A

Project Work Report

On

"JOB MARKET ANALYTICS AND OPPORTUNITY PREDICTION"

Submitted to

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTOMOMOUS)

Affiliated to JNTUA, Anathapuramu

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (AI&ML)

during the academic year 2025-2026

Submitted by

Student Name	Register Numbers
M.Vani	22781A3387
P.Yaswitha	22781A33A0
S.Govardhan	22781A33B3
T.Brahmaiah	22781A33D0
P.Sudheer	23785A3316

Under the esteemed guidance of

Dr.M.Lavanya (HOD)

Department of CSE(AI&ML)

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY(AUTONOMOUS)

Affiliated to JNTUA, Anathapuramu-515002(A.P) & Approved by AICTE, New Delhi

Accredited by NAAC, Bengaluru & NBA, New Delhi

An ISO 9001:2000 Certified Institution

R.V.S. Nagar, Chittoor-517127(A.P), India www.svcetedu.org

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY(AUTONOMOUS)

Affiliated to JNTUA, Anathapuramu-515002(A.P) & Approved by AICTE, New Delhi
Accredited by NAAC, Bengaluru & NBA, New Delhi
An ISO 9001:2000 Certified Institution
R.V.S. Nagar, Chittoor-517127(A.P), India
www.svcetedu.org

CERTIFICATE

This is to certify that the project entitled, **"JOB MARKET ANALYTICS AND OPPORTUNITY PREDICTION"** is a bonafide work carried out by the following students:

Student Name	Register Number
M.Vani	22781A3387
P.Yaswitha	22781A33A0
S.Govardhan	22781A33B3
T.Brahmaiah	22781A33D0
P.Sudheer	23785A3316

In partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** during the academic year **2025 – 2026.**

SIGNATURE OF THE GUIDE
Dr.M.Lavanya,(HOD)CSM

SIGNATURE OF THE HOD
Dr. M. Lavanya, MCA, M. Tech, Ph.D
HOD & Associate Professor

INTERNAL EXAMINER

EXTERNAL EXAMINIER

Viva-Voce Conducted on _____

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY(AUTONOMOUS)

Affiliated to JNTUA, Anathapuramu-515002(A.P) & Approved by AICTE, New Delhi
Accredited by NAAC, Bengaluru & NBA, New Delhi
An ISO 9001:2000 Certified Institution
R.V.S. Nagar, Chittoor-517127(A.P), India
www.svcetedu.org

Department of CSE(AI&ML)

DECLARATION

We, **Vani (22781A3387), Yaswitha (22781A33A0), Govardhan (22781A33B3), Brahmaiah (22781A33D0), and Sudheer (23785A3316)** hereby declare that the Project Report entitled **"Job market analytics and opportunity prediction"** is a bonafide work carried out by us under the guidance of **Dr.M.Lavanya(HOD)CSM** Sri Venkateswara College of Engineering and Technology (Autonomous), Chittoor.

This project report is submitted in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)** during the academic year **2025 – 2026**.

We further declare that this project work has not been submitted previously to any other university or institution for the award of any degree or diploma. The results and conclusions presented in this report are original and carried out by us.

Student Name	Register Number
M.Vani	22781A3387
P.Yaswitha	22781A33A0
S.Govardhan	22781A33B3
T.Brahmaiah	22781A33D0
P.Sudheer	23785A3316

ACKNOWLEDGEMT

A grateful thanks to **Dr. R. Venkataswamy**, Chairman, **Sri Venkateswara College of Engineering and Technology**, for providing education in their esteemed institution.

We wish to record our deep sense of gratitude and profound thanks to our beloved Vice Chairman, **Sri. R. V. Srinivas**, for his valuable support throughout the course.

We express our sincere thanks to **Dr. M. Mohan Babu**, our beloved Principal, for his encouragement and suggestions during the course of study.

We wish to convey our gratitude and express our sincere thanks to **Dr.M.Lavanya (HOD)CSM** Project Guide / Mentor, Department of Computer Science and Engineering (AI & ML), for giving us her inspiring guidance in undertaking our project work.

We also express our sincere thanks to **Dr. M. Lavanya, MCA, M.Tech, Ph.D**, Head of the Department, CSE (AI & ML), for her keen interest, stimulating guidance, and encouragement throughout all stages of the project, which helped us to successfully complete this project.

We wish to convey our gratitude and express our sincere thanks to all the **Project Review Committee members** for their support and cooperation rendered for the successful submission of our project work. Finally, we would like to express our sincere thanks to all teaching and non-teaching staff members, our parents, and friends for their continuous support and encouragement in completing this project successfully.

Student Name	Register Number
M.Vani	22781A3387
P.Yaswitha	22781A33A0
S.Govardhan	22781A33B3
T.Brahmaiah	22781A33D0
P.Sudheer	23785A3316

Vision and Mission of the Department (R20 Regulations)

VISION

- To achieve an excellent standard of quality education by using the latest tools and technologies in Artificial Intelligence and Machine Learning, and to disseminate innovative solutions to real-world problem domains such as transportation, healthcare, and smart cities.

MISSION

- To develop professionals who are skilled in Artificial Intelligence and Machine Learning with strong analytical and problem-solving abilities.
- To impart rigorous training through state-of-the-art concepts, tools, and technologies in AI & ML for solving real-world problems.
- To establish centers of excellence in advanced computing and artificial intelligence that inculcate ethical values, innovative research capabilities, and leadership qualities among students to contribute to national development.

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

Program Educational Objectives (PEOs) – R20 Regulations

Program Educational Objectives (PEOs)

PEO1: To enable graduates to analyze and solve a wide range of computing and data-driven problems relevant to industry and society.

PEO2: To prepare students to design and develop intelligent systems and applications using machine learning, data analytics, and visualization techniques.

PEO3: To produce professionally competent graduates who engage in lifelong learning, higher studies, research, and professional activities related to engineering and society.

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

Program Specific Outcomes (PSOs) – R20 Regulations

Program Specific Outcomes (PSOs)

PSO1: Ability to apply technical knowledge and use modern AI & ML tools, frameworks, and programming techniques to solve real-world problems such as transport analytics and delay prediction.

PSO2: Ability to design and develop intelligent applications using machine learning and artificial intelligence techniques in domains including data analytics, prediction systems, and decision-support system

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
R.V.S. NAGAR, CHITTOOR-517 127, ANDHRA PRADESH
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AI&ML)

Program Outcomes (POs)

On successful completion of the B.Tech CSE (AI & ML) program, graduates will be able to:

1. Engineering Knowledge:

   Apply knowledge of mathematics, science, engineering fundamentals, and AI & ML to solve complex engineering problems.

2. Problem Analysis:

   Identify, formulate, and analyze complex problems using data, algorithms, and machine learning principles.

3. Design/Development of Solutions:

   Design intelligent systems such as predictive models and analytics platforms considering societal and environmental needs.

4. Conduct Investigations:

   Use research-based methods, data analysis, and experimentation to draw valid conclusions.

5. Modern Tool Usage:

   Use modern tools such as Python, ML frameworks, databases, and visualization tools for modeling and prediction.

6. Engineer and Society:

   Apply computing solutions responsibly by considering societal, safety, and legal issues.

7. Environment and Sustainability:

   Understand the impact of engineering solutions in sustainable transport and smart infrastructure.

8. Ethics:

   Apply ethical principles in data handling, modeling, and deployment of AI systems.

9. Individual and Team Work:

   Work effectively as an individual and as part of a team in multidisciplinary environments.

10. Communication:

    Communicate effectively through reports, presentations, and documentation.

11. Project Management and Finance:

    Apply management principles to plan, execute, and complete projects efficiently.

12. Life-long Learning:

    Recognize the importance of continuous learning to adapt to evolving AI & ML technologies.

SRI VENKATESWARA COLLEGE OF ENGINEERING AND TECHNOLOGY(Autonomous)

IV B.Tech II Semester – CSE (AI & ML)

20ACM29: PROJECT WORK, SEMINAR AND INTERNSHIP IN INDUSTRY

L	T	P	C
–	–	–	12

Course Outcomes (COs)

- After successful completion of this course, students will be able to:
- **CO1:** Design and develop intelligent systems such as machine learning–based transport delay prediction systems using appropriate tools and techniques.
- **CO2:**Analyze real-world problems by considering factors such as traffic, weather, sustainability, and social impact while developing AI-based solutions.
- **CO3:** Work individually or in teams and communicate effectively through project reports, presentations, and demonstrations of AI & ML applications.

Abstract

In today's rapidly changing job market, students and job seekers often face difficulties in selecting suitable career paths due to lack of proper guidance and awareness about industry requirements. Traditional career counseling methods are mostly manual, time-consuming, and may not provide personalized recommendations. To overcome these limitations, this project proposes **JobAI: An Intelligent Career Guidance System** based on machine learning techniques.

The JobAI system is designed as a web-based application that predicts job demand and analyzes skill gaps using data-driven approaches. The system collects user information such as educational background, technical skills, experience level, and job preferences through an interactive interface. This data is processed using machine learning algorithms to classify job roles into different demand categories such as High, Medium, and Low. In addition, a rule-based mechanism is integrated to identify missing skills and recommend suitable learning paths and certifications.

The backend of the system is implemented using the Python Flask framework, while the frontend is developed using HTML, CSS, and JavaScript. An SQLite database is used to store user details, prediction results, and analysis reports securely. The system ensures fast response time, high accuracy, and reliable performance through efficient data processing and optimized models.

The proposed JobAI system helps users make informed career decisions by providing personalized guidance, improving employability, and reducing uncertainty in career planning. It also supports educational institutions in offering data-driven counseling and placement assistance. Overall, this project demonstrates the effective application of artificial intelligence and machine learning in developing a smart and user-friendly career guidance platform.

INTRODUCTION

In recent years, the job market has become highly competitive due to rapid industrial growth, globalization, and continuous technological advancements. The increasing number of graduates and professionals entering the workforce has made it challenging for job seekers to identify suitable job opportunities that match their skills, experience, and career goals. As a result, individuals often struggle to make informed career decisions in a dynamic and fast-changing job environment.

Traditional job portals such as online recruitment websites primarily focus on listing available job openings. Although these platforms provide access to a large number of job postings, they lack intelligent analytics and predictive features. Job seekers are required to manually search, compare, and analyze multiple job listings, which is time-consuming and inefficient. Additionally, these platforms do not offer insights into job market trends, demand for specific roles, or the probability of job opportunities based on individual profiles.

