Synopsis
On

"JOB PORTAL SYSTEM"

Submitted to the **Uttaranchal University** in partial fulfilment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Submitted by

SUDHANSHU SHARMA
(Learner ID: 2226010212)

Under the Guidance of

Guide Name with Designation (Faculty Guide)

UTTARANCHAL UNIVERSITY, DEHRADUN

Title

Job Portal System

Introduction and Objectives of the Project

The Job Portal System is a web-based platform designed to bridge the gap between job seekers and recruiters. It provides an intuitive interface for recruiters to post and manage job listings while offering job seekers a user-friendly experience to search and apply for suitable roles. The primary objective of this project is to

create an efficient, secure, and scalable system that facilitates seamless interactions between recruiters and job seekers.

Objectives:

- To develop a platform that simplifies job posting and application processes.
- To ensure a user-friendly experience for both recruiters and job seekers.
- To implement robust security mechanisms to protect user data and system integrity.
- To facilitate dynamic updates and notifications via email systems.

Tools/Platform, Hardware, and Software Requirements

Tools/Platform:

- Frontend: HTML, CSS, JavaScript, EJS
- Backend: ExpressJS with Node.js
- Database: In-memory data structures
- Templating: EJS for dynamic HTML generation

Hardware Requirements:

- Processor: Intel Core i5 or equivalent
- RAM: 8 GB minimum
- Hard Disk: 500 GB minimum
- Network: Broadband internet connection

Software Requirements:

- Operating System: Windows 10/Ubuntu 20.04
- Development Environment: Visual Studio Code
- Browser: Google Chrome, Mozilla Firefox
- Node.js: Version 16 or higher

Problem Definition and Requirement Specifications

Problem Definition: The lack of a centralized, intuitive platform for job seekers and recruiters hampers efficient communication. Existing systems are often overly complex or lack essential features, such as real-time updates, application tracking, and resume management.

Requirement Specifications:

- **Functional Requirements:**
    - Recruiters can register, log in, post, update, and delete job listings.
    - Job seekers can view jobs, apply to positions, and upload resumes.
    - Email notifications for applicants upon successful application submission.

Technical Specifications:

- MVC architecture using ExpressJS.
- Session and cookie-based user tracking.
- Middleware for authentication and file upload.

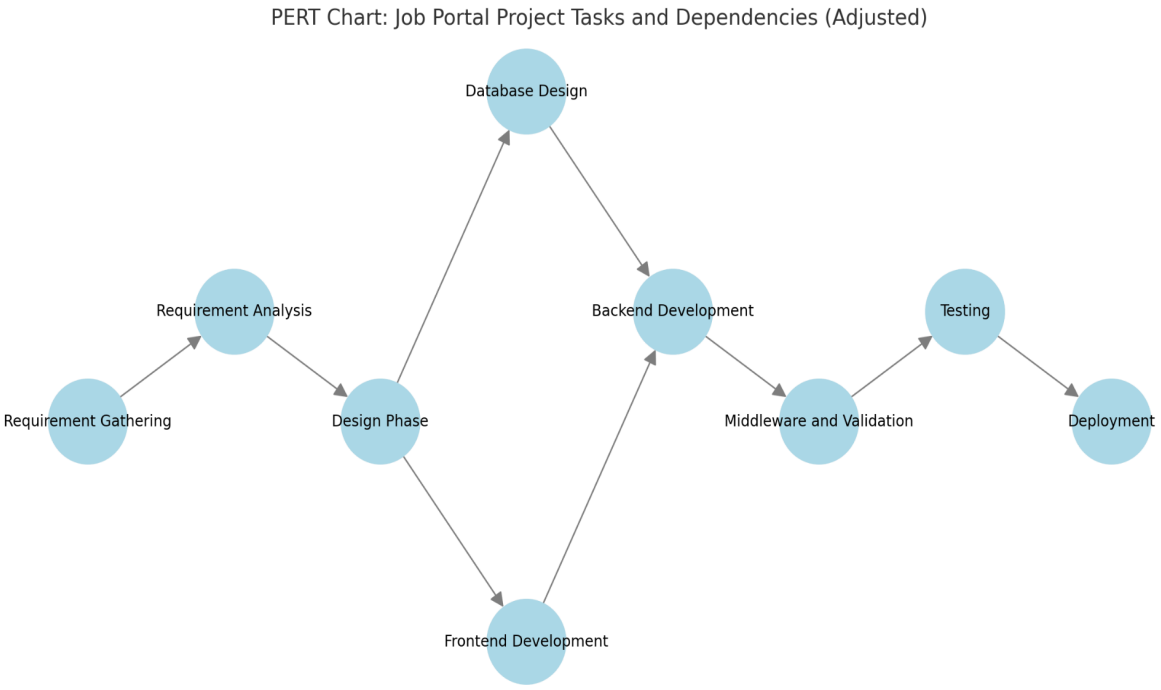
Project Planning and Scheduling:

