitment platforms** face significant challenges in managing growing user demands and staying aligned with modern technological trends. Traditional recruitment systems often lack the flexibility, automation, and efficiency required to keep pace with the expectations of today’s job seekers and employers.

### **• Inefficient Process Management:** Manual posting, tracking, and management of job listings and applications can lead to delays and errors, making it difficult for recruiters to fill positions promptly and for job seekers to find suitable opportunities efficiently.

### **• Limited User Experience:** Outdated platforms may lack intuitive interfaces, responsive design, and effective search and filter functionalities, which can make it challenging for users to navigate job listings, submit applications, or manage their profiles.

### **• High Administrative Overhead:** Manual administrative tasks, such as user management, job posting approvals, and application tracking, consume significant time and resources for admins and recruiters, diverting focus from strategic hiring activities.

### **• Data Security Concerns:** With large volumes of personal and corporate information being shared online, the need to protect sensitive user data has become critical. Legacy systems often lack robust security frameworks, increasing the risk of unauthorized access, data breaches, or misuse of confidential information.

### **1.4 EXISTING SYSTEM:**

### The current state of job-seeking and recruitment systems often relies on a mix of traditional manual processes and outdated job portal software. While these systems have served employers and job seekers for years, they frequently fall short of meeting the expectations of today’s users and the demands of modern technology. Key aspects of the existing system include:

### **• Data Security Risks:** As recruitment moves increasingly online, sensitive data such as personal profiles, resumes, and company information are often exposed to greater risks of security breaches and unauthorized access. Many existing platforms lack advanced security protocols to fully safeguard this information, leaving both job seekers and employers vulnerable to data leaks, phishing, or misuse of personal details.

### **• Limited Scalability:** Legacy recruitment systems often lack the scalability and flexibility needed to adapt to growing user bases and new technological trends. They may struggle to support large volumes of job postings, candidate applications, or advanced features like real-time updates, AI-based recommendations, or seamless integrations with modern tools like React frontends and Firebase backends.

### **1.5 PROPOSED SYSTEM:**

The proposed **Quick Hired** platform aims to revolutionize the way job-seeking and recruitment services operate by leveraging modern web technologies to automate processes, enhance user experience, and improve overall efficiency. Key components of the proposed system include:

• **Integrated Online Platform:**

The system will feature a comprehensive online platform accessible to **admins**, **recruiters**, and **job seekers**. This platform will serve as a centralized hub for all job-related activities, including profile creation, job postings, application tracking, and administrative management.

• **Automated Job Cataloguing and Management:**  
Advanced algorithms will automate the process of posting and managing job listings, reducing manual work for recruiters and ensuring that job postings are accurate, up-to-date, and well-organized. The system will also provide tools to track applications and manage candidate pipelines efficiently.

• **Enhanced User Experience:**  
The platform will deliver an intuitive, responsive interface built with **React**, robust search and filter options for jobs and candidates, and personalized dashboards for admins, recruiters, and job seekers. This will help users quickly find relevant opportunities or candidates, improving engagement and accessibility.

• **Robust Security Measures:**  
Comprehensive security protocols, supported by **Firebase Authentication** and secure database rules, will safeguard sensitive information such as user credentials, resumes, company data, and job postings, protecting the platform from unauthorized access and potential cyber threats.

• **Expert Career Support Page:**  
The system will include a dedicated page where job seekers can directly connect with career coaches or industry experts for personalized advice, resume reviews, and interview preparation tips. This feature may operate on a paid model, providing an additional revenue stream while significantly boosting the value provided to job seekers.

**CHAPTER-2.4 Hardware & Software Requirements**

**HARDWARE REQUIREMENTS**

● Processor: Intel Core i3 or higher (e.g., Intel Core i5, Intel Core i7)

● Ram: 8 GB

● SSD:256GB

**SOFTWARE REQUIREMENTS**

● Front End: HTML, CSS, Bootstrap, JavaScript, ECMA Script, React JS

● DB Tool: Firebase Fire store

● Browser: Mozilla Firefox/Chrome/Edge or any other relevant browser

● OS: Windows operating system/Linux

● Text Editor: Visual Studio

## **CHAPTER-5 Observations, Self-Assessment Report, Results, and Conclusions**

**5.1 Observations**

Throughout the industrial training and development of **QuickHired**, several key observations were made:

**• Engagement and Participation:**

* Regular tasks such as designing, coding, and testing modules kept me actively engaged with the system’s development.
* Collaborative discussions with team members enhanced teamwork, planning, and coordination skills.