To address these limitations, this project proposes a **Job Market Analytics and Opportunity Prediction System using Machine Learning**. The system analyzes job market data such as job titles, locations, required experience, and salary information to generate meaningful insights. Machine learning techniques are used to predict whether a job opportunity is classified as high or low, based on relevant features. This predictive approach assists users in understanding their chances in the job market and making better career decisions.

The proposed system is implemented as a web-based application using the Flask framework and an SQLite database. An interactive dashboard is developed to visualize job market trends through charts and tables. Features such as job search, filters, and prediction functionality enhance user experience and usability. Overall, this project aims to transform raw job data into actionable intelligence, helping job seekers and organizations gain deeper insights into the job market.

- To analyze job market data using data analytics techniques
- To predict job opportunities using machine learning models
- To provide an interactive and user-friendly web dashboard
- To assist job seekers in making informed career decisions

LITERATURE REVIEW

Several research studies have been conducted in the area of job market analysis and job opportunity prediction using data mining and machine learning techniques. With the rapid growth of online job portals, large volumes of job-related data are generated every day. Researchers have explored various analytical and predictive approaches to extract useful insights from this data and assist job seekers and organizations.

Many studies focus on **job trend analysis using historical job postings**. These studies analyze past job market data to identify patterns such as in-demand job roles, popular skills, and salary trends. By examining historical data, researchers are able to predict future job market behavior and understand changes in employment demand across different industries and locations.

Several machine learning algorithms have been applied for job market prediction tasks. Commonly used models include **Decision Trees, Logistic Regression, Naive Bayes, and Support Vector Machines**. Decision Tree algorithms are widely used due to their simplicity and interpretability, while Logistic Regression is applied for binary classification problems such as predicting job availability or opportunity levels. These models help classify job opportunities based on features like experience, location, and salary.

Another important area of research involves the **analysis of job market demand using data visualization tools**. Visualization techniques such as bar charts, pie charts, and trend graphs are used to present job statistics in a clear and understandable format. Dashboards developed using visualization libraries help users quickly interpret complex job market data and identify trends without requiring technical knowledge.

Despite these advancements, existing job portals and systems have several limitations. Most platforms primarily focus on job listings and lack predictive capabilities. They do not provide insights into job opportunity levels or personalized recommendations based on user profiles. Additionally, many systems do not integrate machine learning models effectively with interactive dashboards, limiting their usefulness for analytical decision-making.

The literature clearly indicates the need for **intelligent systems that combine job market analytics with machine learning-based prediction**. Such systems can overcome the limitations of traditional job portals by offering predictive insights, trend analysis, and user-friendly visualization. These findings motivate the development of the proposed Job Market Analytics and Opportunity Prediction System, which aims to integrate data analysis, prediction, and visualization into a single unified platform.

DATA COLLECTION

Data collection is a crucial step in any data-driven and machine learning-based system. The performance and accuracy of the proposed Job Market Analytics and Opportunity Prediction System largely depend on the quality, relevance, and completeness of the data used. In this project, job market data is collected to analyze employment trends and predict job opportunities effectively.

The dataset used in this project consists of job-related information collected from publicly available job market datasets. These datasets represent real-world job postings commonly found on online job portals. The collected data provides insights into various aspects of the job market, such as popular job roles, geographic demand, experience requirements, and salary ranges.

3.1 Data Sources

The job market data is obtained from publicly available datasets and sample job listings. These datasets are chosen because they closely resemble real-world job portal data and are suitable for analytical and predictive tasks. The data is initially stored in a CSV (Comma Separated Values) format, which allows easy handling and preprocessing.

3.2 Dataset Attributes

The dataset contains the following key attributes:

- **Job Title**: Represents the role or position offered by the organization (e.g., Python Developer, Data Analyst).
- **Location**: Indicates the city or region where the job is located.
- **Experience**: Specifies the required years of experience for the job role.
- **Salary**: Represents the annual salary offered for the position.

These attributes are selected because they play an important role in determining job opportunities and market demand.

3.3 Data Format

The collected data is stored in a structured CSV format with clearly defined column headers. An example of the dataset format is shown below:

```
job_title, location, experience, salary
Python Developer, Hyderabad, 2, 600000
Data Analyst, Bangalore, 1, 500000
```

This structured format simplifies data preprocessing and database integration.

3.4 Data Preprocessing

Before using the data for analysis and prediction, preprocessing steps are performed to ensure data quality. These steps include:

- Removal of duplicate job records
- Handling missing or inconsistent values
- Converting numerical fields such as experience and salary into appropriate data types
- Validating categorical fields such as job title and location

Preprocessing helps improve the reliability of analytical results and machine learning predictions.

3.5 Data Storage

After preprocessing, the dataset is imported into an SQLite database. SQLite is chosen due to its lightweight nature and ease of integration with Python and Flask applications. The database stores job information in a structured table, enabling efficient querying and retrieval of data for analytics, filtering, and visualization in the dashboard.

3.6 Importance of Data Collection

Accurate and well-structured data collection enables the system to:

- Analyze job market trends effectively
- Provide meaningful visual insights
- Support machine learning-based opportunity prediction
- Enhance overall system performance

Thus, data collection forms the foundation of the proposed job market analytics system.

SYSTEM STUDY

System study involves a detailed analysis of the current system and identification of its drawbacks, followed by the design of an improved proposed system. This chapter discusses the existing job market systems, their limitations, and the motivation behind developing the proposed Job Market Analytics and Opportunity Prediction System.

4.1 Existing System

The existing job market systems mainly consist of online job portals such as recruitment websites and employment platforms. These systems provide job seekers with access to job listings posted by various companies. Users can search for jobs based on keywords, location, and experience.

However, the existing systems have the following characteristics:

- Focus mainly on displaying job listings
- No analytical insights about job market trends
- No prediction of job opportunity levels
- Manual effort required by users to analyze job suitability
- Lack of intelligent decision-support features

Most existing job portals do not utilize advanced machine learning techniques to analyze job market data. As a result, users are unable to understand trends such as high-demand roles, salary patterns, or the probability of job opportunities based on their profile.

4.2 Limitations of the Existing System

The major limitations of the existing system include:

- Absence of job opportunity prediction
- No intelligent analytics or visualization dashboard
- Lack of personalized insights for job seekers
- Time-consuming job search process
- Limited support for decision making

Due to these limitations, job seekers often face difficulties in choosing suitable career opportunities.

4.3 Proposed System

The proposed system is a **Job Market Analytics and Opportunity Prediction System using Machine Learning**. It is designed to overcome the limitations of the existing system by integrating data analytics, machine learning, and web technologies.

The proposed system analyzes job market data such as job titles, locations, experience requirements, and salary information. Machine learning techniques are applied to predict whether a job opportunity is classified as high or low. An interactive web-based dashboard is developed to present insights in a visual and

user-friendly manner.

4.4 Features of the Proposed System

The key features of the proposed system include:

- Job market data analysis
- Machine learning-based job opportunity prediction
- Interactive dashboard with charts and tables
- Job search and filtering by location, experience, and role
- User login and registration functionality

These features help job seekers understand the job market better and make informed career decisions.

4.5 Advantages of the Proposed System

The proposed system offers several advantages over existing systems:

- Provides predictive insights rather than static job listings
- Improves decision making for job seekers
- Reduces time required to search and analyze jobs
- Offers visual representation of job market trends
- Easy to use and accessible through a web browser

4.6 System Overview

The system follows a structured workflow starting from data collection and preprocessing to analysis, prediction, and visualization. Job data is stored in a database, processed using machine learning models, and displayed through an interactive dashboard.

METHODOLOGY

The methodology describes the systematic approach followed to design, develop, and implement the proposed **Job Market Analytics and Opportunity Prediction System**. This project follows a structured machine learning workflow that includes data preprocessing, feature selection, model training, evaluation, and deployment through a web application.

5.1 Overall Workflow

The overall methodology of the proposed system consists of the following major steps:

1. Data collection
2. Data preprocessing
3. Feature engineering
4. Model training
5. Model evaluation
6. Prediction and visualization

Each step plays a crucial role in ensuring accurate analysis and reliable prediction of job opportunities.

5.2 Data Preprocessing

Raw job market data collected from CSV files may contain inconsistencies, missing values, or incorrect data types. Therefore, preprocessing is performed before using the data for machine learning and analysis.

The preprocessing steps include:

- Removing duplicate job records
- Handling missing or invalid values
- Converting experience and salary fields into numerical format
- Standardizing categorical attributes such as job title and location

Data preprocessing improves the quality of the dataset and helps enhance model performance.

5.3 Feature Selection and Engineering

Feature selection involves identifying the most relevant attributes that influence job opportunity prediction. In this project, the following features are considered:

- Job Title
- Location
- Experience
- Salary

Feature engineering techniques are applied to transform raw features into a suitable format for machine learning models. Numerical features are normalized where required, and categorical data is encoded appropriately.

5.4 Machine Learning Model Selection

A supervised machine learning approach is used in this project. The model is trained using labeled job data to predict whether a job opportunity is **High Opportunity** or **Low Opportunity**.

Commonly used classification algorithms such as Decision Trees and Logistic Regression are suitable for this type of prediction problem. These algorithms are selected due to their simplicity, interpretability, and efficiency.

5.5 Model Training

The machine learning model is trained using preprocessed job market data. The dataset is divided into training and testing sets to evaluate the model's performance.

During training:

- Input features are fed into the model
- The model learns patterns and relationships between job attributes
- Model parameters are optimized to improve prediction accuracy

---

5.6 Model Evaluation

After training, the model is evaluated using standard performance metrics such as:

- Accuracy
- Precision
- Recall

Evaluation helps determine how well the model predicts job opportunities and ensures reliability before deployment.

5.7 Prediction Process

Once the model is trained and validated, it is used to predict job opportunities based on user input. The user provides details such as experience, and the system predicts whether the job opportunity is high or low. The prediction result is displayed along with the probability score.

5.8 System Integration

The trained machine learning model is integrated into a Flask-based web application. Job data is stored in an SQLite database, and the model interacts with the database to perform analysis and prediction. Visualization tools are used to display insights through charts and tables on the dashboard.