- **Gantt Chart:** A Gantt chart is a timeline that outlines the project schedule. Each task is represented as a bar showing the start and end dates.

Task	Start Date	End Date	Duration
Requirement Analysis	Oct 1, 2024	Oct 7, 2024	7 Days
Design Phase	Oct 8, 2024	Oct 14, 2024	7 Days
Development	Oct 15, 2024	Nov 15, 2024	1 Month
Testing	Nov 16, 2024	Nov 25, 2024	10 Days
Deployment	Nov 26, 2024	Dec 4, 2024	8 Days

.

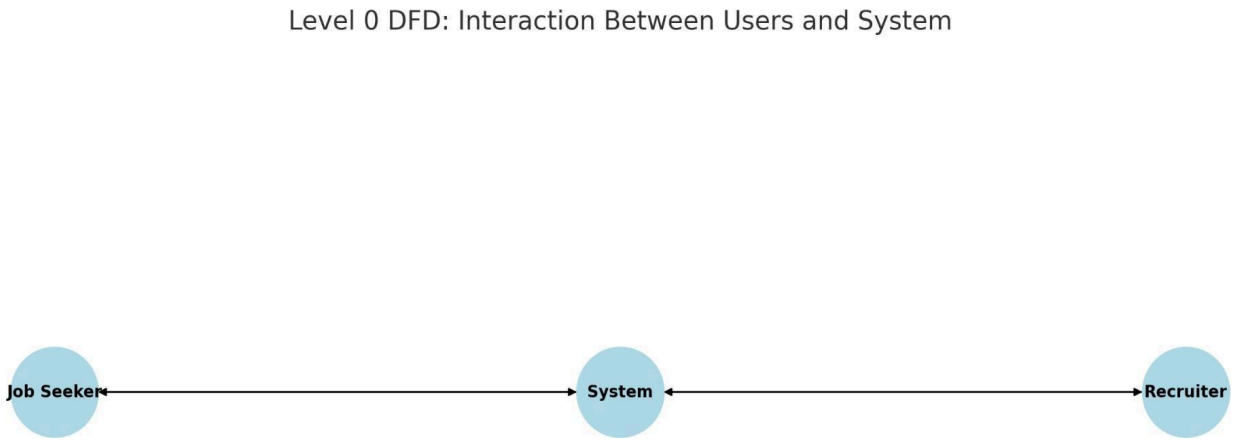
- **PERT Chart:** A PERT (Program Evaluation Review Technique) chart visualizes project tasks, dependencies, and milestones as a flowchart.