**• Application of Concepts:**

* Practical implementation of **React** for the frontend and **Firebase** for authentication, real-time database, and hosting provided valuable hands-on experience.
* Working on real-world scenarios, like secure login, recruiter dashboards, and user profiles, deepened understanding of how modern job portals function.

**• Feedback and Improvement:**

* Code reviews and feedback sessions helped identify performance bottlenecks and security gaps.
* Iterative development using Agile practices allowed for continuous refinement and better integration of user feedback.

**5.2 Self-Assessment Report on Contribution**

**Personal Contributions:**

**• Module Development:**

* Designed and implemented responsive user interfaces using **React**, ensuring an intuitive and engaging user experience.
* Integrated **Firebase Authentication** for secure user and recruiter login/registration workflows.
* Configured **Firebase Firestore** to store and manage user profiles, job postings, and applications.

**• Team Collaboration:**

* Actively participated in project planning, feature breakdown, and assigning tasks using version control tools.
* Assisted peers in solving React component issues, API integrations, and Firebase security rules.
* Contributed reusable components and optimized state management for better performance.

**• Skill Development:**

* Enhanced frontend development skills using **React hooks**, **context API**, and **component lifecycle** techniques.
* Improved backend integration and learned to manage real-time data with **Firebase Realtime Database** and **Firestore**.
* Strengthened problem-solving and debugging skills, especially in resolving async data handling and deployment challenges.

**Areas for Improvement:**

* **Time Management:**
  + Need to better allocate time for testing and deployment stages to prevent last-minute fixes.
  + Break complex modules into smaller deliverables with clear deadlines.
* **Advanced Topics:**
  + Explore **React advanced patterns**, such as Redux for state management and server-side rendering (Next.js).
  + Gain deeper knowledge of Firebase security rules, performance tuning, and cloud functions.

**5.3 Results**

**• Deliverables:**

* Successfully implemented a fully functional **job-seeking platform** with user registration/login, recruiter job posting, application tracking, and admin management.
* The frontend is responsive and dynamic, built entirely with **React**, and seamlessly connected to **Firebase** services.

**• Quality and Feedback:**

* Consistently received positive feedback for clean code, user-friendly UI, and robust authentication workflows.
* Demonstrated steady improvement by incorporating new React features and Firebase best practices as the project progressed.

**5.4 Conclusions**

The development of **QuickHired** has been an invaluable learning experience, providing real-world exposure to modern web technologies and cloud-based backend solutions.

**Key Takeaways:**

* **Technical Proficiency:**
  + Acquired practical expertise in building single-page applications with **React** and leveraging **Firebase** for authentication, database, and hosting.
  + Gained confidence in integrating frontend and backend securely and efficiently.
* **Practical Skills:**
  + Improved debugging, performance optimization, and UI/UX design abilities.
  + Strengthened teamwork, collaboration, and project planning skills.
* **Future Directions:**
  + Plan to deepen knowledge in advanced **React** ecosystems (Next.js, Redux, testing frameworks).
  + Learn more about cloud deployment strategies and Firebase extensions like Cloud Functions for scalability.
  + Aim to expand **QuickHired** with more advanced features such as AI-powered job recommendations and analytics dashboards.

Overall, this project has been a major step forward in my journey as a **full-stack React developer**, equipping me with the skills and confidence needed to tackle real-world **job portal** development challenges.

## **CHAPTER- 3.1 Data Flow Diagram 101: Covering the Basics**

In nutshell, [data flow diagrams](https://www.edrawmax.com/data-flow-diagram/) simply provide a visual representation of how data is handled in a system. The diagrams are commonly used in software engineering and analysis as they can provide a multi-level representation of data and its overall processing.

