

AI Resume Analyzer



ADEN CHUGHTAI

01-134212-012

ACCGAD BIN AAMIR

01-134221-093

Supervisor: Qazi Mohsin Ijaz

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

December 2025

Certificate

We accept the work contained in the report titled “AI Resume Analyzer”, written by Mr. Aden Chughtai AND Mr. Accgad Bin Aamir as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by ...:

Supervisor:

Internal Examiner:

External Examiner:

Project Coordinator:

Head of the Department:

December, 2025

Abstract

The AI-Based Resume Analyzer: a state-of-the-art web app that is aimed at making the job application much easier. This can be used to optimize resumes, which can be analyzed, refined and finally, strengthened by using artificial intelligence so that they are maximally compatible with Applicant Tracking Systems (ATS). The analyzer is aware of the fact that there is usually a discrepancy between the quality of the resume and the expectations of the industry, which is why the analyzer provides a detailed analysis, including grammatical correctness, keywords, and structural clarity, as well as industry-specific and relevant proposals.

The platform exploits an effective technological infrastructure. The user interface is created with React.js, and the backend and database services are handled with the use of Appwrite that offers user authentication, data storage, and server-side services without the necessity to have a traditional server configuration. The system incorporates AI functionality by use of the Gemini API in making smart content analysis and recommendation creation. The analyzer has a real time ATS scoring simulation. The simulation will enable the users to compare the performance of their resume with automated screening system and this will give one a good insight when it comes to improving his or her resume.

Early evidence suggests that there is a significant increase in the rate of job application success, with up to 40% improvement in the same, by implementing AI-based optimization techniques. Estimates reveal that the market in the globe that deals in resume writing services will have reached \$1.2 billion by 2027, which shows that there is a great potential of growth and increased use of this technology.

Acknowledgments

Allah Almighty has given us a chance to pursue this project and we would like to deeply thank Him. We believe His support and directions will open the path to the successful fulfillment of this project. We also owe a great debt to the great support of our supervisor, Mr. Qazi Mohsin Ijaz who has constantly supported, gave us a great vision and was always encouraging during the process of carrying out this project. His encouraging words and faith in our abilities have played a significant role in making us keep going and perform our best. It is a great honor to work under his guidance and we are appreciating his great mentorship. And finally, we want to thank our parents with all our hearts and souls that they prayed and also supported us. They have been encouraging and strong in their blessing and supported us to move on in this project and further.

ADEN CHUGHTAI
Bahria University
E-8, Islamabad, Pakistan

ACCGAD BIN AAMIR
Bahria University
E-8, Islamabad, Pakistan

December 2025

Contents

Abstract	i
1 Introduction	1
1.1 Introduction	1
1.2 Problem Description	1
1.3 Project Overview	2
1.4 Project Objectives	2
1.5 Project Scope	3
1.5.1 Frontend Development	3
1.5.2 Backend Services	3
1.5.3 AI/ML Integration	3
1.5.4 Database Management	3
1.5.5 ATS Simulation	4
1.6 Solution Application Areas	4
2 Literature Review	5
2.1 Existing Resume Analysis Solutions	5
2.1.1 Commercial Resume Analysis Platforms	5
2.1.2 Academic Research in Resume Optimization	6
2.2 AI and Machine Learning in Human Resources	6
2.2.1 Natural Language Processing Applications	6
2.2.2 Machine Learning in Recruitment Processes	7
3 Requirement Specifications	8
3.1 Introduction	8
3.2 Functional Requirements	8
3.2.1 Core Analysis Features	8
3.2.2 User Interface and Experience Features	9
3.3 Non-Functional Requirements	10
3.3.1 Scalability	10
3.3.2 Reliability	10
3.3.3 Security	10
3.3.4 Usability	11
3.4 Hardware Requirements	11
3.4.1 Client-Side Requirements	11
3.4.2 Server-Side Infrastructure	11
3.5 Software Tools and Libraries	12

