AI-POWERED APPLICANT TRACKING SYSTEM (ATS)

INTRODUCTION

PROBLEM STATEMENT:

DEVELOP AN INTELLIGENT SYSTEM THAT AUTOMATES RESUME SCREENING, CALCULATES ATS SCORES, AND STREAMLINES RECRUITER DECISION-MAKING WITH SEAMLESS EMAIL AUTOMATION.

OBJECTIVE:

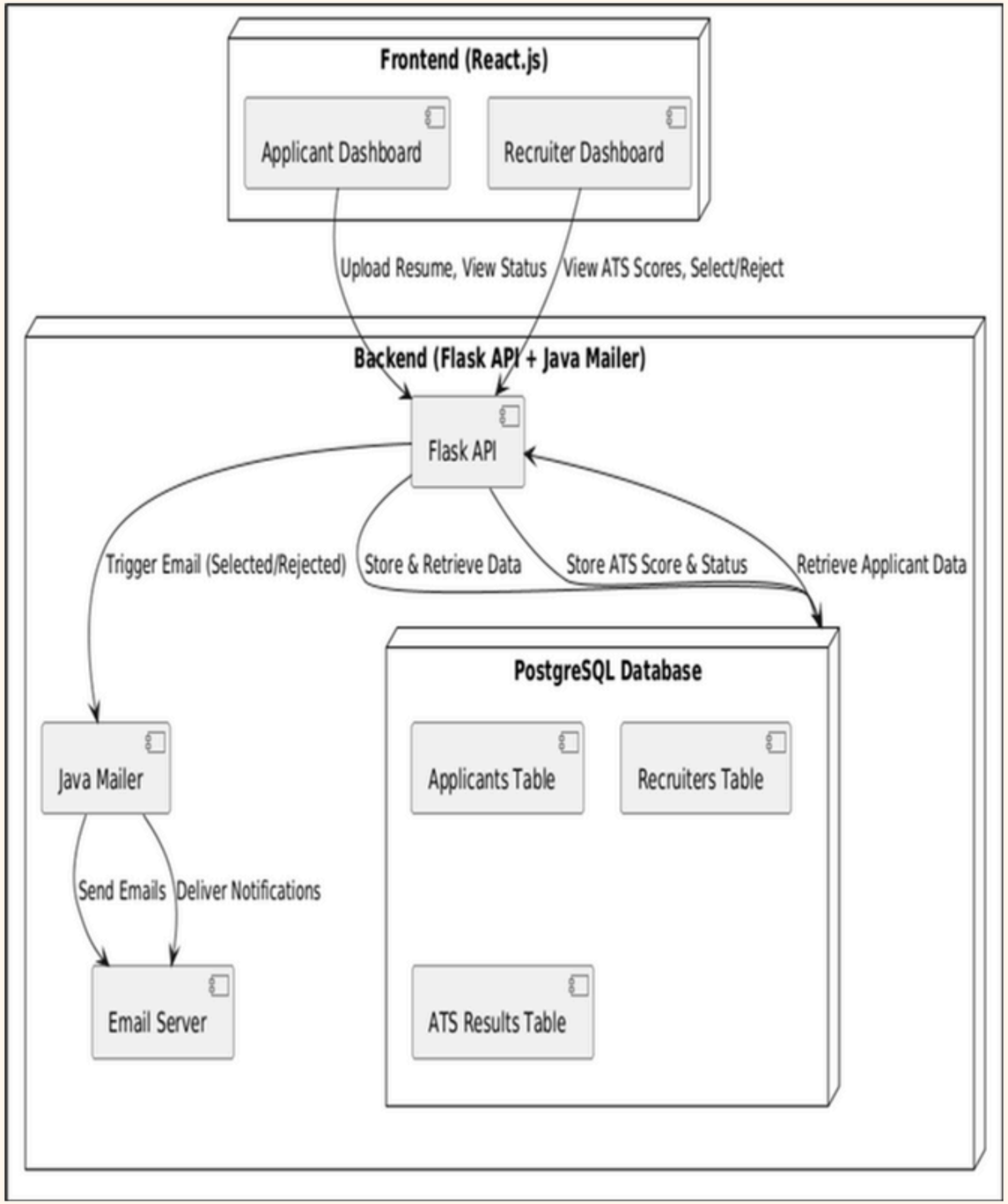
- ✓ SECURE APPLICANT & RECRUITER LOGIN
- ✓ AI-DRIVEN RESUME PARSING & ATS SCORING
- ✓ AUTOMATED EMAIL NOTIFICATIONS FOR CANDIDATE UPDATES
- ✓ FULLY INTEGRATED FRONTEND & BACKEND WORKFLOW

OUTCOME:

A SMART, EFFICIENT, AND AUTOMATED RECRUITMENT SYSTEM
THAT ENHANCES HIRING DECISIONS AND IMPROVES WORKFLOW EFFICIENCY.