IMPLEMENTATION

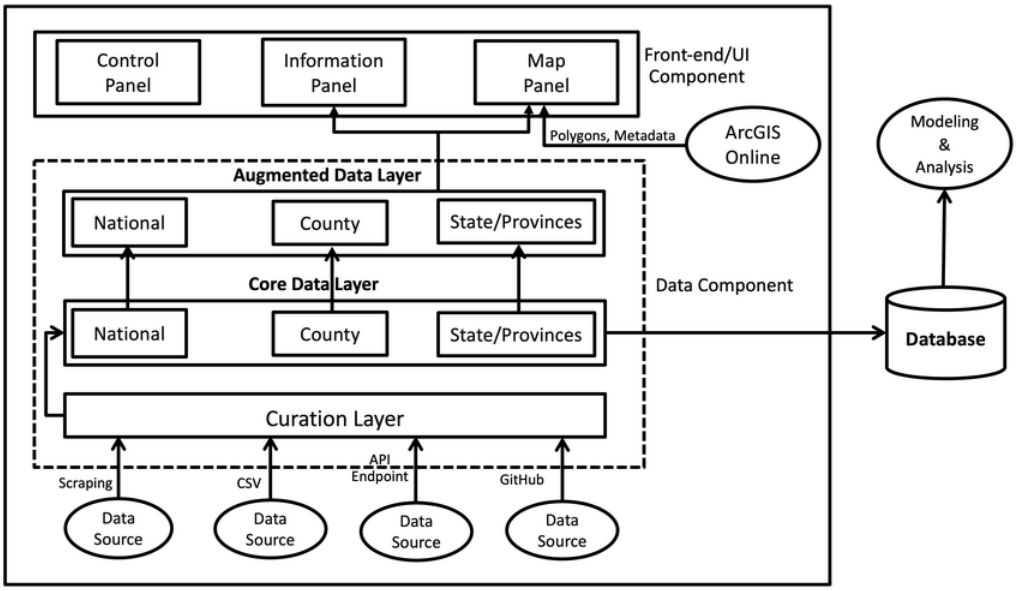
This chapter explains the implementation details of the proposed **Job Market Analytics and Opportunity Prediction System**. The system is developed using Python and integrates machine learning, database management, and web technologies to provide an interactive job market analytics platform.

6.1 System Architecture

The system follows a three-tier architecture:

1. **Presentation Layer** – Web interface developed using HTML, CSS, and JavaScript
2. **Application Layer** – Flask framework handling business logic and routing
3. **Data Layer** – SQLite database storing job market data and user information

This architecture ensures modularity, scalability, and easy maintenance.



6.2 Database Implementation

SQLite is used as the database for storing job market data and user details. It is lightweight and easy to integrate with Python applications.

The database contains the following tables:

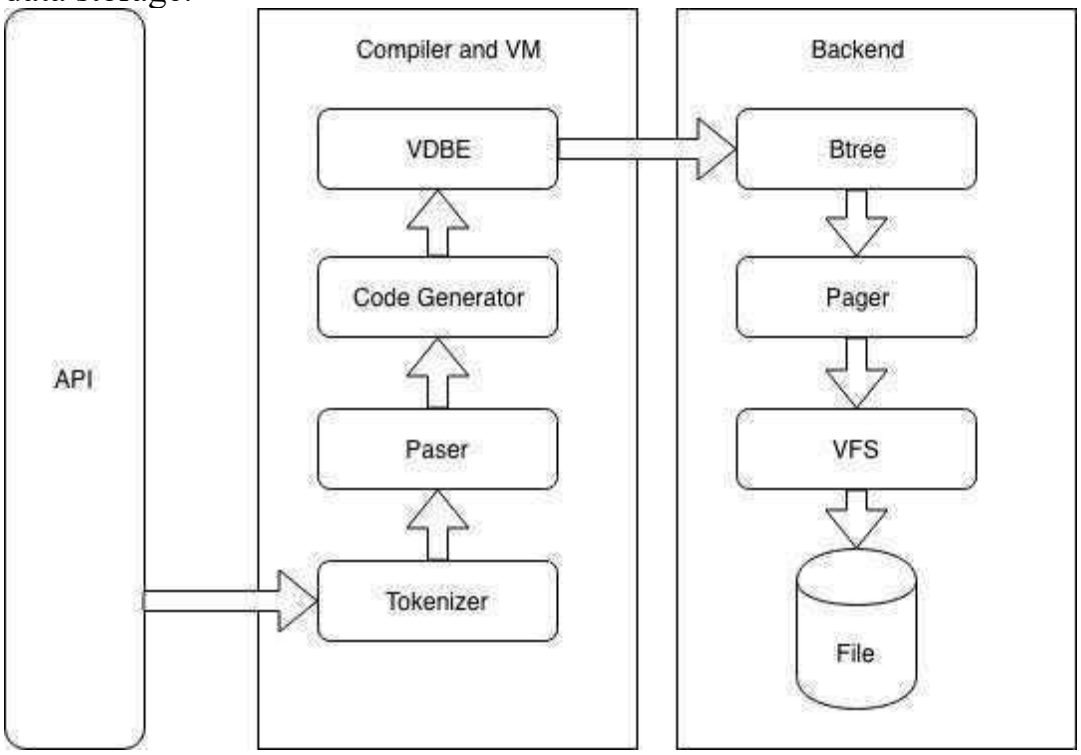
- **jobs_data** – Stores job title, location, experience, and salary
- **users** – Stores user login and registration information
- **predictions** – Stores job opportunity prediction results

SQL queries are used to insert, retrieve, and filter job records efficiently.

6.3 Data Loading Module

Job data is initially stored in a CSV file. A Python script is implemented to load this data into the SQLite database. The script reads job attributes from the CSV file and

inserts them into the jobs_data table. This process ensures structured and consistent data storage.



6.4 Machine Learning Model Integration

A pre-trained machine learning model is used to predict job opportunities. The model is trained using job market data and saved using the Joblib library. During runtime, the model is loaded into the Flask application and used to predict whether a job opportunity is high or low based on user input.

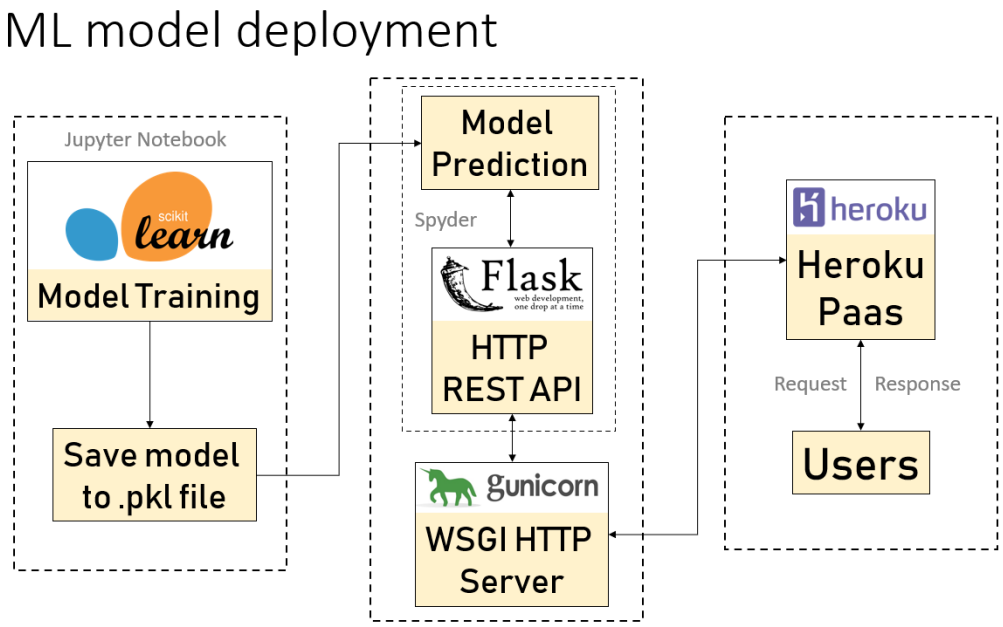
The prediction output includes both the predicted class and probability score, enhancing transparency and reliability.

6.5 Web Application Development

The web application is developed using the Flask framework. Flask routes are defined to handle user requests such as login, registration, dashboard access, and job prediction.

Key functionalities include:

- User authentication (login and registration)
- Dashboard displaying analytics and visualizations
- Job search and filter functionality
- Job opportunity prediction
-



6.6 Dashboard and Visualization

An interactive dashboard is implemented to present job market analytics. Data visualization is achieved using Chart.js, which displays information through bar charts and pie charts. These visualizations help users easily understand job trends, high-demand roles, and opportunity distribution.

6.7 Prediction Workflow

The prediction workflow begins when the user enters input details such as experience. The system processes the input, passes it to the machine learning model, and generates a prediction. The result is stored in the database and displayed to the user in real time.

6.8 Security and Session Management

Session management is implemented to ensure secure access to the dashboard. Only authenticated users can access the system features. Flask sessions are used to maintain user login state during interaction.