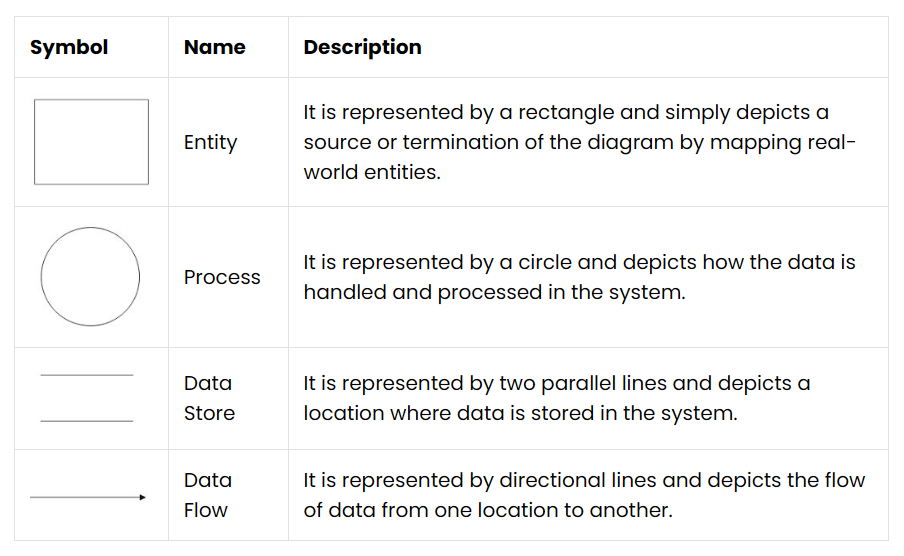
In most cases, they provide the logical representation and processing of data by different components of a system. Apart from its processing, details about its storage, input, and output are also included in DFDs. Therefore, data flow diagrams can be used in the following way:

* To provide the logical flow of information in a system.
* To depict details about the physical connectivity and streaming of data.
* It is also used to represent the network connectivity of data in a system.
* It helps us understand the scalability of the system so that we can improve its functionality.
* Other benefits include a stepwise refinement of the system and learning the behaviour of various components.

## **Common Flow Diagram Symbols and Meanings**

Now when you know the importance of data flow diagrams, let’s get to know how to create them. One of the good things about DFDs is that they are extremely easy to make and only have four major components.

When it comes to**data flow diagram symbols**and meaning, there can be different models such as Yourdon & Marco, Gane & Sarson, SSADM, and so on. Out of them, the Yourdon and Marco model is the most popular approach that is universally followed to create data flow diagrams.



### **2.1 Entity**

Also known as eternal entities, they are often placed at the source or the terminal ends of the diagram. They provide a real-world representation of things in the diagram and are responsible for sending or receiving data. For instance, in the above data flow diagram, customers and warehouses are entities.

### **2.2 Process**

It is one of the most important data flow diagram symbols that depict the processing of data at any stage. A process would have incoming data, a management operation, and an output. It usually provides logical management or change in data. In the diagram, you can see that payment collection or order placement are some of the vital processes.

### **2.3 Data Store**

It simply depicts any location where data is stored in the system. It can be a document or an entire database that is used as a repository to store and handle data. In DFDs, Data Stores can have an input to collect data and an optional output to provide the stored data to any other entity or process. In our diagram, customers, invoices, and orders are some of its data storage units.

### **2.4 Data Flow**

Lastly, any data flow diagram would be incomplete without representing the flow of data from one entity/process to another. As you can see in the diagram, we use directional lines to depict how the data is moving from one place to another. At times, we also add captions on data flow arrows to make others understand these data flow diagram symbols and their meanings.

### **Level 0: Context Diagram**

This level shows the overall system as a single process with external entities.

### **Entities**

### **Job Seeker**

### **Recruiter**

### **Admin**

### **✅ System**

### **• Quick Hired System**

### **✅ Data Flows**

### **• Job Seeker ↔ Quick Hired System: Job Seeker Data, Job Applications, Login Information**

### **• Recruiter ↔ Quick Hired System: Recruiter Data, Job Postings, Application Reviews**

### **• Admin ↔ Quick Hired System: Admin Data, System Management, Reports, Query Resolution**

### **Level 1: DFD**

This level breaks down the system into subprocesses.