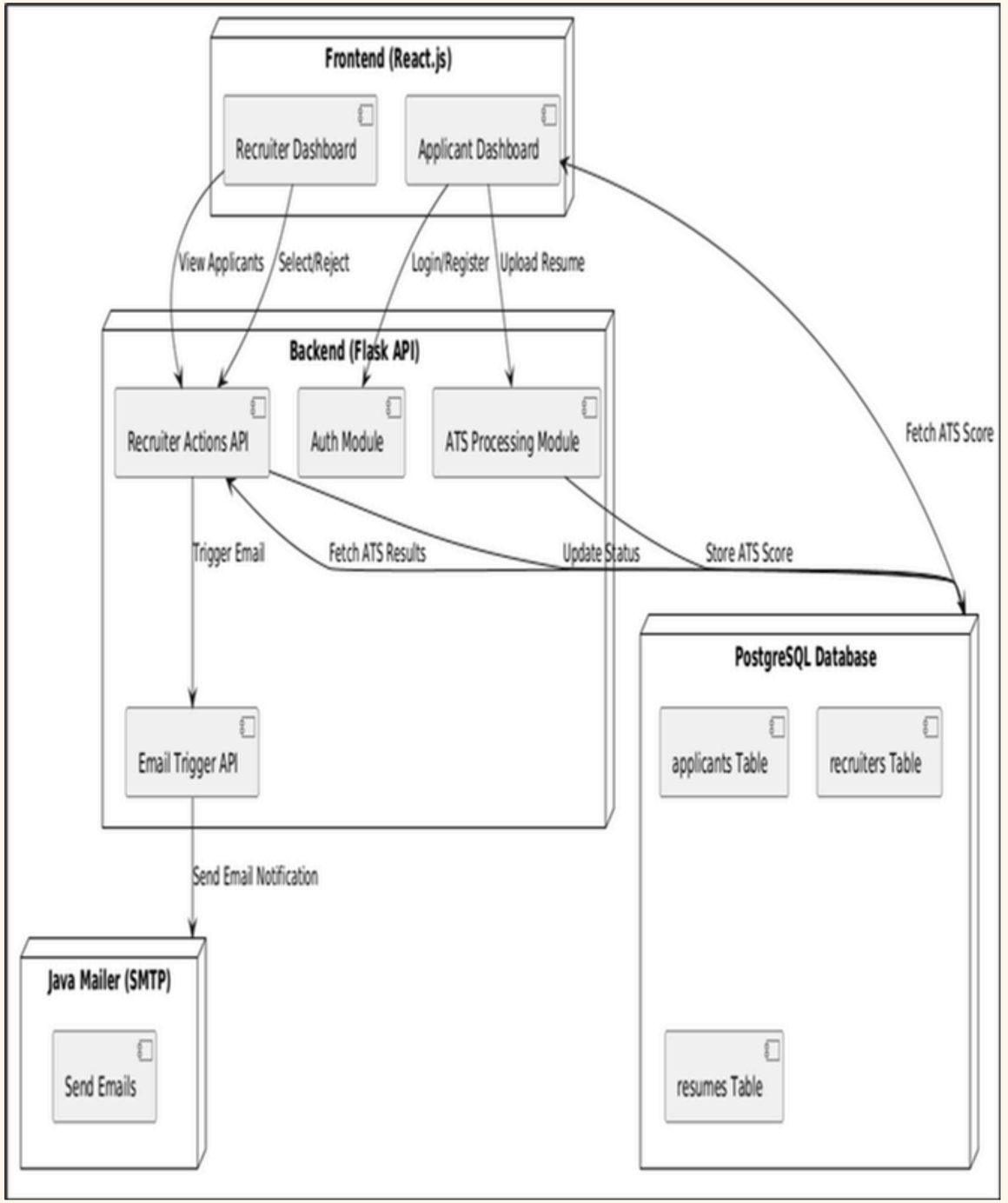
HIGH LEVEL ARCHITECTURE DIAGRAM

HELPS IN PROJECT PLANNING AND STAKEHOLDER COMMUNICATION.



LOW LEVEL ARCHITECTURE DIAGRAM

GUIDES DEVELOPERS WITH PRECISE IMPLEMENTATION DETAILS.



Requirement Engineering Process

INCEPTION

To establish a fundamental understanding of the system and its stakeholders

1. Identify Primary Stakeholders
2. Define High-Level Scope & Goals
3. Perform Feasibility Analysis
4. Risk Identification

ELICITATION

To gather detailed requirements from stakeholders through various methods.

1. Interviews with Stakeholders
2. User Story Mapping
3. Identify Key Challenges

ELABORATION

To refine and structure the gathered requirements into a well-defined workflow.

1. Define the Job Posting Process
2. Define the Job Application Flow
3. Define Recruiter’s Decision-Making Process
4. Ensure Smooth Frontend-Backend Integration

NEGOTIATION

To resolve conflicts between different stakeholders.

1. Recruiters want manual approval vs. Applicants prefer auto-selection.
2. Resume Parsing Challenges (PDF, DOCX, etc.).

SPECIFICATION

To document all functional & non-functional requirements in a structured manner.

1. Software Requirement Specification (SRS)
2. Database Schema
3. API Documentation

VALIDATION

To test if the requirements fulfill stakeholder needs before development.

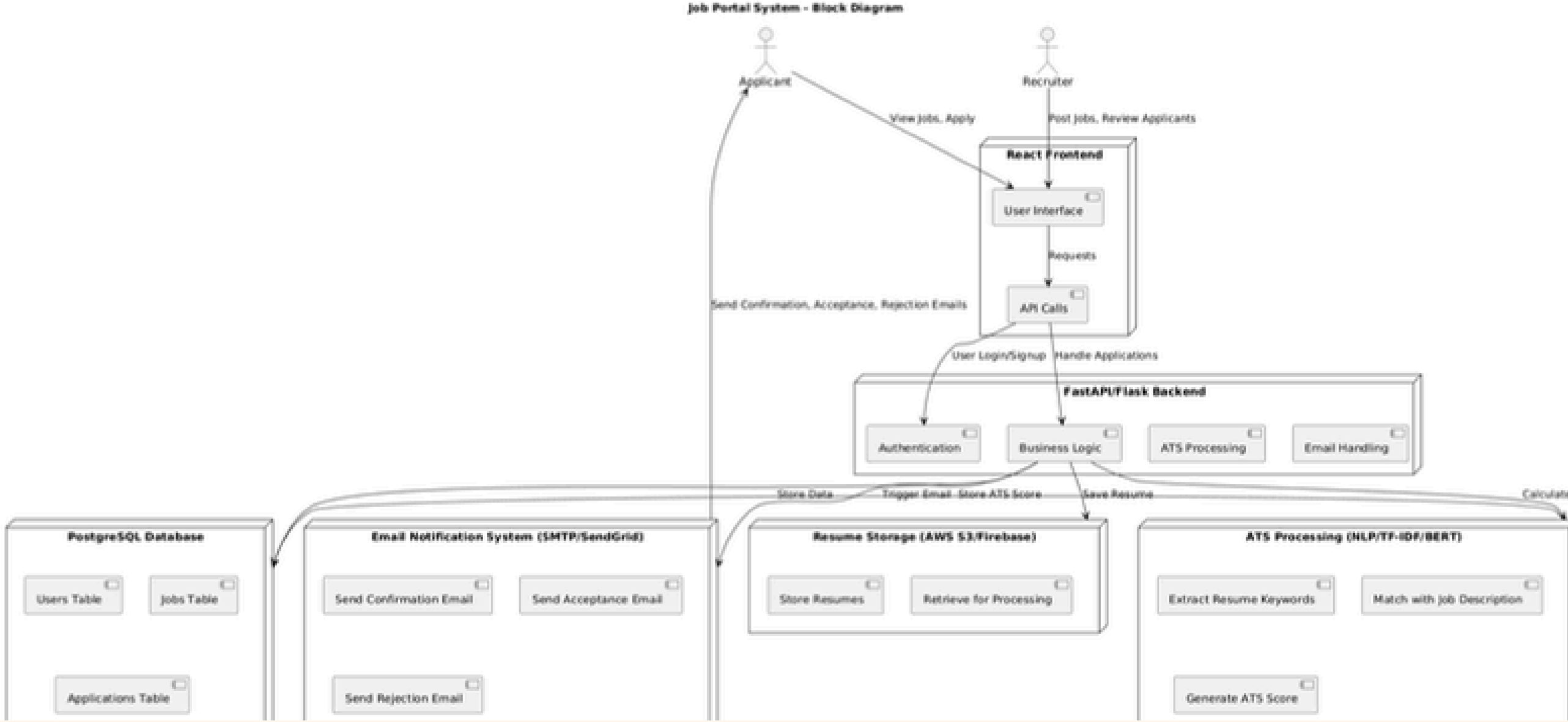
1. Prototype Testing
2. Stakeholder Review
3. System Tests