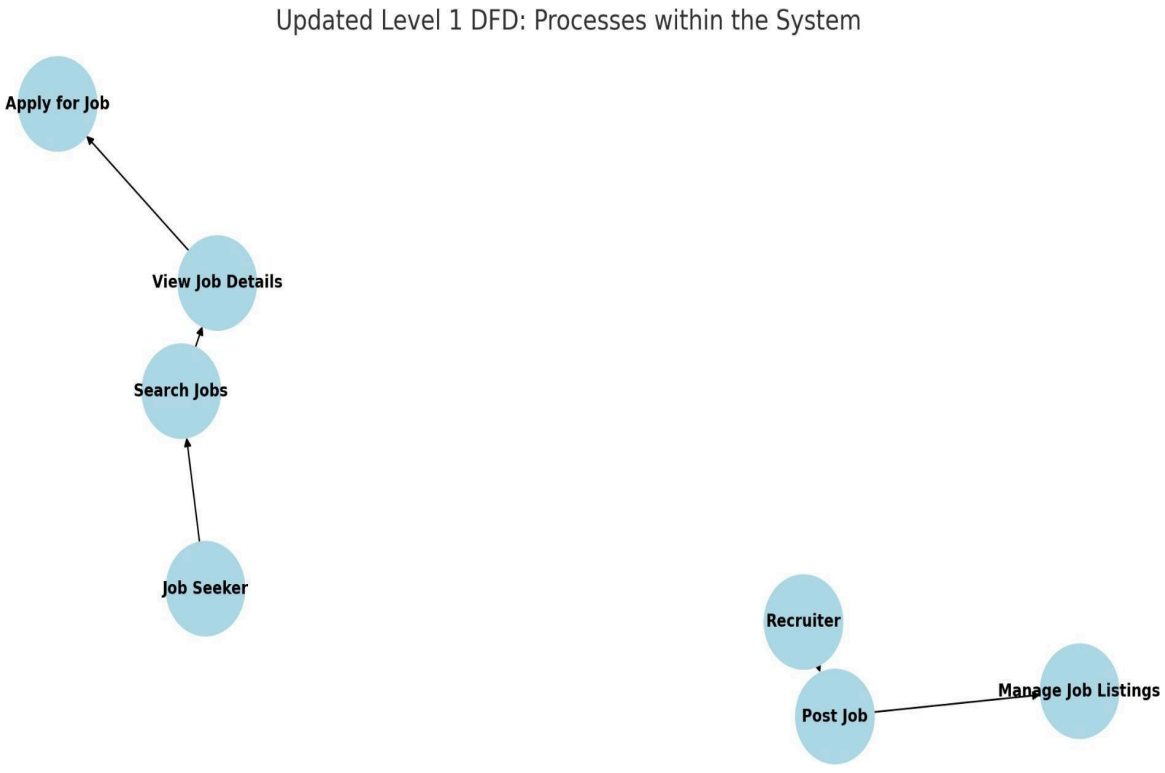
Analysis

Data Flow Diagrams (DFDs):

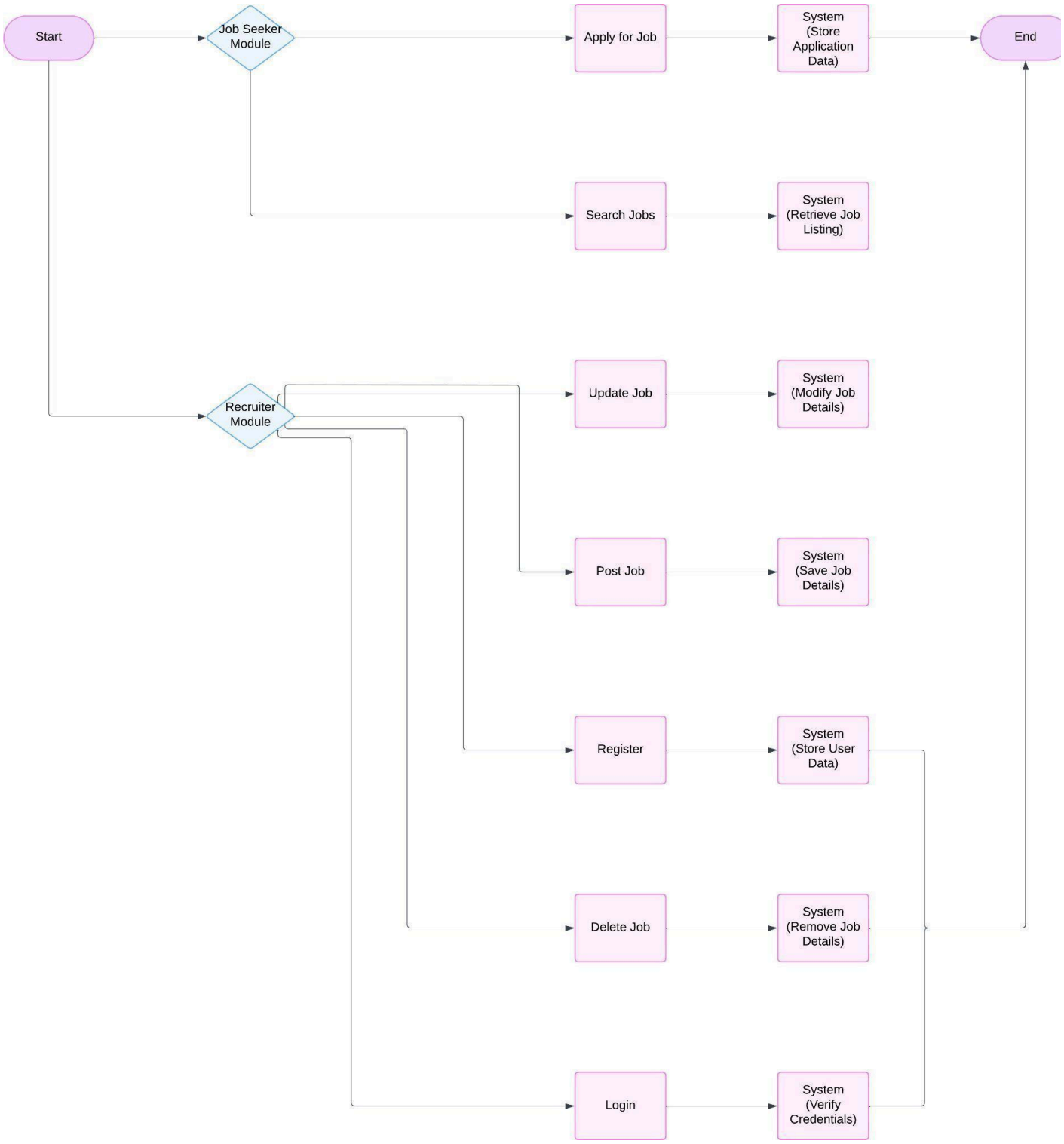
- **Level 0:** High-level interaction between job seekers, recruiters, and the system.