#### **Processes:**

#### **1. Job Seeker Management**

#### Register Job Seeker

#### Login Job Seeker

#### View Job Listings

#### Apply for Jobs

#### Raise Query

#### **2. Recruiter Management**

#### Register Recruiter

#### Login Recruiter

#### Post Job Openings

#### Manage Job Posts

#### View Applications

#### Raise Query

#### **3. Admin Management**

#### Login Admin

#### Manage Job Seekers

#### Manage Recruiters

#### Manage Job Postings (Approve/Remove)

#### Resolve Queries

#### Your diagram structure matches well:

#### **Users → Job Seekers**

#### **Tutors → Recruiters**

#### **Courses → Jobs**

#### **Applications → Applications**

#### **Queries → Queries**

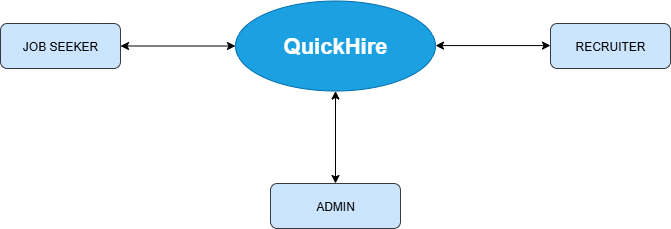
#### **Data Stores:**

1. **Job Seeker Data Store**
2. **Recruiter Data Store**
3. **Job Postings Data Store**
4. **Applications Data Store**
5. **Queries Data Store**

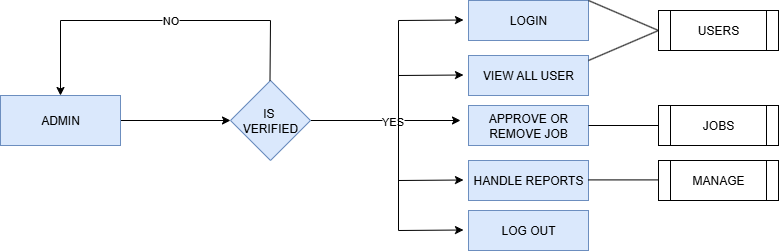
**✅ Data Flows for QuickHired**

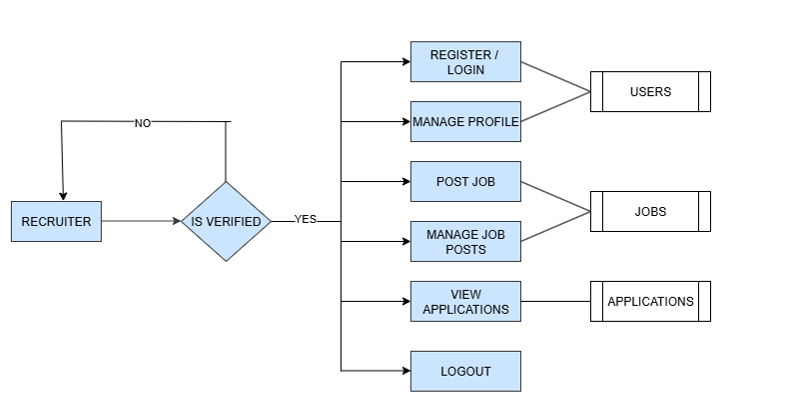
• **Job Seeker Data Store** ↔ **Job Seeker Management**: Job Seeker Info  
• **Recruiter Data Store** ↔ **Recruiter Management**: Recruiter Info  
• **Job Postings Data Store** ↔ **Job Seeker Management**: Job Info  
• **Job Postings Data Store** ↔ **Recruiter Management**: Job Postings  
• **Applications Data Store** ↔ **Job Seeker Management**: Job Applications  
• **Applications Data Store** ↔ **Recruiter Management**: Job Applications  
• **Queries Data Store** ↔ **Job Seeker Management**: Query  
•**Queries Data Store** ↔ **Recruiter Management**: Query  
• **Queries Data Store** ↔ **Admin Management**: Query Resolution

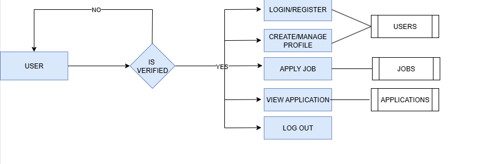
**Level 0 Diagram**



**Level 1 Diagram**







**CHAPTER-3.2 Quick Hire Data Sheet**

**Project Overview:**

* **Project Name:** Quick Hire
* **Modules:** User Panel, Recruiter panel, Admin Panel
* **Technologies:** React,, Bootstrap, HTML, CSS, JS

**User Panel**

**User Registration**

**Functionalities:**

* View Courses: Access a list of available courses.
* Enroll in Course: Register for a course.
* Take Assessment: Participate in assessments related to the courses.
* Provide Feedback: Submit feedback for the courses.
* Raise Query: Submit queries related to courses.