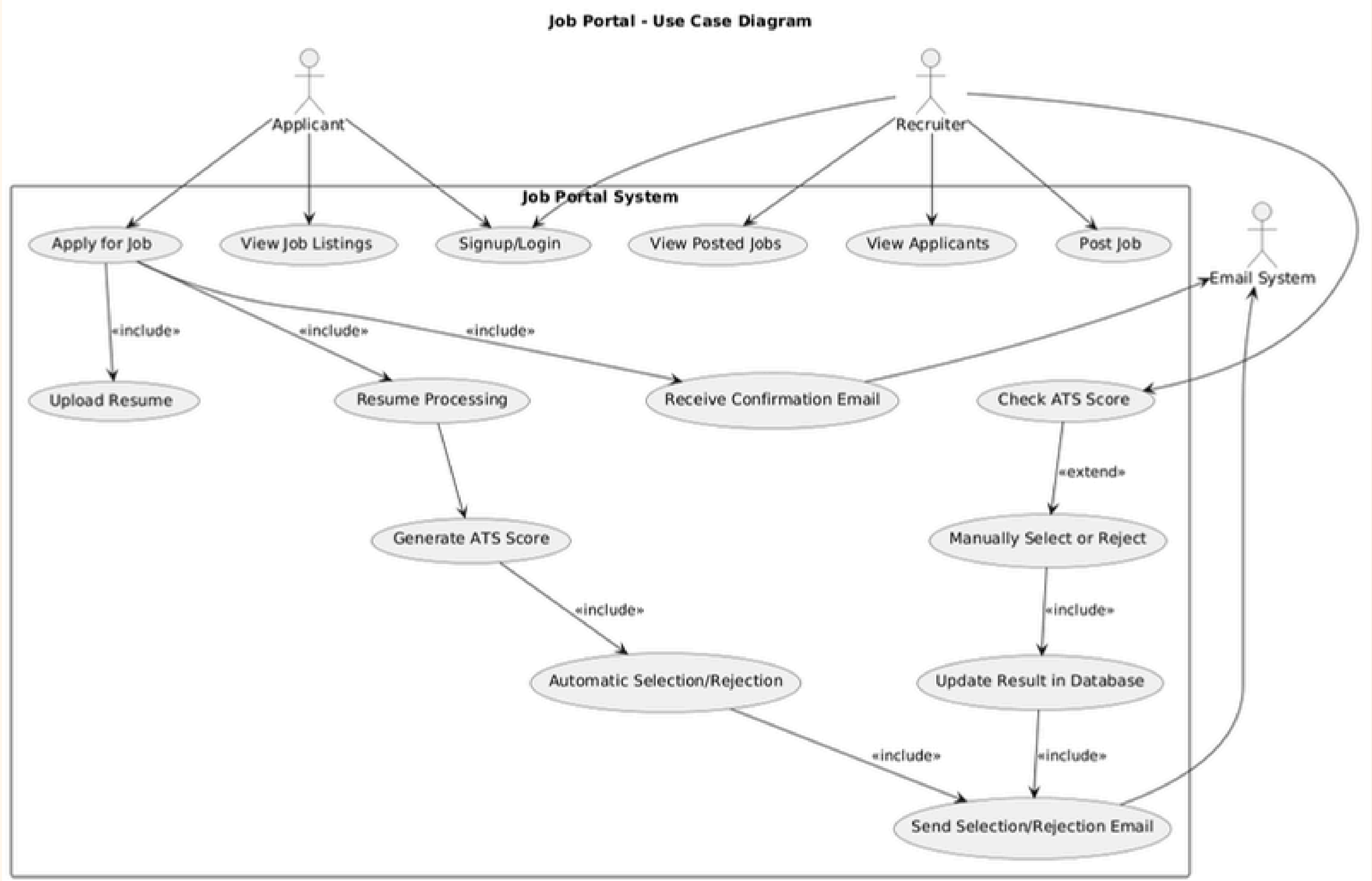
MANAGEMENT

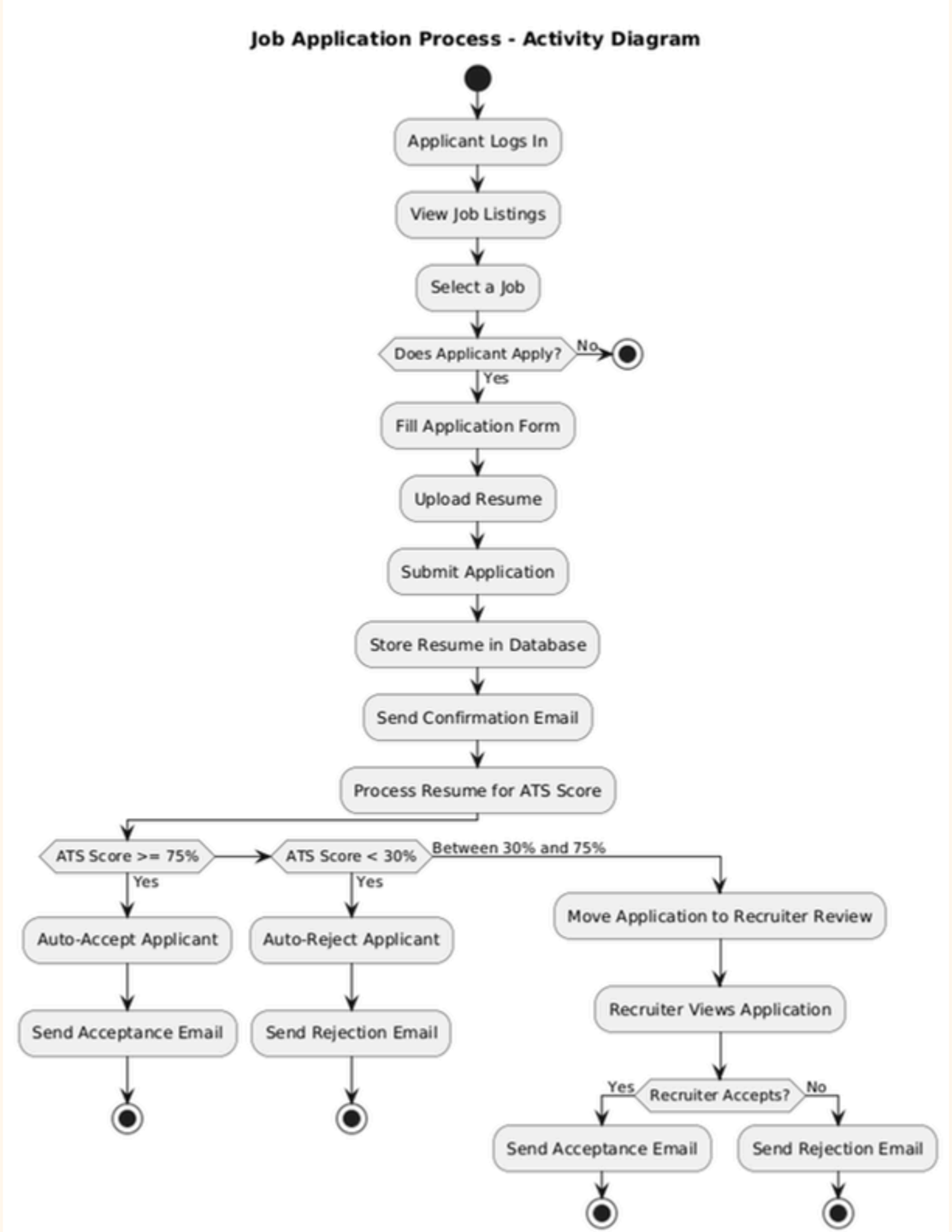
To ensure that job portal requirements are effectively tracked, controlled, updated, and validated throughout the project lifecycle.