3.5.1	Frontend Dependencies	12
3.5.2	Backend Dependencies	12
3.6	Architectural Overview	13
3.6.1	Frontend Layer	13
3.6.2	Backend and Middleware Layer(Appwrite Platform)	13
3.6.3	AI Processing Layer	13
3.6.4	System Deployment and Scalability	14
3.6.5	Security and Data Flow	14
3.7	System Assumptions	14
3.7.1	User Behavior Assumptions:	14
3.7.2	Technical Environment Assumptions:	14
4	Design	15
4.1	Introduction	15
4.2	System Block Diagram	15
4.2.1	Presentation Layer	15
4.2.2	Backend Service Layer	16
4.2.3	AI Processing Layer	16
4.2.4	Data Storage Layer	16
4.3	Data Flow Diagram (DFD)	17
4.3.1	Level 0 DFD (Context Diagram)	17
4.3.2	Level 1 DFD (System Overview)	17
4.3.3	Level 2 DFD (Detailed Process Flows)	19
4.4	Design Considerations	20
4.4.1	Performance Optimization	20
4.4.2	Scalability Planning	20
4.4.3	Security Implementation	20
4.4.4	Maintainability Factors	20
4.5	System Flowchart	20
4.5.1	User Registration and Authentication Flow	20
4.5.2	Resume Analysis Workflow	20
4.5.3	Recommendation Implementation Process	21
4.5.4	System Flow Overview Explanation	21
4.5.5	Backend Processing Details	21
4.6	UML Sequence Diagram	23
4.6.1	User Authentication Sequence	23
4.6.2	Resume Analysis Sequence	23
4.7	Use Case Diagram	26
4.7.1	Primary Actors	26
4.7.2	Use Cases for Job Seeker	26
4.7.3	Use Cases for Administrator	26
4.7.4	System Use Cases	26
4.8	UML Component Diagram	27
4.8.1	Frontend Components	27
4.8.2	Backend Components	28
4.8.3	Integration Components	28

5	System Implementation	30
5.1	Introduction	30
5.2	Implementation Environment	30
5.3	Frontend Implementation	31
5.3.1	User Interface Design	31
5.3.2	Routing and State Management	37
5.4	Backend Implementation	37
5.4.1	Authentication and Authorization	37
5.4.2	Cloud Functions for Business Logic	37
5.5	AI Model Integration	38
5.5.1	Data Processing Workflow	38
5.5.2	ATS Simulation	38
5.6	Database and Storage Configuration	38
5.7	System Deployment	39
5.7.1	Key Deployment Steps	39
6	System Testing and Evaluation	40
6.1	Testing Overview	40
6.2	Functional Testing	41
6.3	Integration Testing	41
6.4	Performance Testing	41
6.5	Security Validation	41
6.6	Usability and Error Messaging	42
6.7	AI Evaluation and Consistency	42
7	Conclusions	50
	References	53
A	User Manual	55
A.1	Introduction	55
A.2	System Overview	55
A.3	System Requirements	55
A.3.1	Hardware Requirements	55
A.3.2	Software Requirements	56
A.4	User Roles	56
A.5	Accessing the System	56
A.6	User Registration	56
A.7	User Login	57
A.8	Dashboard Overview	57
A.9	Resume Upload	57
A.9.1	Supported Formats	57
A.9.2	Upload Procedure	57
A.10	Resume Analysis	57
A.11	Viewing Analysis Results	58
A.12	Implementing Recommendations	58
A.13	Downloading Reports	58
A.14	Account Management	58

CONTENTS

A.15 Error Handling and Feedback	58
A.16 Security and Privacy	58
A.17 Logout Procedure	59
A.18 System Limitations	59
A.19 Support and Troubleshooting	59
Index	55

List of Figures

4.1	System Block Diagram	16
4.2	Level 0 Data Flow Diagram (Context Diagram)	17
4.3	Level 1 Data Flow Diagram (System Overview)	18
4.4	Level 2 Data Flow Diagram (Detailed Process Flows)	19
4.5	System Flowchart	22
4.6	User Authentication UML Sequence Diagram	24
4.7	Resume Analysis UML Sequence Diagram	25
4.8	Use Case Diagram	27
4.9	UML Component Diagram	29
5.1	Home Page (1)	31
5.2	Home Page (2)	32
5.3	Home Page (3)	32
5.4	Sign Up	33
5.5	Sign In	33
5.6	Upload Page	34
5.7	Result (1)	34
5.8	Result (2)	35
5.9	Result (3)	35
5.10	History Page	36
5.11	About Page (1)	36
5.12	About Page (2)	37