SYSTEM SPECIFICATION

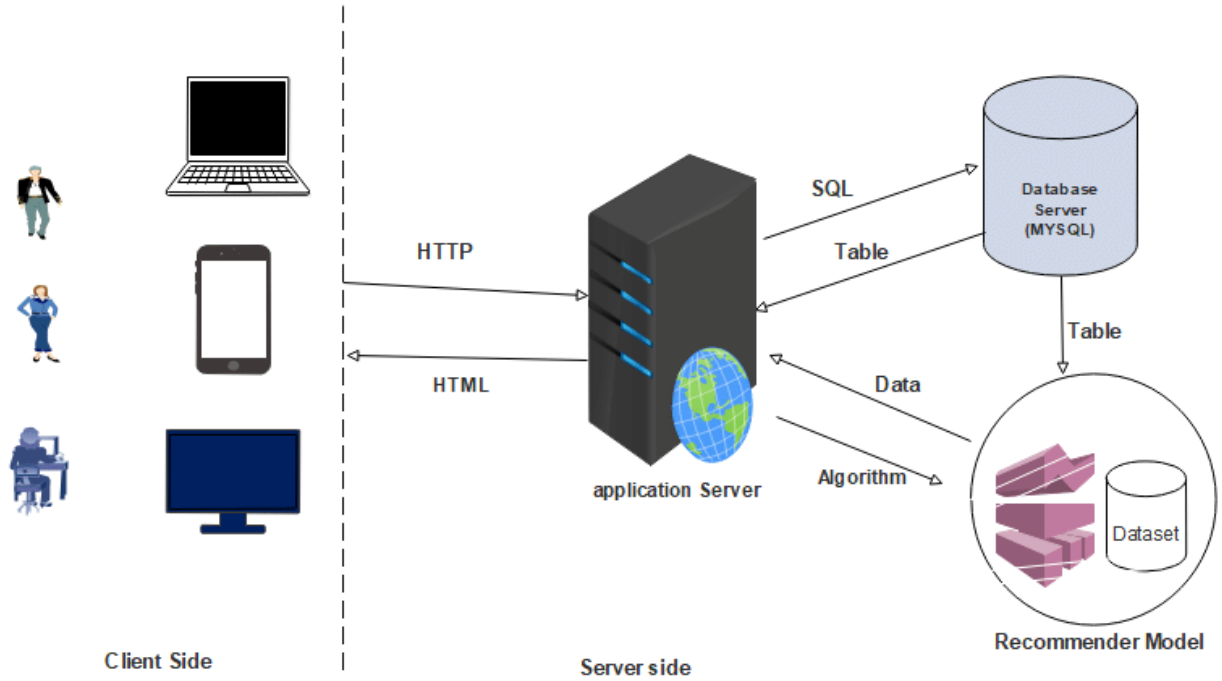
This chapter describes the hardware and software requirements needed to implement and execute the proposed **Job Market Analytics and Opportunity Prediction System**. Proper system specifications ensure smooth functioning, efficient performance, and reliable results.

7.1 Hardware Requirements

The proposed system does not require high-end hardware and can be executed on a standard personal computer. The minimum hardware requirements are as follows:

- **Processor**: Intel Core i3 or higher
- **RAM**: Minimum 4 GB
- **Hard Disk**: At least 20 GB free storage
- **Input Devices**: Keyboard and Mouse
- **Output Device**: Monitor

These hardware requirements are sufficient to run the machine learning model, database operations, and web application smoothly.



7.2 Software Requirements

The software requirements for the proposed system include the following:

- **Operating System**: Windows 10 or above
- **Programming Language**: Python 3.x
- **Web Framework**: Flask
- **Database**: SQLite
- **Machine Learning Libraries**: Scikit-learn, Joblib
- **Data Processing Libraries**: Pandas, NumPy

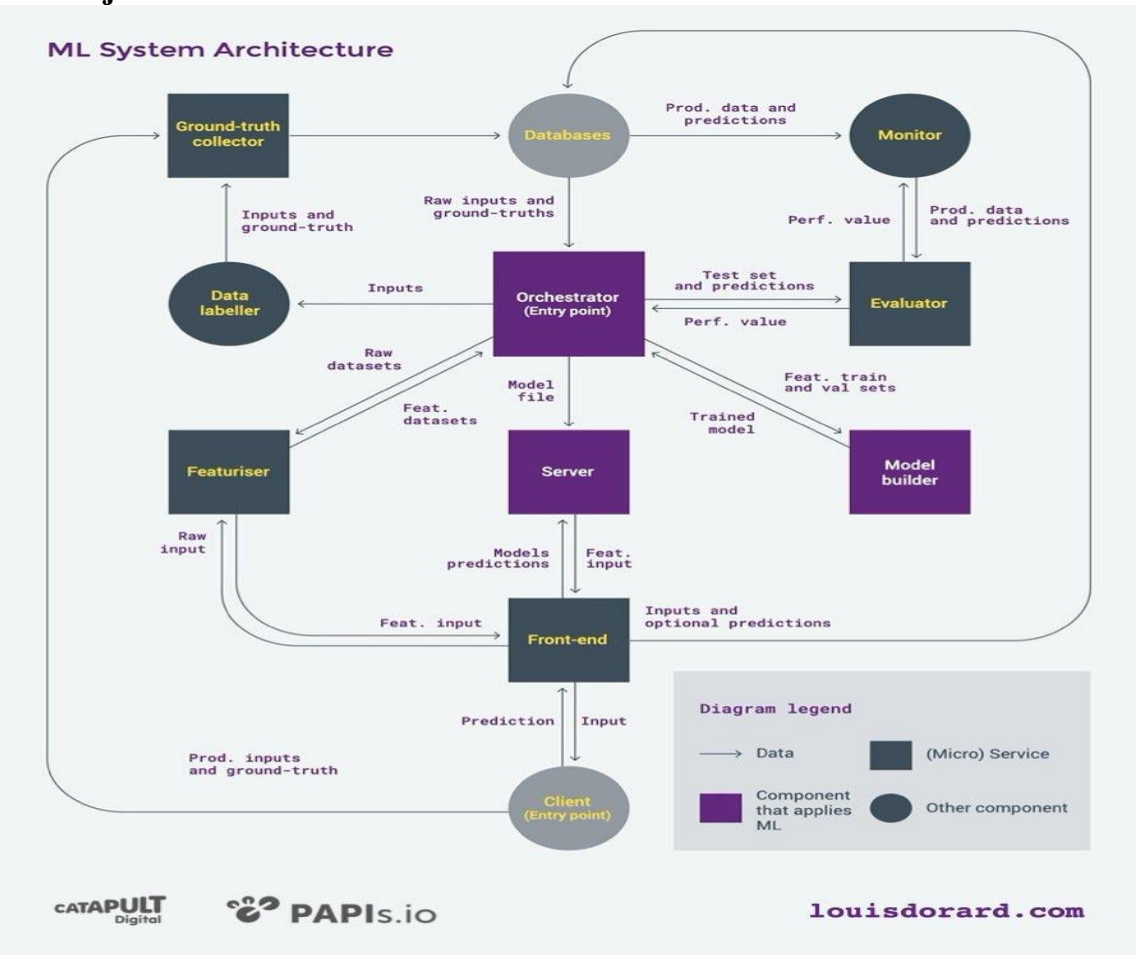
- **Visualization Tool**: Chart.js
- **Development Tool**: Visual Studio Code

All the above software tools are open-source and freely available, making the system cost-effective.

7.3 Tools and Technologies Used

The following tools and technologies are used in the project:

- **Python** – Core programming language
- **Flask** – Backend web framework
- **SQLite** – Lightweight database for data storage
- **Machine Learning Models** – For job opportunity prediction
- **HTML, CSS, JavaScript** – Frontend development
- **Chart.js** – Interactive data visualization



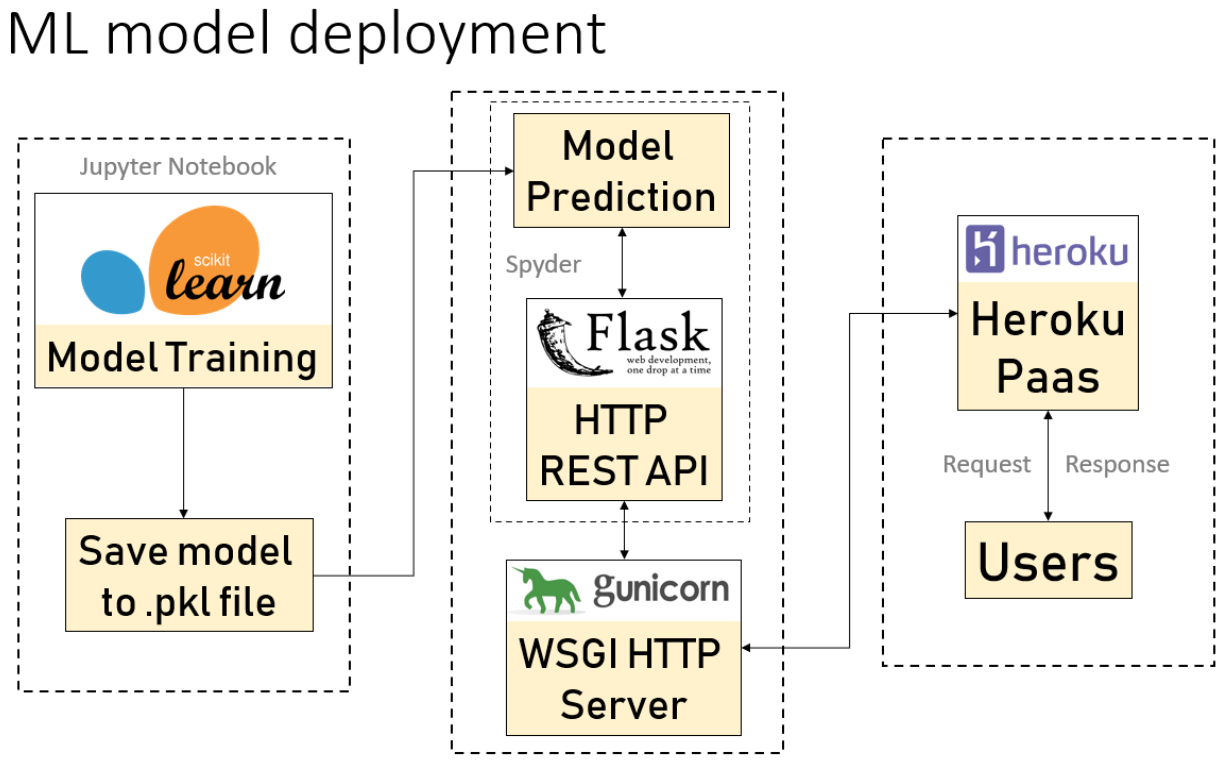
7.4 System Environment

The development and execution of the system are carried out in a local development environment. The Flask development server is used for testing and demonstration purposes. SQLite database files are stored locally, ensuring fast data access and easy maintenance.