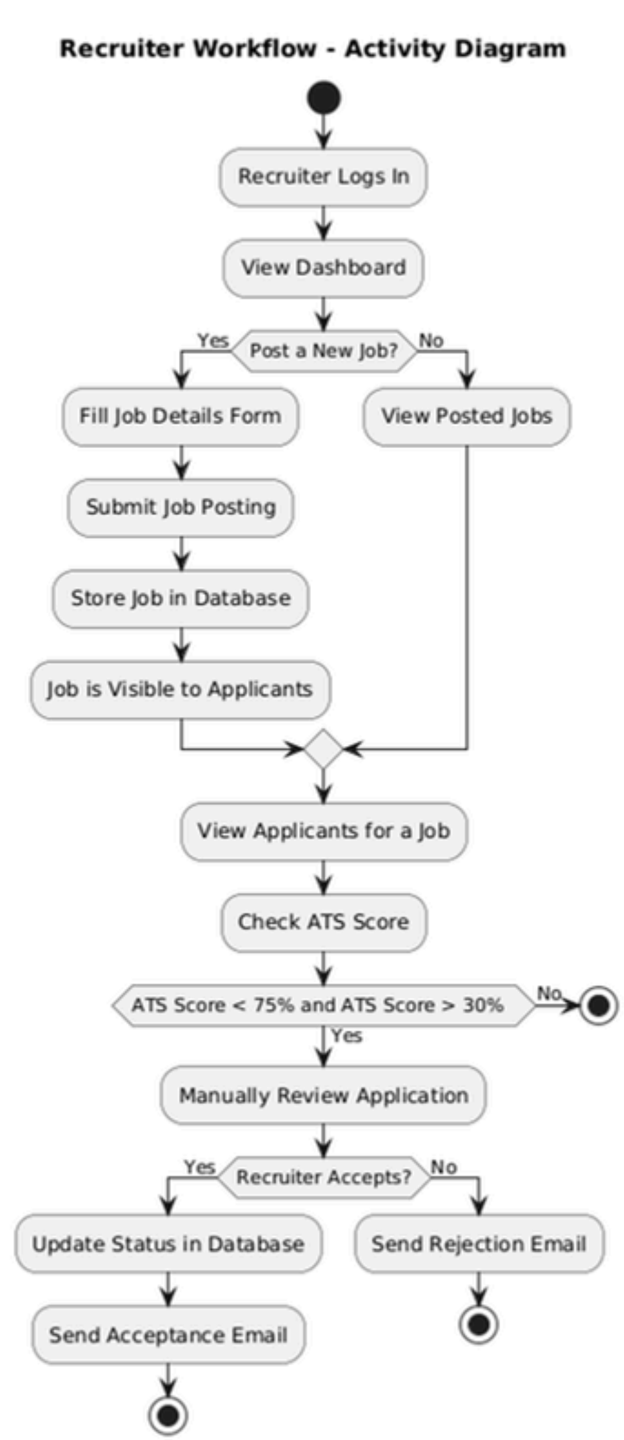
1. Managing multiple requirements across different stakeholders
2. Ensuring changes in requirements do not break the system.

BLOCK DIAGRAM

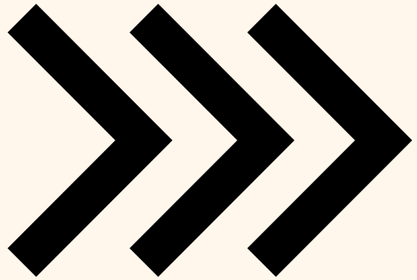



Job Portal - Use Case Diagram
Applicant
Recruiter
Job Portal System
Email System
Apply for Job
View Job Listings
Signup/Login
View Posted Jobs
View Applicants
Post Job
«include»
«include»
«include»
Upload Resume
Resume Processing
Receive Confirmation Email
Check ATS Score
«extend»
Generate ATS Score
Manually Select or Reject
«include»
«include»
Automatic Selection/Rejection
Update Result in Database
«include»
«include»
Send Selection/Rejection Email


Job Application Process - Activity Diagram
Applicant Logs In
View Job Listings
Select a Job
Does Applicant Apply?
No
Yes
Fill Application Form
Upload Resume
Submit Application
Store Resume in Database
Send Confirmation Email
Process Resume for ATS Score
ATS Score >= 75%
ATS Score < 30%
Between 30% and 75%
Yes
Yes
Auto-Accept Applicant
Auto-Reject Applicant
Send Acceptance Email
Send Rejection Email
Move Application to Recruiter Review
Recruiter Views Application
Yes
Recruiter Accepts?
No
Send Acceptance Email
Send Rejection Email


Recruiter Workflow - Activity Diagram
Recruiter Logs In
View Dashboard
Yes
Post a New Job?
No
Fill Job Details Form
View Posted Jobs
Submit Job Posting
Store Job in Database
Job is Visible to Applicants
View Applicants for a Job
Check ATS Score
ATS Score < 75% and ATS Score > 30%
No
Yes
Manually Review Application
Yes
Recruiter Accepts?
No
Update Status in Database
Send Rejection Email
Send Acceptance Email

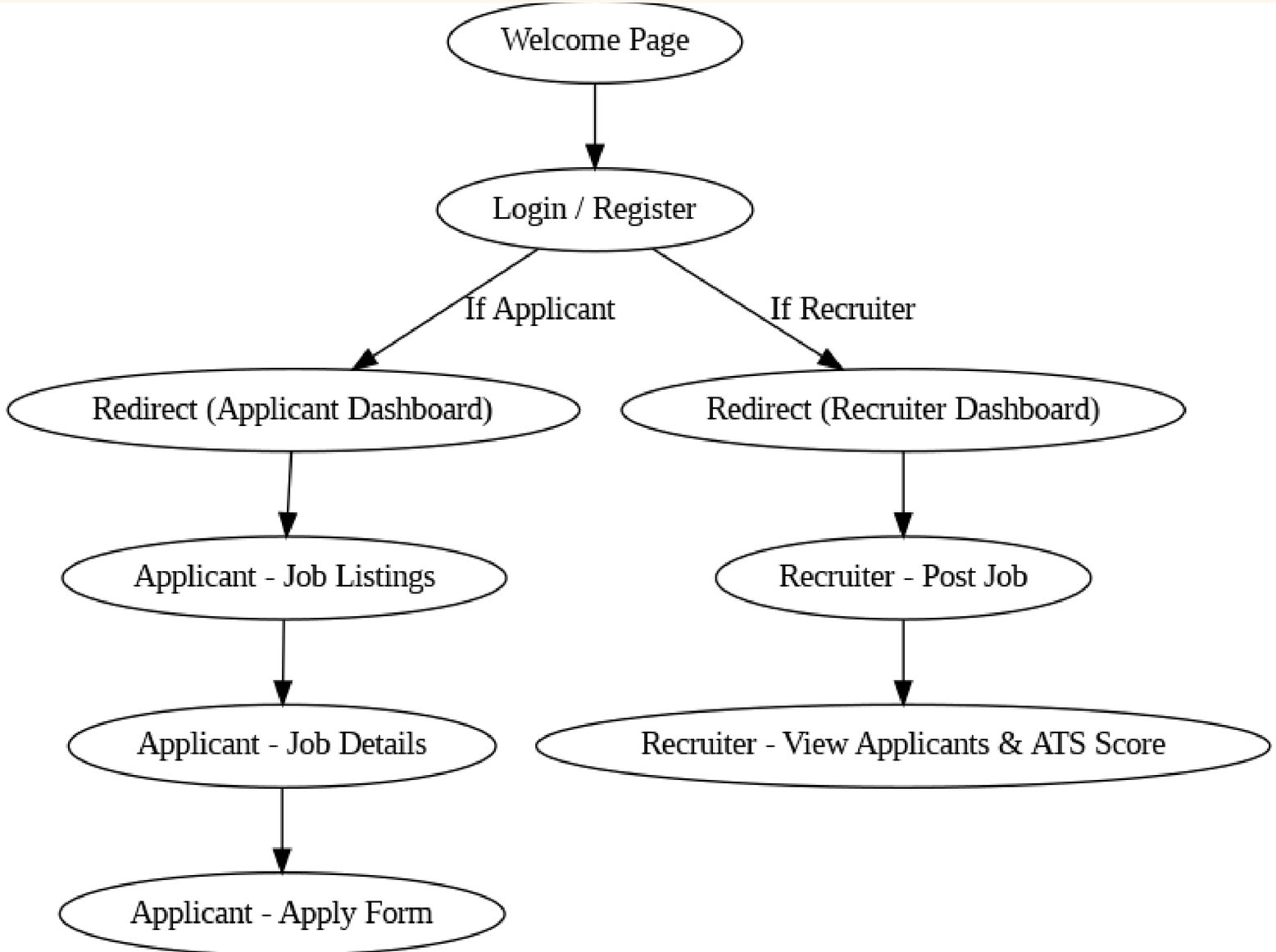
FRONT END



TECHNOLOGIES & TOOLS USED

React.js	JavaScript library for building dynamic and interactive user interfaces.
JavaScript	Programming language that adds interactivity, logic, and functionality to web pages.
Tailwind CSS	Utility-first framework for fast and consistent UI styling.
React Router	Library for handling navigation and routing in React applications.

