

A Major Project Synopsis on

Job Portal

Submitted to Manipal University,

Jaipur

Towards the partial fulfillment for the Award of the

Degree of

MASTER OF COMPUTER APPLICATIONS

2023-2025

by

Pawani Awasthi

23FS20MCA00004



**MANIPAL UNIVERSITY
JAIPUR**

Under the guidance of

Dr. Avichandra Singh

Department of Computer Applications

School of AIML, IoT&IS, CCE, DS and Computer Applications

Faculty of Science, Technology and Architecture

Manipal University Jaipur

Jaipur, Rajasthan

2025

External Internship

Company Name: WENDOR

Role: Frontend Developer Intern

Location: Gurugram, Haryana, India

Company Website: [Wendor](https://wendor.com)

About the Company

WENDOR is a technology-driven company specializing in smart vending and automated retail solutions. By integrating cutting-edge IoT technology with seamless payment systems, Wendor enables businesses to offer cashless, contactless, and highly efficient vending experiences. Their solutions include AI-powered inventory management, real-time sales tracking, and personalized customer engagement through data-driven insights. Trusted by numerous brands and enterprises, Wendor is revolutionizing the automated retail industry by enhancing convenience, optimizing operations, and driving higher sales through intelligent vending ecosystems.

I. Introduction

The project '**Job Portal**' is a full-stack web application designed to enhance job searching and recruitment efficiency. It caters to two primary user personas: **Employers** and **Job Seekers**. The platform includes dedicated dashboards for both user types - Employers can post new job openings and track the status of previously posted jobs, while Job Seekers can monitor the status of their job applications. To ensure secure user authentication for both Employers and Job Seekers, the system implements **JWT** (JSON Web Tokens) and **bcrypt**, a JavaScript package used to hash passwords. Instead of storing passwords in plain text, bcrypt converts them into cipher text, thereby enhancing security. For managing resumes uploaded by job seekers, the system integrates **Cloudinary**, a cloud-based image and video management service. Additionally, **Socket.io** is used to send real-time notifications, these technologies create a seamless, efficient, and secure job portal experience.

II. Motivation

1. Simplifies the hiring process by providing a structured platform for both Employers and Job Seekers.
2. Offers intuitive dashboards for Employers to manage job postings and for Job Seekers to track their applications efficiently.
3. Implements JWT authentication and bcrypt encryption to protect user credentials and sensitive data.
4. Uses Cloudinary for secure and scalable storage of uploaded resumes, ensuring smooth document handling.
5. Connects job seekers with potential employers, making the hiring process more accessible and effective.

III. Problem Statement

- **Key challenges include:**
 1. Ensuring secure authentication and password protection for Employers and Job Seekers while preventing unauthorized access.
 2. Developing a system that allows Employers to post jobs seamlessly and Job Seekers to track their applications in real time.
 3. Implementing a scalable solution for storing resumes and providing timely job updates through notifications and emails.

IV. Methodology/ Planning of Work

1. **Creating the UI of the project using React.js, which will include different components, such as:**
 - a. Home
 - b. Job Listings
 - c. Job Application Form
 - d. Employer Dashboard
 - e. Job Seeker Dashboard
 - f. Profile
2. **Creating different portals:**
 - a. **Admin Portal**
 - i. Manage Users (Employers & Job Seekers)
 - ii. Approve/Reject Job Postings

- iii. Monitor System Performance
- iv. Manage Subscriptions & Premium Features
- v. Contact & Support

b. Employer Portal

- i. Post New Job Listings
- ii. Track Job Applications
- iii. Review Job Seeker Profiles & Resumes
- iv. Schedule Interviews
- v. Send Notifications to Applicants
- vi. Manage Job Posting Status (Active/Closed)

c. Job Seeker Portal

- i. Browse & Search Jobs
- ii. Apply for Jobs
- iii. Upload & Manage Resume (Cloudinary)
- iv. Track Application Status
- v. Receive Real-Time Notifications (Socket.io)
- vi. Receive Email Updates (Nodemailer)

3. Working with backend using:

- a. **Node.js and Express.js** for server-side logic
 - b. **MongoDB** for database management
 - c. **JWT & bcrypt** for secure authentication
 - d. **Cloudinary** for resume storage
 - e. **Socket.io** for real-time job notifications
 - f. **Nodemailer** for email updates
 - g. **Postman** for API testing
-
- **User Authentication & Security:** Implementing JWT (JSON Web Tokens) for secure authentication and bcrypt for encrypting user passwords to ensure data privacy.
 - **Database Management:** Storing job postings, user profiles, and applications securely in MongoDB for efficient data handling.
 - **Resume Storage:** Using Cloudinary to manage and store resumes uploaded by job seekers securely in the cloud.
 - **Job Posting & Application Tracking:** Developing dashboards for Employers to post jobs and track applications, and for Job Seekers to monitor the status of their applications.

- **Real-Time Notifications & Email Updates:** Integrating Socket.io for real-time notifications and Nodemailer for automated email updates regarding job application status.
- **Frontend Development:** Designing a user-friendly and responsive interface using React.js for seamless interaction.
- **Backend Development:** Building a robust server using Node.js and Express.js to handle API requests, authentication, and database operations.
- **Testing & Debugging:** Ensuring the reliability and performance of the system by testing APIs with Postman and debugging errors efficiently.
- **Deployment:** Hosting the application using AWS, for accessibility and scalability.

➤ **Tools & Technologies:**

- **Frontend:** React.js
- **Backend:** Node.js and Express.js
- **Database:** MongoDB
- **Authentication & Security:** JWT, bcrypt
- **Resume Management:** Cloudinary
- **Notifications & Emails:** Socket.io, Nodemailer
- **API Testing & Management:** Postman API
- **Communication & Collaboration:** Slack, Google Docs, Google Sheets

V. Requirements for Proposed Work

- **Software Requirements**
 1. **Operating System:** Windows, Linux
 2. **User Interface:** React.js
 3. **Database:** MongoDB
 4. **Backend:** Node.js and Express.js
 5. **Authentication & Security:** JWT, bcrypt
 6. **Resume Storage:** Cloudinary
 7. **Notifications & Email Services:** Socket.io, Nodemailer
 8. **API Testing & Management:** Postman
 9. **Collaboration Tools:** Slack, Google Docs, Google Sheets

- **Hardware Requirements**

1. **Processor:** Minimum Intel i5 or equivalent
2. **RAM:** 8GB (minimum)
3. **Storage:** 100 GB Hard Disk Space

VI. Bibliography/References

1. **Bcrypt for Secure Password Hashing** - <https://www.npmjs.com/package/bcrypt>
2. **MongoDB Documentation** - <https://www.mongodb.com/docs/>
3. **React.js Official Documentation** - <https://react.dev/>
4. **Node.js & Express.js Official Guide** - <https://expressjs.com/>