List of Tables

6.1	Test Case for User Registration (ID: TC-1)	42
6.2	Test Case for Email Verification (ID: TC-2)	42
6.3	Test Case for User Login (Valid) (ID: TC-3)	43
6.4	Test Case for User Login (Invalid) (ID: TC-4)	43
6.5	Test Case for Resume Upload (Valid Formats) (ID: TC-5)	43
6.6	Test Case for Resume Upload (Invalid/Corrupted File) (ID: TC-6)	44
6.7	Test Case for Resume Parsing Accuracy (ID: TC-7)	44
6.8	Test Case for Analysis Report Generation (ID: TC-8)	44
6.9	Test Case for Single Resume Processing Time (ID: TC-9)	45
6.10	Test Case for Concurrent Uploads (Load) (ID: TC-10)	45
6.11	Test Case for API Latency Monitoring (ID: TC-11)	45
6.12	Test Case for Stress Test Until Failure (ID: TC-12)	46
6.13	Test Case for Access Control Enforcement (ID: TC-13)	46
6.14	Test Case for Session Expiry and Logout (ID: TC-14)	46
6.15	Test Case for Dashboard Clarity and Navigation (ID: TC-15)	47
6.16	Test Case for Progress & Error Messaging (ID: TC-16)	47
6.17	Test Case for Appwrite Storage + Database Consistency (ID: TC-17)	47
6.18	Test Case for Gemini API Error Handling (ID: TC-18)	48
6.19	Test Case for Cross-Browser Functionality (ID: TC-19)	48
6.20	Test Case for Responsive Layout on Mobile (ID: TC-20)	48
6.21	Test Case for AI Output Consistency (Repeatability) (ID: TC-21)	49
6.22	Test Case for AI Relevance and Precision (ID: TC-22)	49
7.1	Core Feature Comparison Between Our System and Existing Platforms	52
7.2	Features Present in Competitor Platforms but Not in Our System	52

Acronyms and Abbreviations

AI	Artificial Intelligence
ATS	Applicant Tracking System
API	Application Programming Interface
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
ML	Machine Learning
NLP	Natural Language Processing
PDF	Portable Document Format
REST	Representational State Transfer
UI	User Interface
UML	Unified Modeling Language
UX	User Experience

Chapter 1

Introduction

1.1 Introduction

The better and more well structured your resume, the higher your chances of getting a job in the modern job market. Most companies currently rely on special computer software known as Applicant Tracking System to peruse through resumes. Almost all types of businesses use these systems, and they complicate the functionality of the traditional resume writing techniques. It has been found out that approximately 98 percent of the largest corporations, such as the Fortune 500, adopt ATS to view resumes. Each job advert receives approximately 250 resumes, on average [1].

The issue is ensuring that your resume is presentable enough to the eyes of the people reading it as well as the computer system. Before a human will even consider reading through a resume of a skilled individual, many of these skilled individuals are overlooked due to the resume having failed to use the right keywords, the resume might have been formatted improperly, or might not have been done according to the rules that the ATS follows. This may prevent the talented individuals to obtain the desired job [1].

1.2 Problem Description

The job search and the recruitment process are a major challenge to job seekers and organizations. Applicants are finding it difficult to prepare resumes that meet industry standards and the expectations of Applicant Tracking System (ATS) which tends to sieve out resumes because of formatting, lack of key words or skills are not well presented. In the absence of a clear guidance, job seekers do not have a good understanding of what their resume is lacking including relevant terminology, skills deficiency or proper structure. This

complicates and wastes time trying to optimize it. Consequently, successful candidates will not even make it through the first automated screening process.

On the part of the recruiters, it is time consuming to sift through the resumes in bulk, and it is also subject to discrepancy. The process of matching candidates to job descriptions is quite manual and it becomes even harder to ensure fairness in all applicants. To recruiters, they require the tools that are capable of assisting in measuring the relevancy of resumes, determining the necessary skills, and facilitate the initial screening process, ultimately, easing the workload but enhancing the quality and efficiency of the candidate screening.