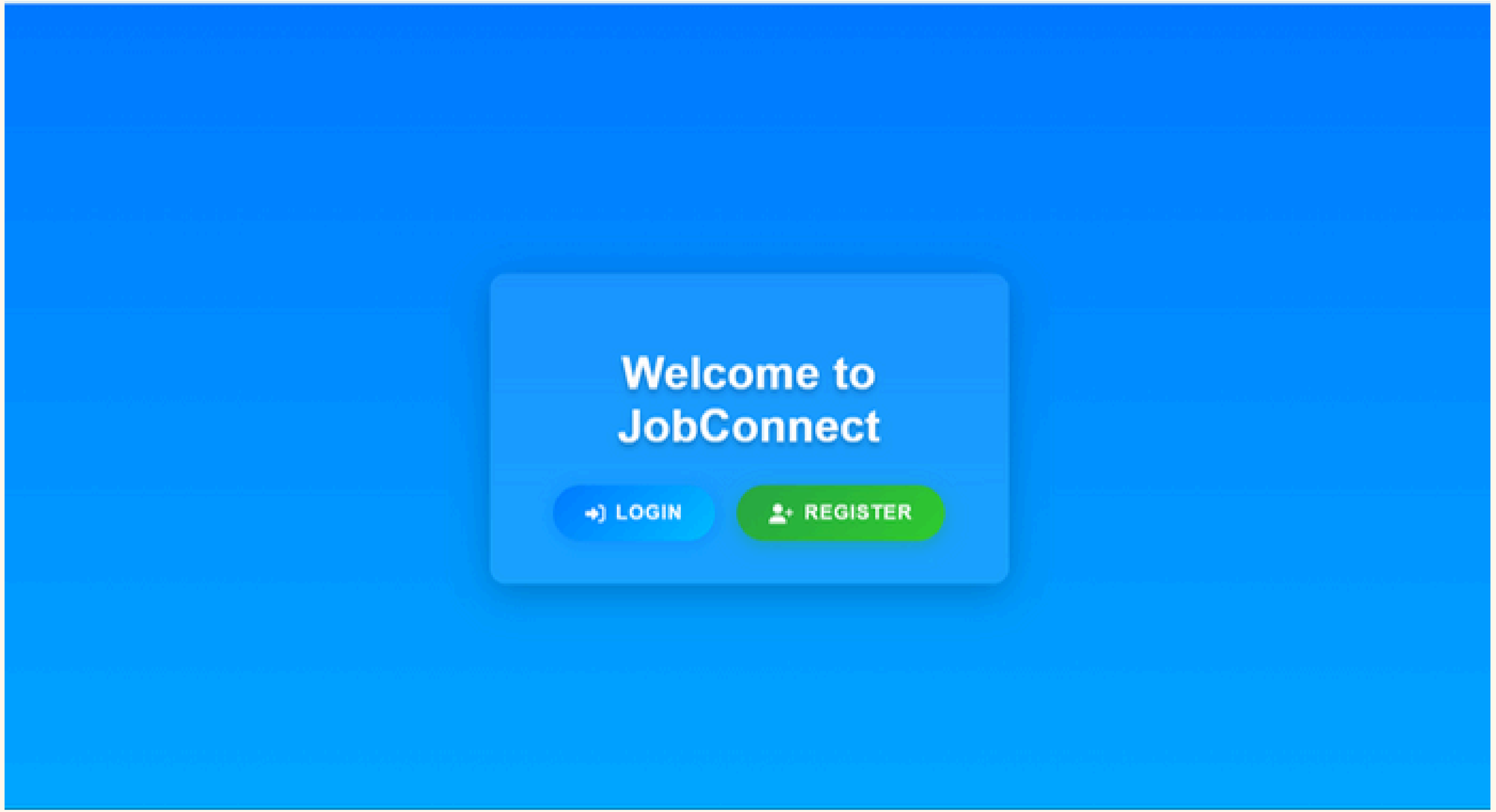
USER FLOW DIAGRAM



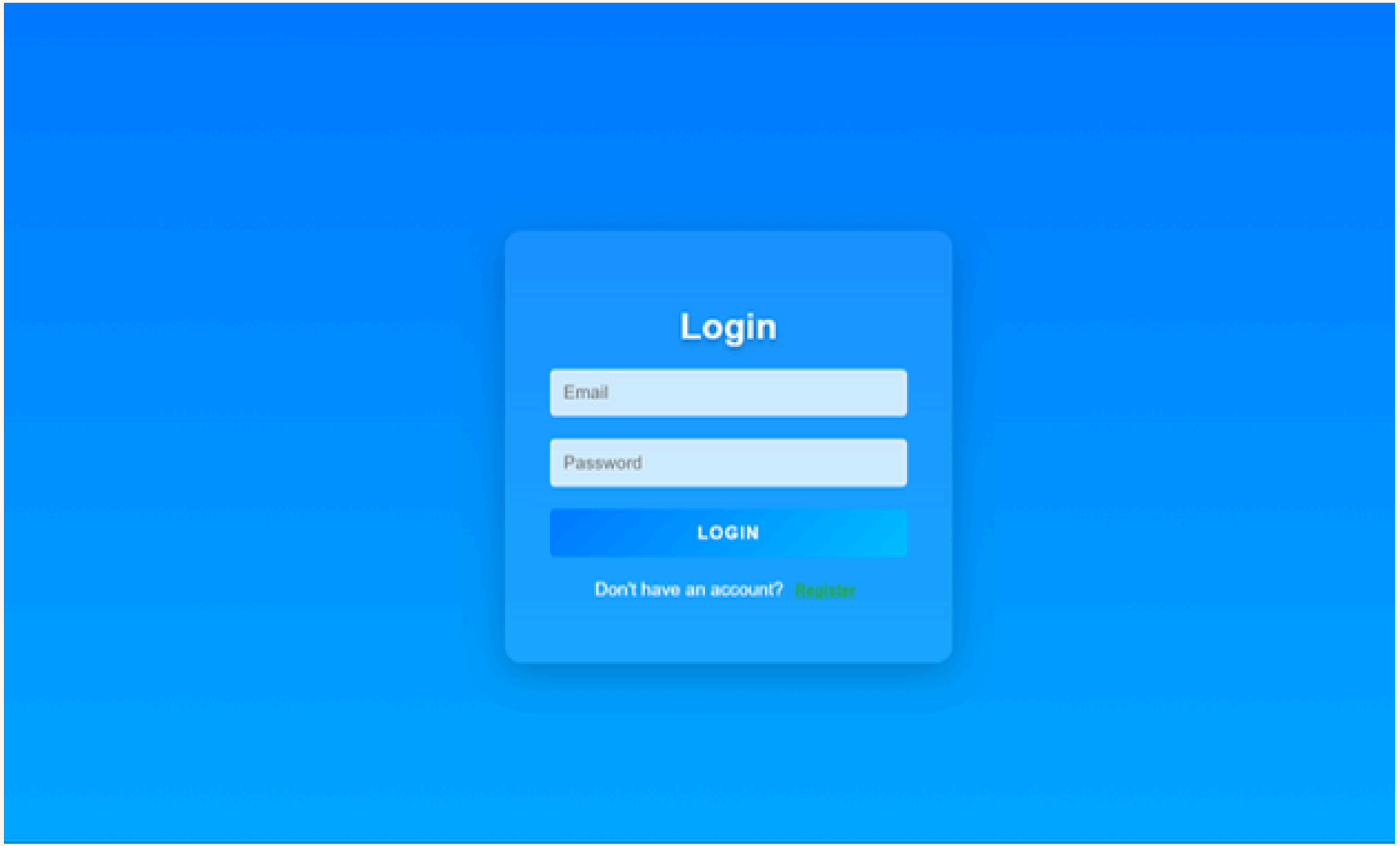
FRONTEND IMPLEMENTATION

Step 1: Setup Project Environment
Step 2: Install Required Dependencies
Step 3: Setup Project Structure
Step 4: Implement Routing in App.js
Step 5: Implement Key Components
Step 6: Connect Frontend to Backend
Step 7: Testing