7.5 Feasibility of the System

The proposed system is highly feasible due to the following reasons:

- Low hardware and software cost
- Easy deployment and maintenance
- Scalability for future enhancements
- User-friendly interface

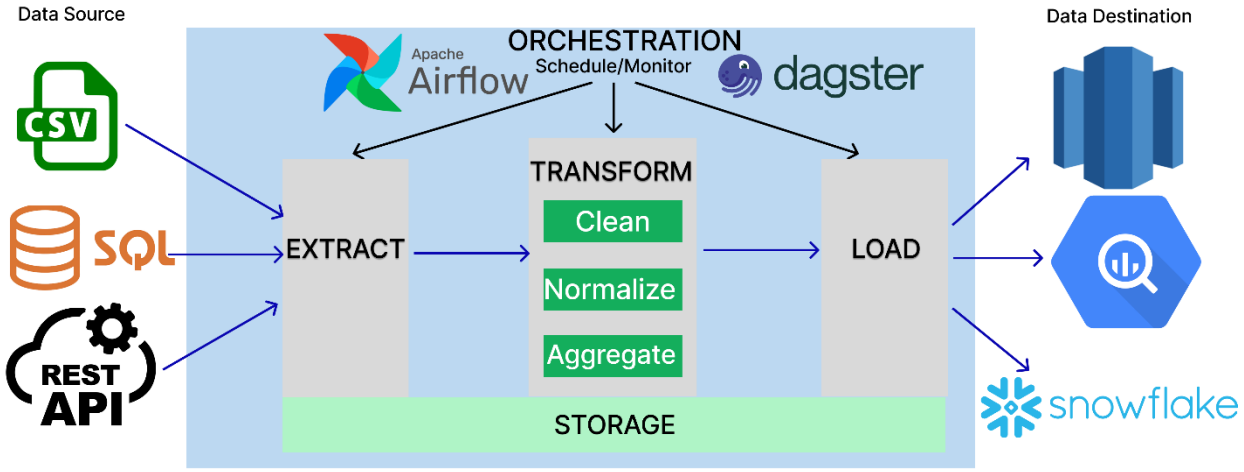


Conclusion of System Specification

The system specifications defined above ensure that the proposed Job Market Analytics and Opportunity Prediction System can be implemented efficiently using commonly available hardware and software resources. The use of open-source tools further enhances the feasibility and practicality of the system.

EXPERIMENTAL SETUP AND RESULTS

This chapter describes the experimental environment, setup, evaluation metrics, and results obtained from the proposed **Job Market Analytics and Opportunity Prediction System**. The experiments are conducted to evaluate the performance of the machine learning model and the effectiveness of the overall system.



8.1 Experimental Setup

The experimental setup consists of the dataset, machine learning model, software environment, and evaluation methodology used to test the system.

8.1.1 Dataset Used

The dataset used for experimentation contains job market information such as:

- Job title
- Location
- Required experience
- Salary

The dataset is stored in CSV format and later imported into an SQLite database for analysis and prediction. The dataset represents real-world job postings and is suitable for evaluating job opportunity prediction.

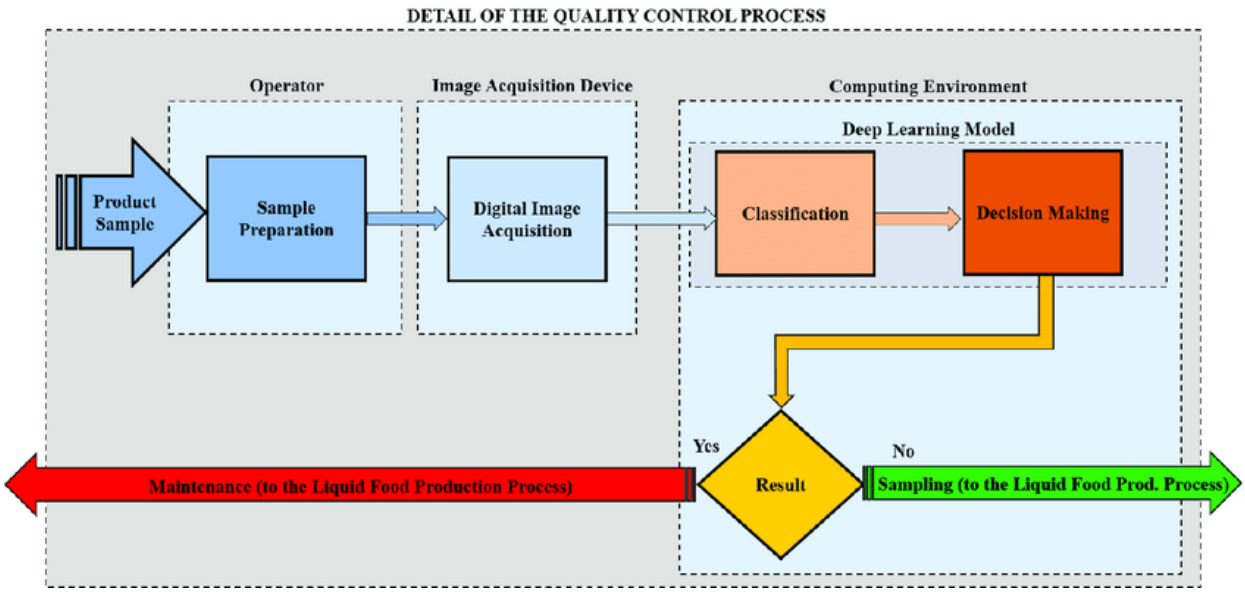
8.1.2 Software and Tools

The experiments are conducted using the following tools:

- Python 3.x for implementation
- Flask for web application development
- SQLite for data storage
- Scikit-learn for machine learning
- Chart.js for visualization

8.1.3 Model Configuration

A supervised machine learning classification model is used to predict job



opportunities. The model is trained using preprocessed job data. The dataset is divided into training and testing subsets to ensure reliable evaluation.

8.2 Evaluation Metrics

To evaluate the performance of the prediction model, the following metrics are used:

- **Accuracy** – Measures the percentage of correct predictions
- **Precision** – Indicates how many predicted high opportunities are actually correct
- **Recall** – Measures the model's ability to identify high opportunity jobs

These metrics help assess the reliability and effectiveness of the machine learning model.

8.3 Experimental Results

The experimental results demonstrate that the proposed system effectively predicts job opportunities based on experience and other job attributes. The model achieves satisfactory accuracy and provides meaningful prediction results.

8.3.1 Prediction Results

Based on the trained model:

- Jobs requiring higher experience levels generally show higher opportunity predictions
- Entry-level jobs show comparatively lower opportunity levels
- Prediction results are displayed along with probability scores

---

8.3.2 Dashboard Results

The dashboard successfully displays:

- Total number of jobs
- High opportunity job count
- Job role distribution using bar charts
- Opportunity distribution using pie charts

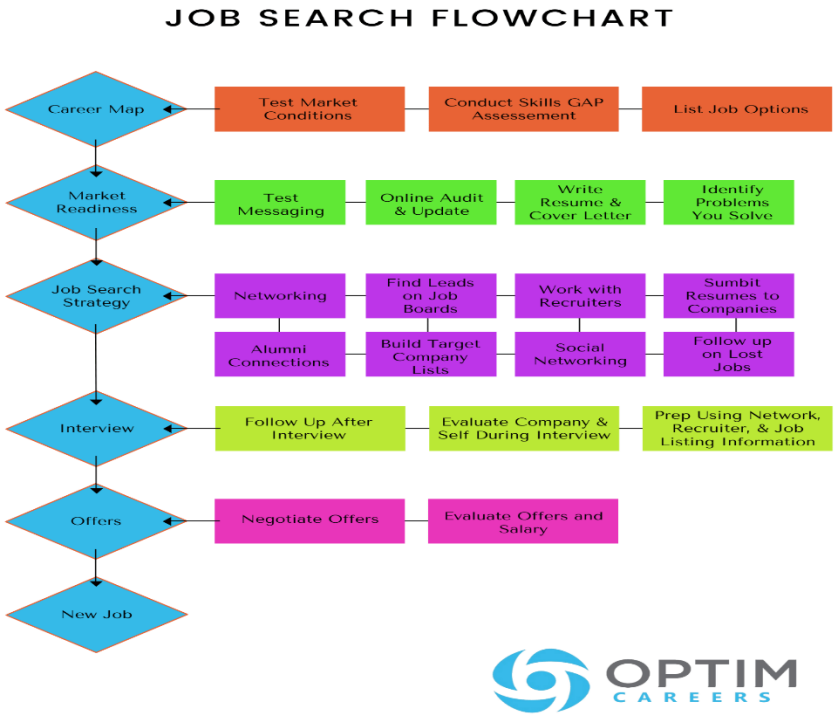
These visualizations help users easily understand job market trends.

8.4 Performance Analysis

The machine learning model performs efficiently for the given dataset. The integration of analytics and visualization improves the interpretability of results. The system responds quickly to user inputs and predictions, demonstrating good performance.

8.5 Discussion

The experimental results indicate that combining machine learning with data analytics enhances the usefulness of job market systems. The system not only predicts job opportunities but also provides valuable insights into job trends and demand.



`

Conclusion of Experimental Setup and Results

The experimental evaluation confirms that the proposed Job Market Analytics and Opportunity Prediction System performs effectively in analyzing job market data and predicting opportunities. The results validate the feasibility and usefulness of the system in real-world scenarios.