1.3 Project Overview

The AI-Based Resume Analyzer is a full-scale tool, which assists job seekers to connect to the current hiring practices better. It is an online service that operates based on the latest artificial intelligence and machine learning to provide in-depth feedback on the resume, advising of any necessary improvements, and a rating of the efficacy of a resume with automated systems [2].

It does not merely look at grammatical mistakes. It examines at industry-related information, ensures that the appropriate keywords are employed and recommends alterations that suit various career domains. Through the simulation of actual automated screening systems, the platform provides users with concise recommendations on how their resume can be distinguished during the hiring process [2]. It also enhances efficiency in recruitment because it automates the process of matching resumes with job descriptions and this means that less human effort is required and more job matches are achieved.

1.4 Project Objectives

The primary objectives of this project include:

- **Automated Resume Analysis:** Develop intelligent software to scrutinize resumes in terms of grammar, the level of organization and professional appearance.
- **ATS Optimization:** Have an AI-based assessment based on the Gemini model that uses resumes as input data and gives the correct compatibility score and improvement recommendations in the manner of Applicant Tracking Systems (ATS).
- **Industry-Specific Customization:** Build personalised recommendations based on the requirements of various industries and relevant trends at the moment with the use of machine learning.
- **Enrichment of User Experiences:** Develop user-friendly designs that enable individuals of various levels of skills to make their resumes better.

- **Commercial Viability:** Develop a platform that can expand and can manage various methods of monetizing by subscribing and providing special functionality to premium users.
- **Performance Optimization:** Ensure that they are fast and perform well even in situations where lots of other individuals are also using the service.

1.5 Project Scope

The project scope involves the design of full stack web application comprising the following major components:

1.5.1 Frontend Development

Developing a responsive and simple interface based on React.js that can be used both on desktop and mobile platforms. The interface allows uploading of resumes through drag and drop, displays results of analysis in real-time, and provides interactive suggestions.

1.5.2 Backend Services

Application: One of the applications is the use of backend functionality that uses Appwrite, a framework that handles authentication, database operations, file storage, and server-side code. Appwrite is a new technology that does not require a dedicated server infrastructure (such as Node.js and Express.js, among others) instead, it offers all-inclusive user management, data processing, and cloud functions APIs [3]. Such a solution guarantees high availability, safety of data usage, and scalability and minimizes the maintenance of the backend by humans.

1.5.3 AI/ML Integration

Adding the functionality of AI, the Gemini API that works with the analysis of resume content, the level of key words and the generation of recommendations. This API allows understanding natural-language and makes smart suggestions to improve a resume. The system takes text uploaded on a resume and forwards this text to the Gemini model to be analyzed and provides the user with a structured feedback.

1.5.4 Database Management

Using the in-built database and storage services of Appwrite to store user data, result of analysis of resumes and associated metadata. This avoids the use of different database systems such as the MongoDB but also provides real-time data synchronization, scalability, and safe storage [3].

1.5.5 ATS Simulation

The adoption of an AI-based simulator that assesses the resume according to the way Applicant Tracking Systems (ATS) are normally handled and rated. Gemini model examines the content, structure and keyword relevance to determine approximate ATS compatibility and offer specific improvement recommendations.

1.6 Solution Application Areas

The AI-Based Resume Analyzer could be applied to several spheres of the professional development ecosystem:

- **Personal Job Seekers:** New graduates, career-changers and professionals who wish to clean up their resumes to suit some jobs or careers. The site offers personalized recommendations that enable receiving more calls to interviews and to make the job-finding more effective [2].
- **Career Services Centers:** The platform can also be used by career services centers in universities and professional organizations to provide superior services to their students and members. The system enables the analysis of huge quantities of information and generation of institution reports [4].
- **Recruitment Agencies:** Professional recruiting agencies can check and optimize their job applications via the platform before submitting them to their clients, which will increase the likelihood of successful placements and makes clients happier [5].
- **Corporate HR Departments:** The companies can utilize the platform to provide career growth tools to employees interested in progressing or swapping jobs in the company [1].