**Data Flow:**

1. **User Registration:**
   * **Input:** Name, Email, Password, Qualification, Experience, Contact, Address
   * **Output:** Job Seeker Profile Created
2. **User Login:**
   * **Input:** Email, Password
   * **Output:** User Authenticated
3. **View Courses:**
   * **Input:** User Request
   * **Output:** List of Available Jobs
4. **Apply for Job:**
   * **Input:** User ID, Job ID, Resume/Details
   * **Output:** Application Submitted
5. **Raise Query:**
   * **Input:** User ID, Job ID, Query
   * **Output:** Query Stored/Forwarded to Employer or Support Team

**Recruiter Panel**

**Data Flow:**

1. **Recruiter Registration:**
   * **Input:** Name, Email, Password, Company Name, Position, Experience, Contact, Address
   * **Output:** Recruiter Profile Created
2. **Recruiter Login:**
   * **Input:** Email, Password
   * **Output:** Recruiter Authenticated
3. **Create Job Posting:**
   * **Input:** Recruiter ID, Job Details (Title, Description, Requirements, Salary, Location)
   * **Output:** Job Posting Created
4. **Manage Job Postings:**
   * **Input:** Recruiter ID, Updated Job Details
   * **Output:** Job Posting Updated or Removed
5. **Review Applications:**
   * **Input:** Recruiter ID, Job ID
   * **Output:** Applications Reviewed and Shortlisted
6. **Raise Query:**
   * **Input:** Recruiter ID, Query
   * **Output:** Query Stored/Forwarded to Admin

**Functionalities:**  
• **Create Job Posting:** Post new job openings with detailed information.  
• **Manage Job Postings:** Edit, update, or remove job listings as needed.  
• **Review Applications:** Access and review applications submitted by job seekers.  
• **Raise Query:** Submit queries to the admin or support team for assistance.

**Admin Panel**

**Admin Registration**  
**Fields:**

* Name
* Email
* Password

**Admin Login**  
**Fields:**

* Email
* Password

**Functionalities:**  
• **Manage Users:** Add, update, or delete job seeker profiles.  
• **Manage Recruiters:** Add, update, or delete recruiter profiles.  
• **Manage Job Postings:** Approve, update, or remove job postings.  
• **Resolve Queries:** Address and resolve queries raised by job seekers or recruiters.

**Data Flow:**

1. **Admin Registration:**
   * **Input:** Name, Email, Password
   * **Output:** Admin Profile Created
2. **Admin Login:**
   * **Input:** Email, Password
   * **Output:** Admin Authenticated
3. **Manage Users:**
   * **Input:** Admin ID, User Data
   * **Output:** User Data Managed (Add/Update/Delete)
4. **Manage Recruiters:**
   * **Input:** Admin ID, Recruiter Data
   * **Output:** Recruiter Data Managed (Add/Update/Delete)
5. **Manage Job Postings:**
   * **Input:** Admin ID, Job Data
   * **Output:** Job Posting Status Updated (Approved/Removed)
6. **Resolve Queries:**
   * **Input:** Admin ID, Query ID
   * **Output:** Query Resolved

**SCREENSHOTS**

**CONCLUSION**

In conclusion, the development of “Quick Hire” represents a significant step forward in modernizing job search services and adapting to the demands of the digital age. Through the implementation of advanced technologies and innovative solutions, this project aims to address key challenges faced by job seekers and employers in efficiently posting, finding, and managing job opportunities.

By providing a user-friendly interface, streamlined administrative processes, and robust security measures, the proposed system not only improves operational efficiency but also fosters a culture of professional growth and community engagement. The transition from traditional, manual job-hunting methods to a modern, digital platform enables job seekers and employers to better meet their goals and expectations while staying competitive in an ever-evolving employment landscape.

**BIBLIOGRAPHY**

**Websites:**

* [Google](https://www.google.com)
* [Stack Overflow](https://www.stackoverflow.com)