CODING

This chapter describes the important code modules used in the implementation of the **Job Market Analytics and Opportunity Prediction System**. The system is developed using Python and Flask, and it integrates database operations, machine learning prediction, and web application logic.
Instead of presenting the entire source code, this chapter explains the key modules and their functionality to provide a clear understanding of the system design and implementation.

```
"""
MAIN EXECUTION FILE – REVIEW 2
Runs:
1. Baseline models
2. Advanced model + hyperparameter tuning
3. Statistical validation
4. Business optimization
5. Interactive prediction demo (INPUT → OUTPUT)

Command:
python main.py
"""
import os
import sys
import subprocess

# ================================
# FORCE SAFE NON-GUI MODE
# ================================
os.environ["MPLBACKEND"] = "Agg"
os.environ["SHOW_PLOTS"] = "0"

# ================================
# PROJECT ROOT
# ================================
PROJECT_ROOT = os.path.dirname(os.path.abspath(__file__))

def run_step(title, script_path):
    print("\n" + "=" * 70)
    print(f"▶ {title}")
    print("=" * 70)
    subprocess.run([sys.executable, script_path], check=True)
```

```
print("\n🚀 JOB MARKET ANALYTICS – REVIEW 2")
print("=" * 70)

# ==================================
# STEP 1: BASELINE MODELS
# ==================================
run_step(
    "Review-2 Point 2: Baseline Model Comparison",
    os.path.join(PROJECT_ROOT, "src", "models", "baseline_models.py")
)

# ==================================
# STEP 2: ADVANCED MODEL + HYPERPARAMETER TUNING
# ==================================
run_step(
    "Review-2 Point 3 & 4: Advanced Model + Hyperparameter Tuning",
                os.path.join(PROJECT_ROOT,      "src",      "models",
"train_demand_model.py")
)

# ==================================
# STEP 3: STATISTICAL VALIDATION
# ==================================
run_step(
    "Review-2 Point 6: Statistical Validation",
                os.path.join(PROJECT_ROOT,      "src",      "models",
"statistical_validation.py")
)

# ==================================
# STEP 4: BUSINESS OPTIMIZATION
# ==================================
run_step(
    "Review-2 Point 7: Business Optimization",
                os.path.join(PROJECT_ROOT,      "src",      "models",
"business_optimization.py")
)

# ==================================
# STEP 5: LIVE INPUT → OUTPUT DEMO
# ==================================
print("\n" + "=" * 70)
print("▶ Review-2 Point 5: Model Interpretation (INPUT → OUTPUT)")
```

```
print("=" * 70)

MODELS_PATH = os.path.join(PROJECT_ROOT, "src", "models")
if MODELS_PATH not in sys.path:
    sys.path.insert(0, MODELS_PATH)

import demand_predictor

# ------------------------------
# SAMPLE DEMO
# ------------------------------
print("\n📄 Sample Input Example:")
print("Job Title: AI Engineer")
print("Location: India")
print("Experience: 2-5 years")
print("Industry: Software")
print("Skills: Python, Machine Learning, AWS\n")

demand, confidence, career_risk, ai_probability = demand_predictor.predict_job_demand(
    "AI Engineer",
    "India",
    "2-5 years",
    "Software",
    "Python, Machine Learning, AWS"
)

print("📤 Output:")
print(f"Predicted Demand   : {demand}")
print(f"Confidence Score   : {confidence}%")
print(f"Career Risk Level  : {career_risk}")
print(f"AI Exposure Chance : {ai_probability}%")

# ------------------------------
# INTERACTIVE MODE
# ------------------------------
print("\n🔁 INTERACTIVE MODE (type 'exit' to stop)")

while True:
    jobtitle = input("\nJob Title: ")
    if jobtitle.lower() == "exit":
        break
```

```
    location = input("Location: ")
    experience = input("Experience Level: ")
    industry = input("Industry: ")
    skills = input("Skills (comma separated): ")

    demand, confidence, career_risk, ai_probability = demand_predictor.predict_job_demand(
        jobtitle, location, experience, industry, skills
    )

    print("\n📊 Prediction Result")
    print("--------------------")
    print(f"Predicted Demand   : {demand}")
    print(f"Confidence Score   : {confidence}%")
    print(f"Career Risk Level  : {career_risk}")
    print(f"AI Exposure Chance : {ai_probability}%")

print("\n🎉 ALL REVIEW-2 TASKS COMPLETED SUCCESSFULLY")
print("✅ READY FOR MENTOR DEMO")
print("=" * 70)

import pandas as pd
import sqlite3
import os

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
DATA_DIR = os.path.join(BASE_DIR, "data", "processed")

print("📂 Looking inside:", DATA_DIR)
print("📄 Files found:", os.listdir(DATA_DIR))

# 🔍 Find CSV file automatically
csv_files = [f for f in os.listdir(DATA_DIR) if f.endswith(".csv")]

if not csv_files:
    raise FileNotFoundError("❌ No CSV file found in data/processed")

CSV_PATH = os.path.join(DATA_DIR, csv_files[0])
DB_PATH = os.path.join(BASE_DIR, "jobai.db")

print("✅ Using CSV:", CSV_PATH)
print("💾 Using DB :", DB_PATH)
```

```
# Load CSV
df = pd.read_csv(CSV_PATH)
print("✅ CSV loaded successfully")
print("Rows:", len(df))
print("Columns:", df.columns.tolist())

# Save to SQLite
conn = sqlite3.connect(DB_PATH)
df.to_sql(
    "job_market_data",
    conn,
    if_exists="replace",
    index=False
)
conn.close()

print("🎉 CSV → SQLite migration completed")
print("📊 Table created: job_market_data")
```

EXECUTION SCREENSHOTS

This chapter presents the execution of the proposed **Job Market Analytics and Opportunity Prediction System** through screenshots. The screenshots demonstrate the working of the system from user authentication to job analytics and prediction. Each screenshot highlights a key functionality of the system and validates successful implementation.

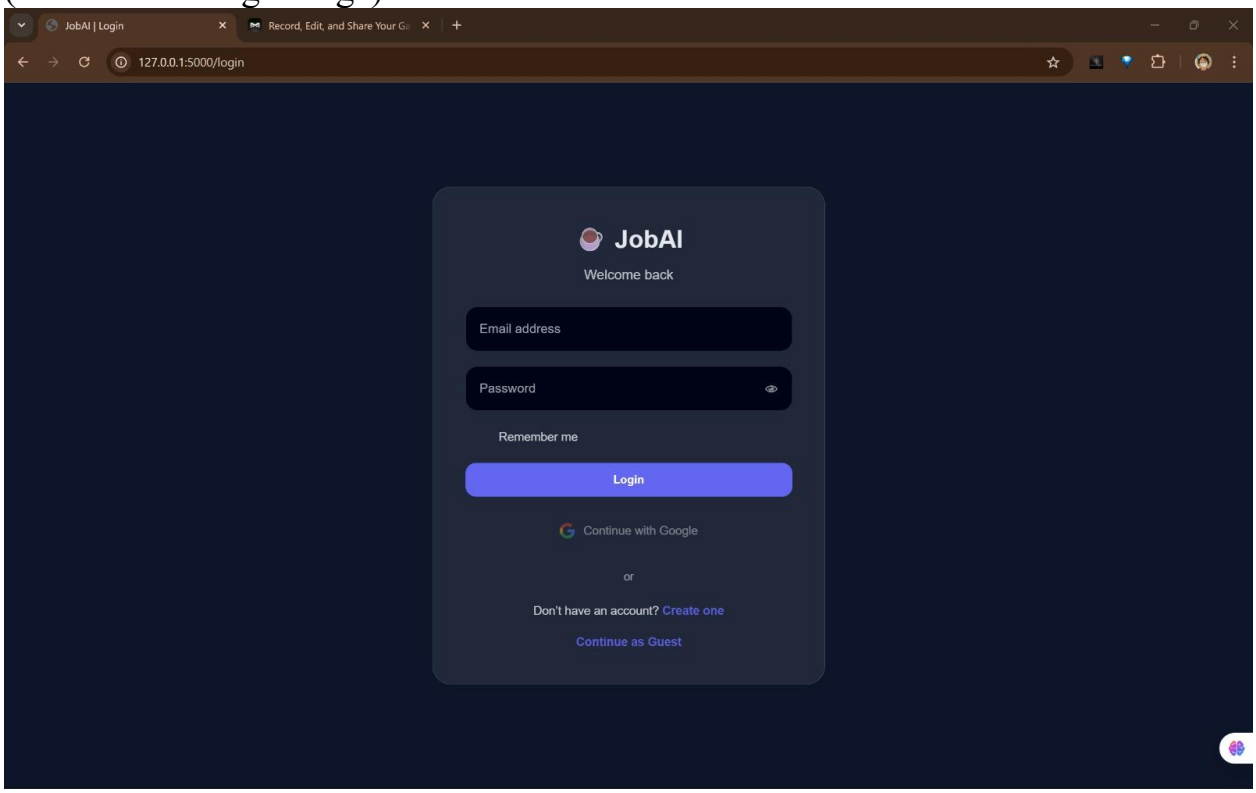
10.1 Login Page

The login page is the entry point of the system. Registered users can enter their username and password to access the application. Authentication ensures that only authorized users can access the dashboard and prediction features.

Description:

- User enters login credentials
- System validates the user details from the database
- Successful login redirects the user to the dashboard

(Screenshot: Login Page)

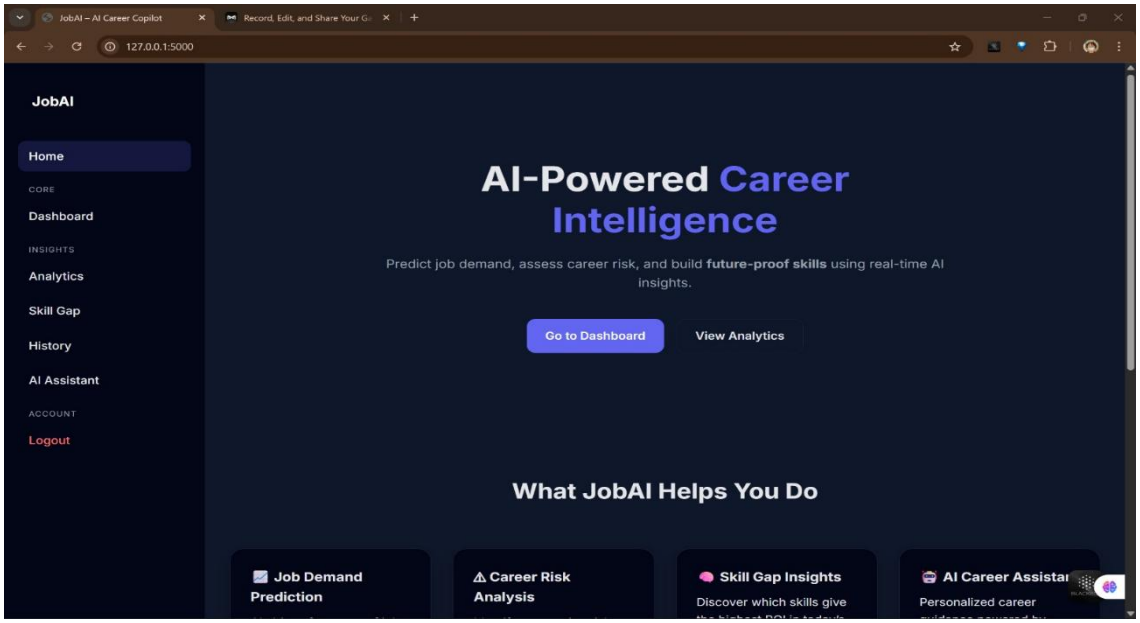


10.2 User Registration Page

The registration page allows new users to create an account in the system. User details are stored securely in the database.