Chapter 2

Literature Review

2.1 Existing Resume Analysis Solutions

2.1.1 Commercial Resume Analysis Platforms

There are some popular tools used to analyze the resumes in the job market, which have their own mechanism of operation and a few weaknesses [2]. Jobscan is among the best tools, which primarily assists in the optimization of ATS (Applicant Tracking System). It compares resumes to job descriptions and scores depending on the match of keywords. Such applications as Jobscan and Resume.io are based on the principle of matching keywords to enhance resumes, yet they are mostly based on certain general rules, and do not profoundly comprehend the sense of the words [6]. In 2018, a study by Luo et al. proposed a neural network-based type of smart system called ResumeNet to assess resumes. Such a system is able to examine the text and the way it is formatted to ensure quality, which means that it is not only the number of keywords used that will determine the quality of a resume [4].

On the same note, Li et al. (2020) investigated the applicability of transformer-based models to align resumes with job descriptions. They demonstrated that conventional template-based systems are limited and they proposed that AI-driven systems with contextual understanding would be superior [7]. ResumeGenius and other platforms of this kind have templates and certain fundamental tools in order to enhance resumes, but they lack sophisticated AI scanning to render a complete and efficient resume upgrade [2].

Zety and Resume.com have included certain simple AI functions, yet the analysis is restricted to grammar checking and providing basic suggestions of keywords. These platforms do not provide any specific industry customization or do not use a high level of natural language processing and that is a significant missing component of what is currently available.

2.1.2 Academic Research in Resume Optimization

Scholarly studies have achieved enormous gains regarding the production of resume reviewing mechanisms and enhancing the method of job application judgment. The authors Li, Fisher, and their team (2020) employed BERT-based transformer models to score the level of skill of the resume and its suitability in respect to a job description. They were in a position to classify with a high accuracy of above 70%. Their analysis revealed that contextual embedding may prove to be quite helpful in the assessment of professional papers [7]. Luo and others (2018) also demonstrated that deep learning could be a strong method of forecasting the resume quality, and their models behaved on big volumes of resume data [4]. Moreover, the issue of bias and equity has become more and more heated regarding AI-based resume screening. Li, Lu, and Li (2023) indicated that machine learning systems tend to incorporate bias in society, particularly, their origins, and this is why there are interventions to help make AI recruitment less biased [8].

Resume analysis based on Natural Language Processing tools has yielded satisfactory results [6]. Such techniques as finding essential words, major subjects, and matching words by definition assist systems in comprehending unsystematized information of text. Indicatively, transformer based models have significantly enhanced the ability of the system to test relevance of a resume as well as the quality by learning the relationship of words within a context [7][8]. Such enhancements have ensured that there are more efficient, accurate, and fair methods of screening resumes.

2.2 AI and Machine Learning in Human Resources

2.2.1 Natural Language Processing Applications

Application of Natural Language Processing in human resources has altered the manner in which hiring is done. New NLP is capable of a better analysis of the content, something that is not possible by simply searching keywords to grasp the meaning, emotions, and the communicative effectiveness of an individual on a professional level [6].

Transformer models, such as BERT and GPT, have recently improved and altered the text analysis process significantly. In 2023, Jagwani and their team developed a resume scoring method that used Latent Dirichlet Allocation and entity recognition with SpaCy and they scored over 80 percent on a small number of resume features [6]. In 2024, Saatçı and their team also developed the system that is capable of screening candidates more simply and with less time, as it employs NLP and Jaccard similarity to compare the skills of the candidates to the job requirements [9].

Semantic analysis is useful to make sense of a resume by identifying links between skills, work experience, and achievements. It becomes simpler to align the candidates with

the appropriate jobs and the process of hiring employees becomes smooth with both the job seekers and the recruiter [5].

In the given project, resumes are analyzed and evaluated by the use of the Gemini model that is based on the modern large language model (LLM) architectures. In contrast to conventional transformer-based systems, which need a lot of fine-tuning, Gemini offers strong contextual knowledge using API integration, which can accurately evaluate resume content and generate recommendations with little model management [10].