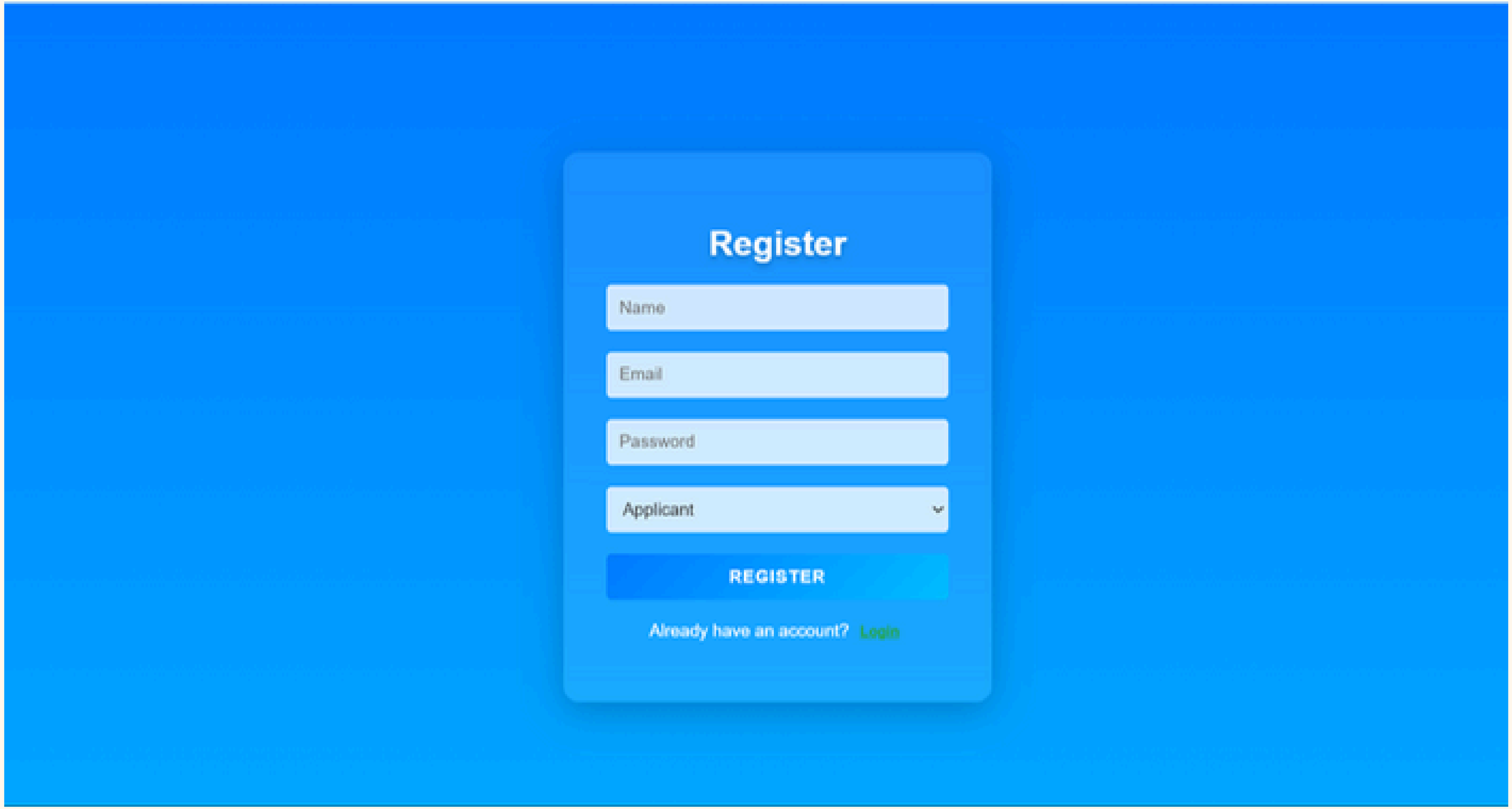
Welcome Page



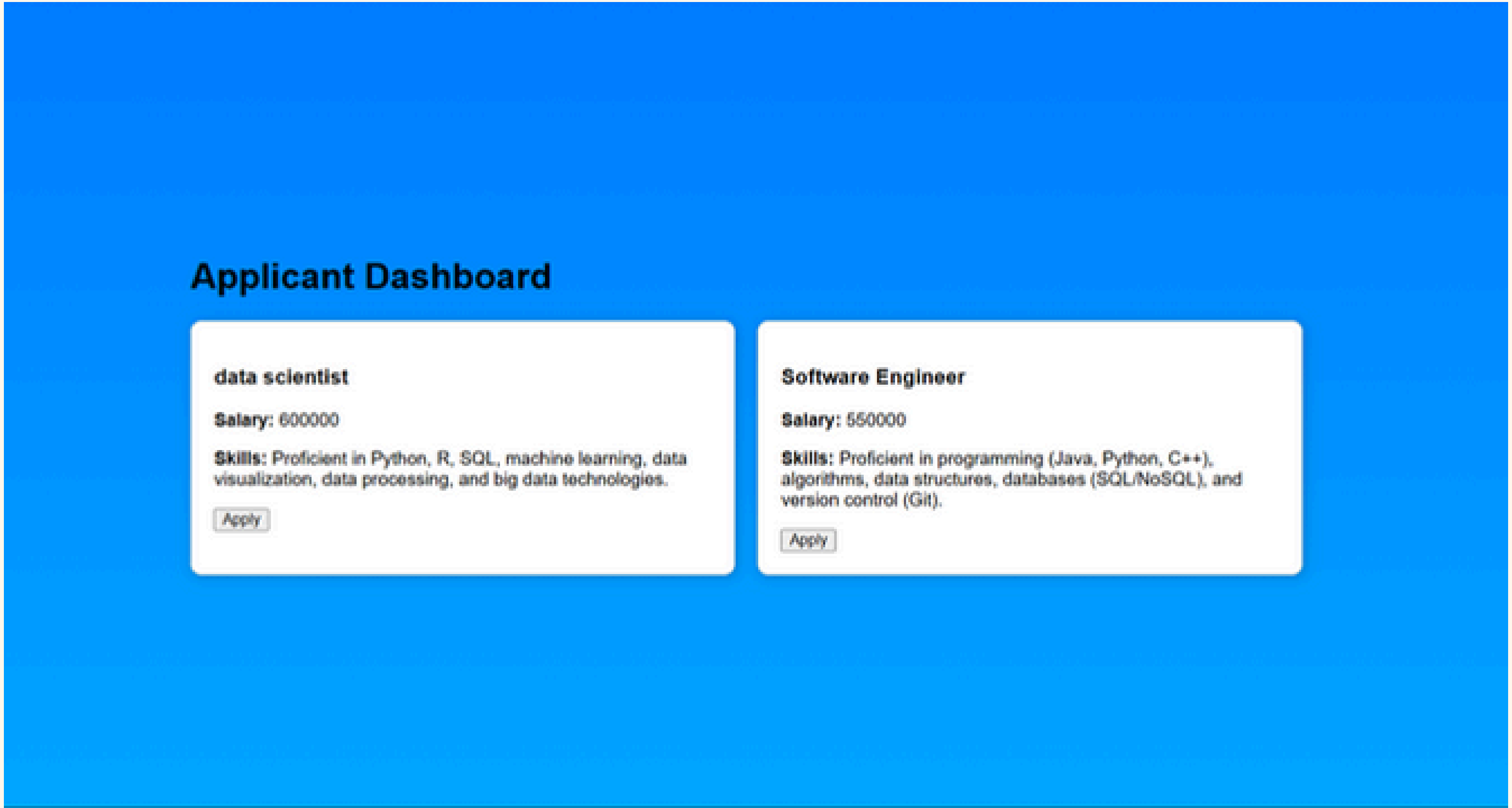
Login Page



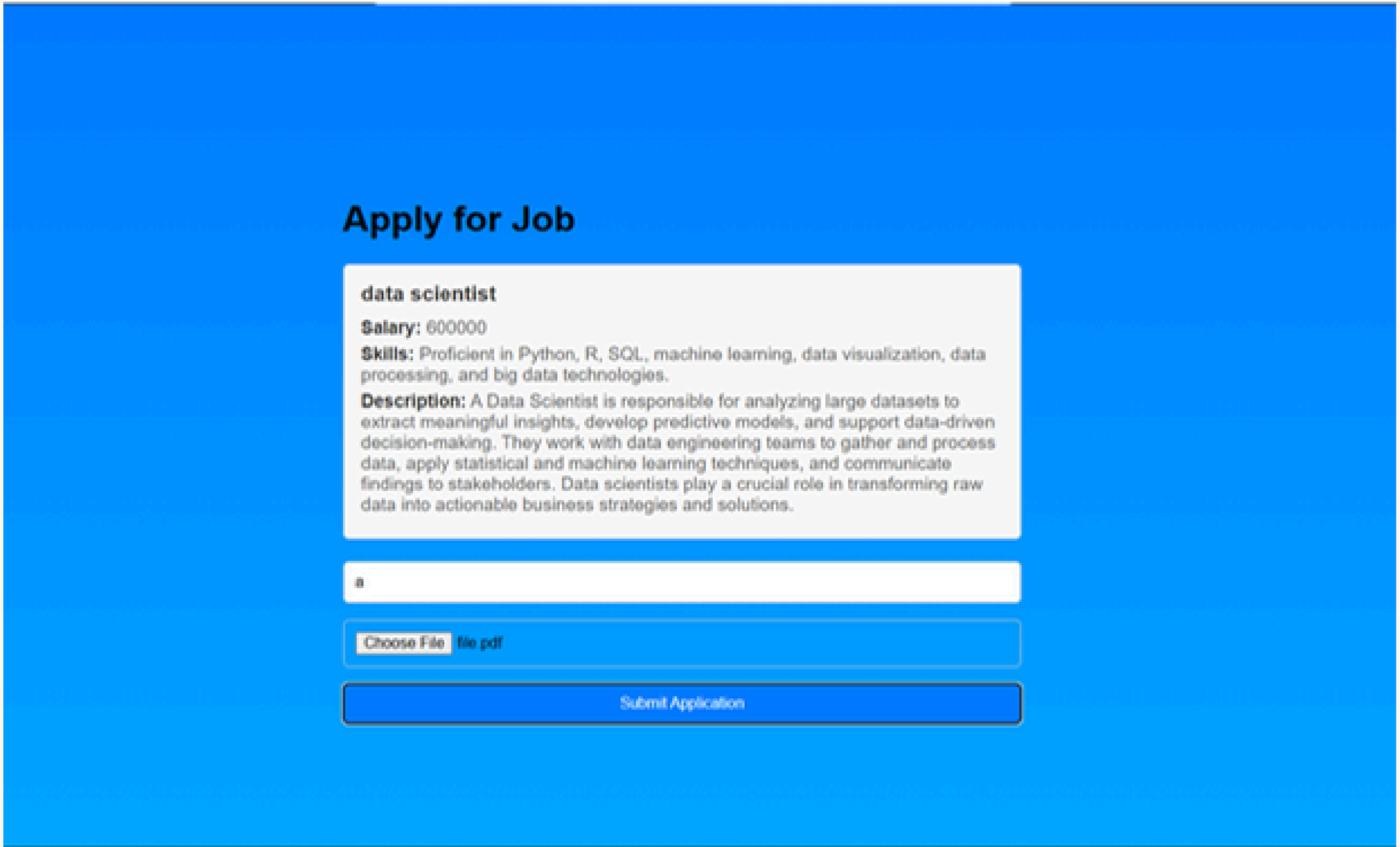
Register Page



Applicant Dashboard



Apply Form Page



Recruiter Dashboard

data scientist

A Data Scientist is responsible for analyzing large datasets to extract meaningful insights, develop predictive models, and support data-driven decision-making. They work with data engineering teams to gather and process data, apply statistical and machine learning techniques, and communicate findings to stakeholders. Data scientists play a crucial role in transforming raw data into actionable business strategies and solutions.

Salary: 600000

Skills: Proficient in Python, R, SQL, machine learning, data visualization, data processing, and big data technologies.

Applicants:

Name: a

Resume: View Resume

Name: a

Resume: View Resume

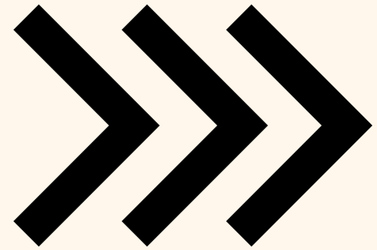
Name: a

Resume: View Resume

Name: q

Resume: View Resume

BACK END



TECHNOLOGIES & TOOLS USED IN BACKEND

Flask	Python web framework for handling requests and responses
PostgreSQL	Database for storing job listings, user data, applications, etc.
psycopg2	PostgreSQL adapter for connecting Flask with the database
Flask CORS	Allows frontend to communicate with backend securely

DATABASE SCHEMA (TABLES USED)

Users Table (user) → Stores user details like ID, name, email, password,role.