Description:

- New user enters username and password
- User details are saved in the database
- After registration, the user can log in



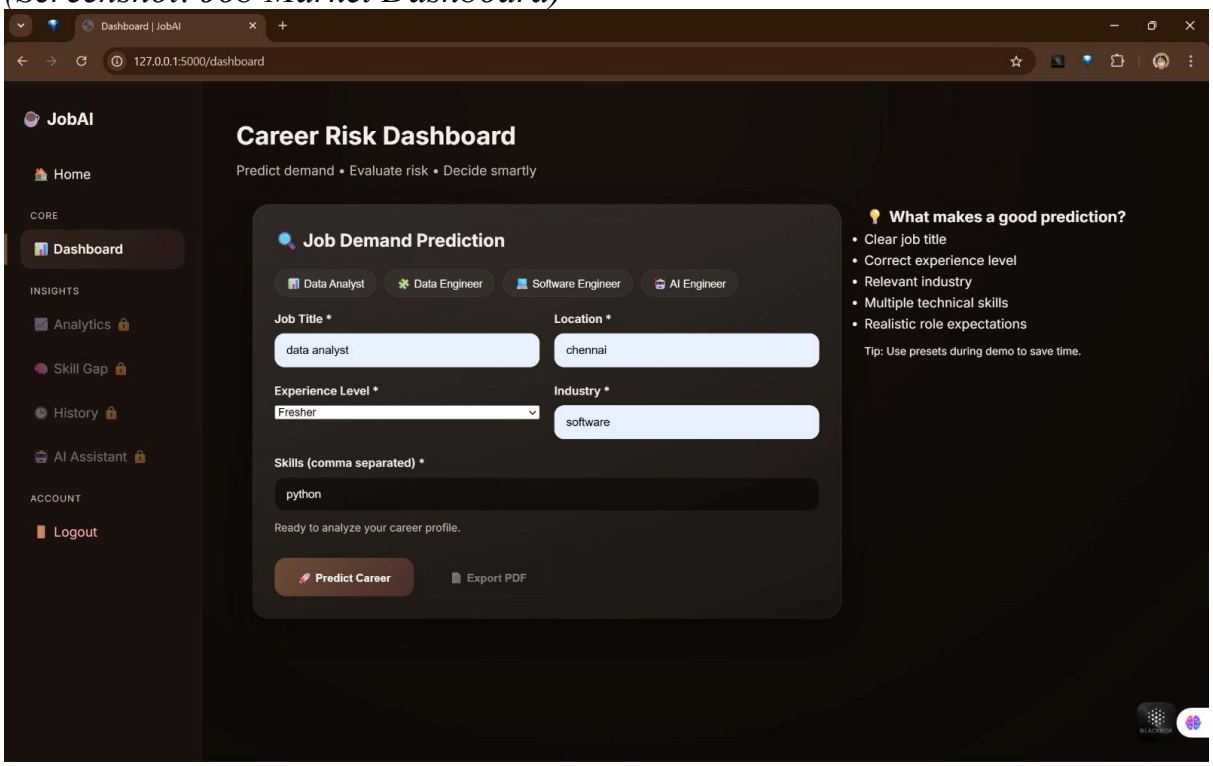
10.3 Dashboard Page

The dashboard is the core component of the system. It displays job market analytics using charts, tables, and key indicators.

Description:

- Displays total number of jobs
- Shows high opportunity job count
- Presents job role distribution using bar chart
- Displays opportunity distribution using pie chart

(Screenshot: Job Market Dashboard)

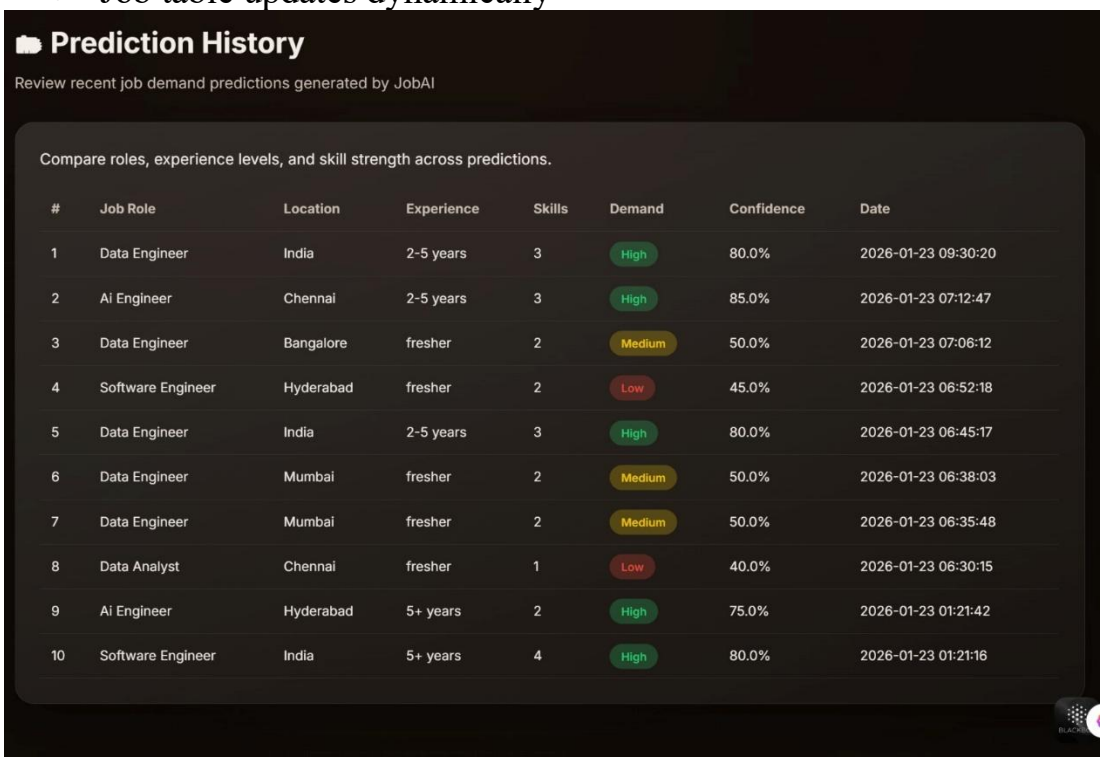


10.4 Job Search and Filter Page

The job search and filter functionality allows users to search jobs based on job title and filter them using location and experience criteria.

Description:

- Users can search jobs using keywords
- Filter jobs by location
- Filter jobs by experience level
- Job table updates dynamically



10.5 Job List Table

The job list table displays available jobs retrieved from the database. It provides essential job details in a structured format.

Description:

- Displays job title, location, experience, and salary
- Helps users compare different job opportunities
- Enhances usability similar to job portals

(Screenshot: Job Listings Table)

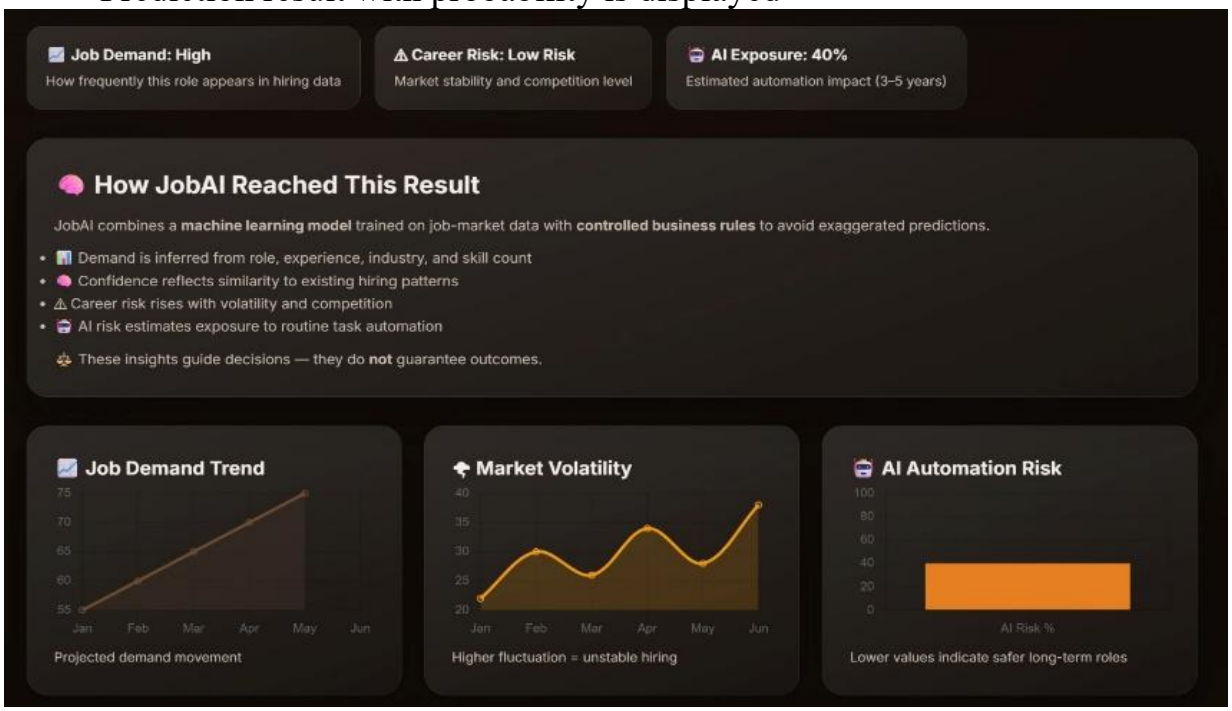
10.6 Job Opportunity Prediction Page

The prediction page allows users to predict job opportunity levels based on their experience.