- **Level 1:** Breakdown of modules such as user authentication, job management, and application handling**.**

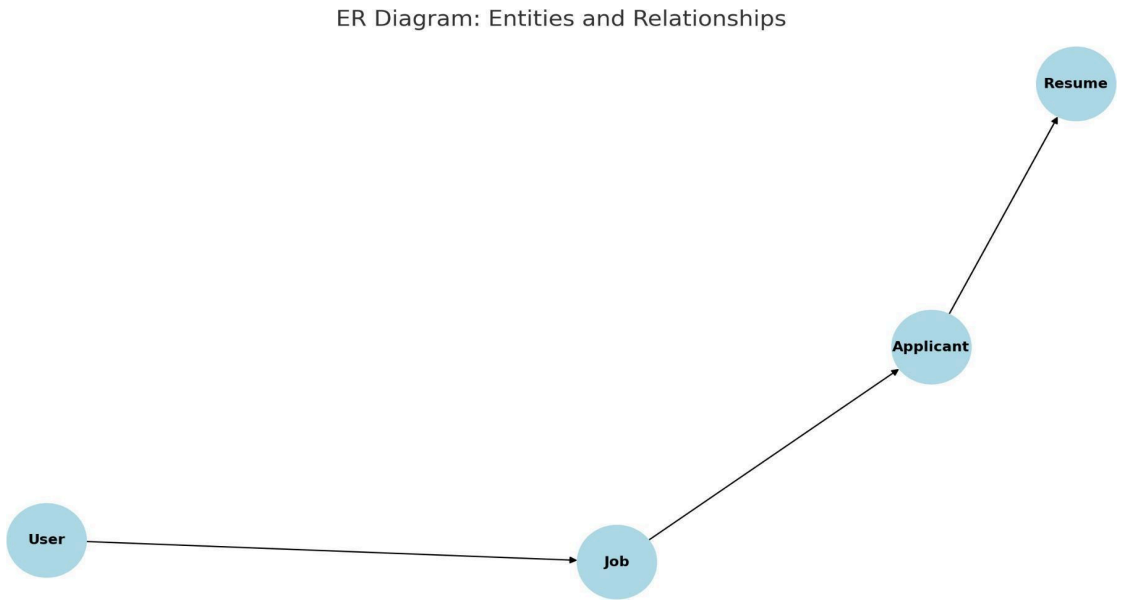


Level 2: Detailed flow of processes like job application submission and recruiter actions



Entity-Relationship (ER) Diagram:

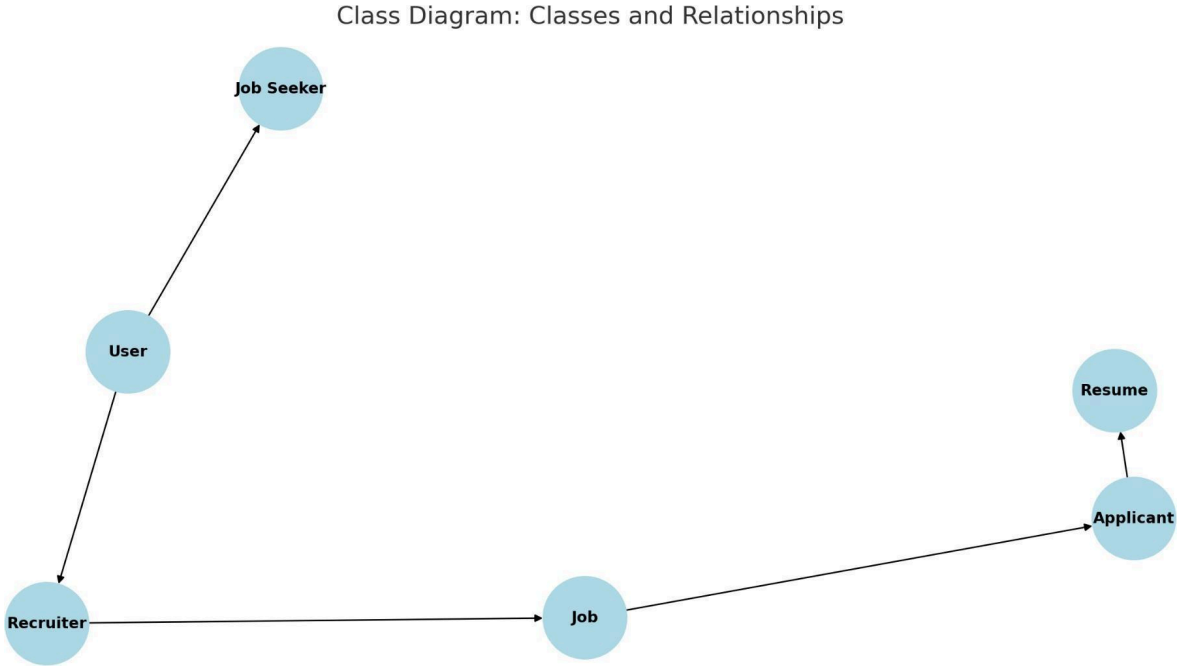
- **Key entities:** User, Job, Applicant
- **Relationships:** One-to-many (Recruiter to Jobs), One-to-many (Job to Applicants)



Class Diagrams:

- **Classes:** User, Job, Applicant, Session, Middleware

Attributes and methods defined as per data structures and functionalityComplete



Complete Structure

Modules and Description:

1. **User Management Module:** Handles login, registration, and session management**.**
2. **Job Management Module:** Manages job postings, updates, deletions, and listings**.**
3. **Application Module:** Enables job seekers to apply and recruiters to view applications.
4. **Notification Module:** Sends email confirmations to applicants.

Process Logic:

- **Recruiter Actions:** Authenticate by Registering /Login ->-> Post Job -> View Applicants -> Update/Delete Job.
- **Job Seeker Actions:** Browse Jobs -> Apply to Job -> Receive Confirmation.

Data Structures:

- **User:** `id`, `name`, `email`, `password`
- **Job:** `id`, `jobcategory`, `jobdesignation`, `joblocation`, `companyname`, `salary`, `applyby`, `skillsrequired`, `numberofopenings`, `jobposted`, `applicants`
- **Applicant:** `applicantid`, `name`, `email`, `contact`, `resumePath`

Reports:

- List of all jobs posted.
- Applicant details for a specific job.
- Activity logs for recruiters.

Proposed Security Mechanisms

- User authentication with hashed passwords.
- Role-based access control for recruiters and job seekers.
- Middleware for secure file uploads and access validation.
- Input validation to prevent SQL injection and XSS attacks.

Future Scope and Further Enhancements

- Integration with external job boards for increased reach.
- Implementation of machine learning algorithms for job recommendations.
- Advanced analytics and reporting features for recruiters.
- Mobile application development for enhanced accessibility.

Bibliography

- Node.js Documentation: https://nodejs.org
- ExpressJS Documentation: https://expressjs.com
- EJS Documentation: https://ejs.co
- Software Engineering Textbook by Ian Sommerville.