Query | Query History

```
1  select * from users;
```

Data Output | Messages | Notifications

SQL

	id [PK] integer	name character varying (100)	email character varying (100)	password character varying (100)	role character varying (20)
1	1	a	a@gmail.com	a	applicant
2	2	q	q@gmail.com	q	applicant
3	3	z	z@gmail.com	z	recruiter
4	4	w	w@gmail.com	w	recruiter
5	5	e	e@gmail.com	e	recruiter

DATABASE SCHEMA (TABLES USED)

Jobs Table (jobs) → Stores job postings, descriptions, salary,company info

Query | Query History

```
1  select * from jobs;
```

Data Output | Messages | Notifications

SQL

	id [PK] integer	recruiter_id integer	title character varying	description text	salary character var	skills text	posted_date timestamp wi
1	1	3	data scientist	A Data Scientist is responsible f...	600000	Proficient in...	2025-03-0...
2	2	3	Software En...	A Software Engineer designs, d...	550000	Proficient in...	2025-03-0...

DATABASE SCHEMA (TABLES USED)

Applications Table (applications) → Tracks which user applied to which job

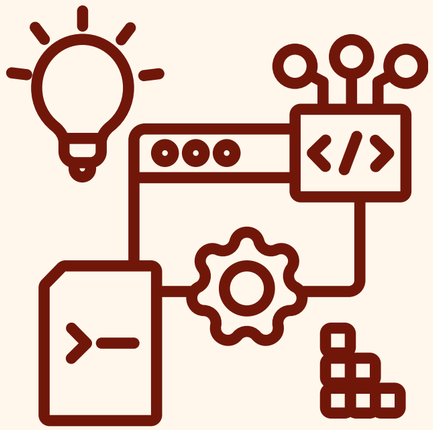
Query | Query History

```
1  select * from applications;
```

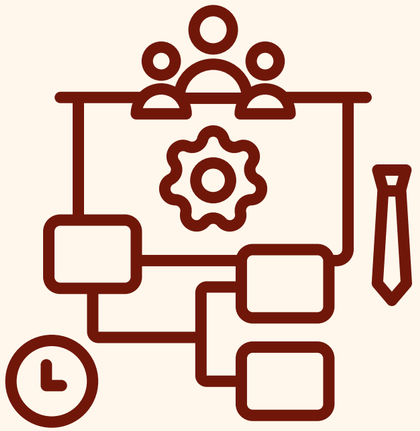
Data Output | Messages | Notifications

	id [PK] integer	job_id integer	applicant_id integer	resume_path character varyin	ats_score double precis	status character varyi	name character var	email character varying (7
1	10	1	[null]	uploads\file....	[null]	pending	aa	[null]
2	11	1	1	uploads\Doc...	[null]	pending	[null]	[null]
3	12	1	1	uploads\dha...	[null]	pending	a	a@gmail.com
4	13	1	1	uploads\dha...	45.94	pending	a	a@gmail.com
5	14	1	2	uploads\cert...	1.13	pending	q	q@gmail.com
6	15	1	1	uploads\oral...	0.71	pending	a	a@gmail.com
7	16	1	1	uploads\dha...	45.94	pending	a	a@gmail.com
8	17	2	1	uploads\dha...	31.17	pending	a	a@gmail.com
9	18	2	2	uploads\Mul...	0	pending	q	q@gmail.com

HOW BACKEND INTERACTS WITH FRONTEND?

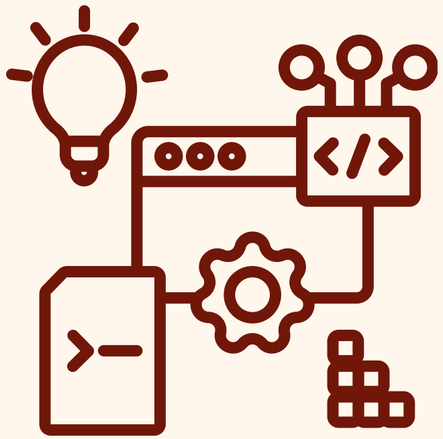


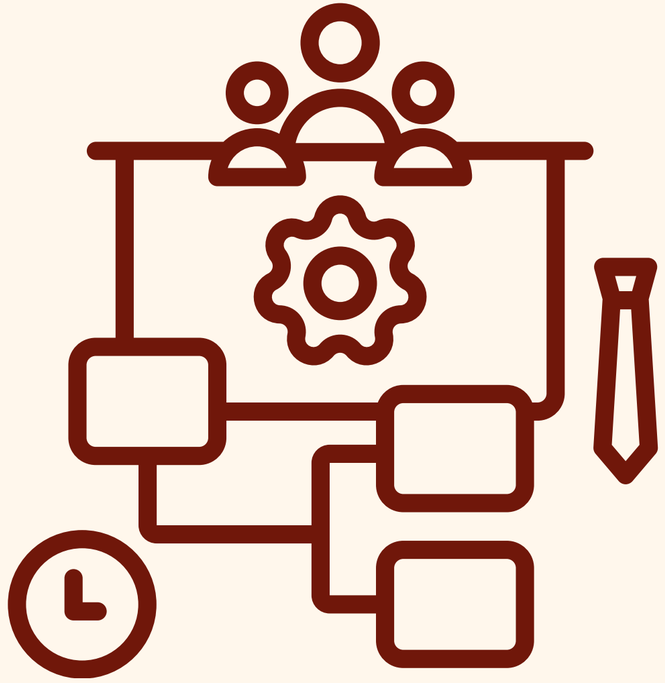
- Frontend sends requests (React page calls Flask API).
  Example: User clicks "Login"
- Flask processes request, interacts with PostgreSQL database.
  Checks credentials in PostgreSQL
- Database returns data (job listings, user details, application status, etc.)
  Fetches user details from the database
- Flask converts data into JSON and sends it back to the frontend.
- Frontend displays data to users (Job Listings, Application Confirmation, etc.)
  Sends back success/failure message to frontend

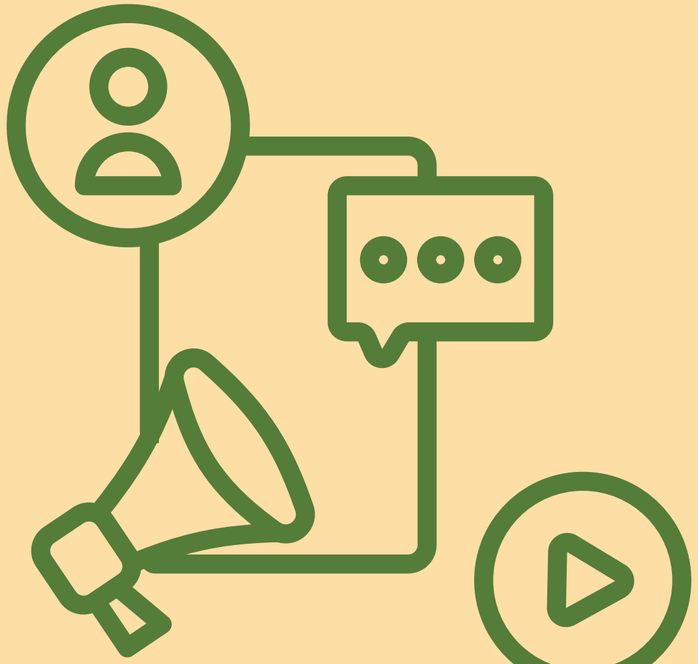


WORKS

- Email integration for automated resume selection .
- Recruiters are able to do other operations like delete ,add or edit the job postings.
- Job application restrictions implementation.







THANK YOU