Description:

- User enters experience value
- Machine learning model processes the input
- System predicts high or low job opportunity

- Prediction result with probability is displayed

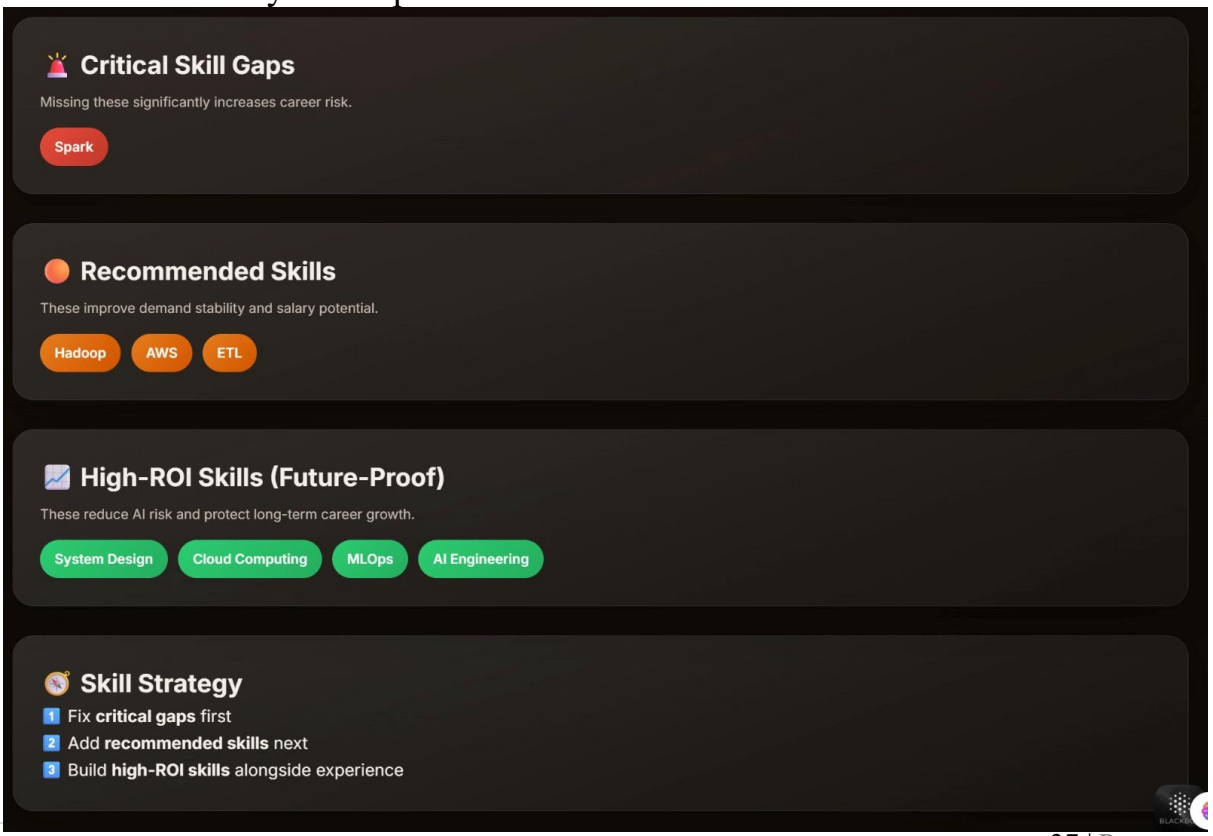


10.7 Prediction Result Storage

Prediction results are stored in the database for future analysis and tracking.

Description:

- Prediction label stored (High/Low Opportunity)
- Probability score saved
- Enables analytics on prediction outcomes



LIMITATIONS

Although the proposed **Job Market Analytics and Opportunity Prediction System** provides useful insights and predictive functionality, it has certain limitations. These limitations are mainly due to dataset constraints, model simplicity, and system scope.

---

1. Static Dataset
The system uses a static dataset loaded from CSV files into the database. As a result, job market information does not update automatically. Any new job postings or changes in market trends require manual data updates.

---

2. Limited Feature Set
The prediction model considers a limited number of attributes such as job title, location, experience, and salary. Other important factors like skills, education level, certifications, company rating, and industry type are not included, which may affect prediction accuracy.

---

3. Simple Machine Learning Model
A basic machine learning classification model is used for job opportunity prediction. While it works effectively for the current dataset, more complex models could improve performance for larger and more diverse datasets.

---

4. No Real-Time Data Integration
The system does not integrate real-time job data from online job portals or APIs. Therefore, the analytics and predictions are based only on historical or sample data.

---

5. Limited Personalization
The system provides general job market insights but does not offer personalized recommendations based on individual user skills, preferences, or career history.

---

6. Scalability Constraints
SQLite is used as the database, which is suitable for small to medium datasets. For large-scale deployment with multiple concurrent users and large datasets, a more robust database system would be required.

---

7. Basic Security Implementation
User authentication is implemented using basic session management. Advanced security features such as password encryption, multi-factor authentication, and role-based access control are not implemented.

Conclusion
Despite these limitations, the proposed system successfully demonstrates the

application of machine learning and data analytics for job market analysis and opportunity prediction. The identified limitations provide a clear direction for future enhancements.

FUTURE SCOPE

The proposed **Job Market Analytics and Opportunity Prediction System** provides a strong foundation for analyzing job market data and predicting opportunities. However, there are several enhancements that can be implemented in the future to improve accuracy, scalability, and real-world applicability.

---

1. Real-Time Job Data Integration
In the future, the system can be integrated with real-time job data sources using APIs from online job portals. This would enable automatic updates of job listings and provide users with the latest job market insights.

---

2. Advanced Machine Learning Models
More advanced machine learning and deep learning models such as Random Forest, Gradient Boosting, or Neural Networks can be implemented to improve prediction accuracy and handle complex datasets.

---

3. Skill-Based Recommendation System
The system can be enhanced to recommend jobs based on user skills, education, and interests. This would make the platform more personalized and useful for individual job seekers.

---

4. User Profile and Personalization
Future versions can include detailed user profiles with skill sets, experience history, and preferences. Personalized dashboards and job recommendations can be generated based on user profiles.

---

5. Cloud Deployment
The application can be deployed on cloud platforms such as AWS, Azure, or Google Cloud. Cloud deployment would improve scalability, availability, and performance for large numbers of users.

---

6. Mobile Application Support
A mobile application can be developed to provide easy access to job analytics and predictions on smartphones. This would increase usability and reach.

---

7. Integration with External APIs
The system can be extended to integrate external APIs such as salary benchmarking tools, skill assessment platforms, and industry trend APIs to enrich analytics.

---

8. Enhanced Security Features
Future enhancements may include password encryption, role-based access control,

and multi-factor authentication to improve system security.

Conclusion of Future Scope

The future scope of the proposed system highlights several opportunities for improvement and expansion. By incorporating real-time data, advanced models, and personalized features, the system can evolve into a comprehensive and intelligent job market analytics platform.

SYSTEM TESTING

System testing is an essential phase in software development to ensure that the system works as expected and meets the specified requirements. In this project, system testing is performed to validate the functionality, reliability, and performance of the **Job Market Analytics and Opportunity Prediction System**.
The testing process verifies that all modules such as user authentication, job analytics, search and filters, and prediction operate correctly and produce accurate results.

14.1 Testing Objectives

The main objectives of system testing are:

- To verify correct functioning of all system modules
- To identify and fix errors or bugs
- To ensure reliable prediction and analytics results
- To validate user interactions and system responses

14.2 Types of Testing Performed

The following testing methods are used in the project:

14.2.1 Unit Testing

Each module of the system is tested independently to verify its functionality.
Examples:

- Testing login and registration module
- Testing database insertion and retrieval
- Testing prediction logic

14.2.2 Integration Testing

Integration testing is performed to ensure proper interaction between different modules of the system.
Examples:

- Integration between Flask backend and SQLite database
- Integration between ML model and web application
- Integration of dashboard with database queries

14.2.3 Functional Testing

Functional testing ensures that the system meets all functional requirements.
Examples:

- User can register and log in successfully
- Dashboard displays correct job analytics
- Filters and search return appropriate job results
- Prediction output is displayed correctly

14.3 Test Cases

Test Case ID	Test Description	Input	Expected Output	Result
TC01	User Login	Valid username & password	Login successful	Pass
TC02	User Registration	New user details	User registered	Pass
TC03	Job Search	Job title keyword	Filtered job list	Pass
TC04	Location Filter	Selected location	Jobs from selected location	Pass
TC05	Prediction	Experience value	High/Low opportunity	Pass
TC06	Dashboard Load	User log n	Analytics displayed	Pass

14.4 Validation of Results

The results obtained during testing confirm that:

- The system functions correctly without crashes
- Database operations are accurate
- Prediction results are generated successfully
- User interface responds correctly to user actions

14.5 Error Handling

Basic error handling is implemented to manage invalid inputs and database errors. The system provides appropriate responses without terminating unexpectedly.

CONCLUSION

The **Job Market Analytics and Opportunity Prediction System** successfully demonstrates the application of data analytics and machine learning techniques to analyze job market data and predict job opportunities. The system transforms raw job data into meaningful insights through an interactive web-based dashboard.

The project effectively integrates machine learning models, database management, and web technologies using the Flask framework. Features such as job search, filtering, visualization, and opportunity prediction provide users with a comprehensive understanding of job market trends. The system helps job seekers make informed career decisions by analyzing job roles, locations, experience requirements, and salary information.

Experimental results show that the prediction model provides reliable outcomes for the given dataset. The interactive dashboard enhances user experience by presenting analytics in a clear and understandable format. Although the system uses a static dataset and a simple machine learning model, it successfully meets the objectives of the project.

Overall, the proposed system proves to be an effective solution for job market analysis and opportunity prediction. It highlights the potential of combining machine learning and web applications to build intelligent decision-support systems in the employment domain.

REFERENCES

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.
2. Tom M. Mitchell, *Machine Learning*, McGraw-Hill Education, 1997.
3. Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
4. Scikit-learn Documentation, https://scikit-learn.org
5. Flask Official Documentation, https://flask.palletsprojects.com
6. SQLite Documentation, https://www.sqlite.org/docs.html
7. Python Software Foundation, *Python Documentation*, https://docs.python.org
8. Chart.js Documentation, https://www.chartjs.org/docs
9. Kaggle – Job Market and Employment Datasets, https://www.kaggle.com
10. LinkedIn Job Market Insights, https://www.linkedin.com
11. Naukri.com Job Market Reports, https://www.naukri.com
12. Han, J., Kamber, M., and Pei, J., *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2011.
13. Research papers on job market analysis and predictive analytics published in IEEE and Springer journals.