2.2.2 Machine Learning in Recruitment Processes

Machine learning finds application in the hiring processes beyond resume scanning. It is also useful in matching the candidates with the correct jobs, it also minimizes the unfair prejudices and also uses data to assess the potential of an individual doing a good job. These systems are informed by the past hiring results where individuals performed well and they get to know what qualities are associated with good job performance and how well an individual will fit in the culture of a company [4].

The use of AI in hiring has gained a large subject of ethical concerns. In case the information involved in training such systems is biased, it can proceed to unfairly treat individuals hence there is an urgent need to ensure that any AI tool used in HR is fair and clear [1]. Recent regulations and recommendations point to the importance of explainability of AI in hiring in order to make people aware of the decision-making process [5].

Combining machine learning methods has been more effective in multifaceted HR activities. Such approaches involve the reading of textual materials, evaluation of structure, and transformation of certain industries to be better. This assists in designing tools that process numerous components of the enhancement of resumes simultaneously [4].

Chapter 3

Requirement Specifications

3.1 Introduction

The technical requirements and specifications of the AI-Based Resume Analyzer are all the essential features and performance requirements that will enable the software to be robust and capable of managing growth as well as simplifying the use of the software by the users. These requirements have been collected based on the thorough study of the market, the knowledge of what the users require and verification of what can technically be done in the process of planning the project.

The system is constructed to serve an individual and a large scale company with various levels of service and features, which could be configured to their requirements [2]. It is all about ensuring that things are secure, fast, and can be expanded at any point necessary and ensure that the user experience is easy to navigate on all the devices where it is being accessed.

3.2 Functional Requirements

3.2.1 Core Analysis Features

FR-1: Resume Upload and Parsing

- The system will allow uploading resumes in PDF format.
- The limit of file size upload is 10MB.
- Automatic text extraction and formatting context.
- Corrupted/ unsupported file format error handling.
- Status of upload and processing advancement.

FR-2: Content Analysis powered by Artificial Intelligence

- Spelling and grammar checking through high-order NLP algorithms.
- Language testing and revising recommendations.
- Evaluation of the content structure such as section structure and flow.
- Analysis and recommendations of achievement quantification.
- Optimization of industry specific terms.

FR-3: ATS Compatibility Scoring

- ATS behavior simulation with Gemini model to check resume formatting, structure and alignment by keywords based on how major ATS systems (e.g., Workday, Taleo, Green- house) process documents [10].
- Live compatibility score with elaborate computer-generated descriptions.
- Recommendations of format optimization.
- Keywords density and optimization recommendations.
- Standardization of section header and layout to increase ATS readability.

FR-4: Segmentation by Industry

- Profiles of pre-configured industry analysis in 20+ industries.
- Dynamic job market-based keyword libraries.
- Formatting and best practices in the industry.
- Industry trend scoring of skills.
- Appropriateness of certification and education.

3.2.2 User Interface and Experience Features

FR-5: Interactive Dashboard

- Summary of analysis findings in the form of visual indicators.
- Advancement monitoring of recommendations put in place.
- Comparison of historical analysis.
- Export features of reports and refined resumes.

- Mobile-responsive design.

FR-6: Recommendation Implementation

- Detailed instructions on improvements.
- On the job description and job title, the recommendation priorities can be customized.

3.3 Non-Functional Requirements

3.3.1 Scalability

The system is designed for scalability by using Appwrite’s cloud-based backend architecture which automatically manages increased workloads without manual server scaling. Appwrite handles user management, databases, and storage through distributed infrastructure, ensuring consistent performance even under high user activity. At the service level, auto-scaling and resource allocation are managed. This reduces operational overhead and maintains cost efficiency [3].

3.3.2 Reliability

It is based on the managed services of Appwrite in order to ensure high reliability and uptime (targeting 99.9% availability). The data replication and backup processes are automatic to ensure that the user data and the configuration settings are secure and can be restored. The backend environment is designed to include continuous health monitoring and failover management so that no one is left without access or to the environment and that it remains consistent even when updating or the faults occur unexpectedly.

3.3.3 Security

Data security policies include end-to-end encryption of all information and files exchanged by users. The personal information handling is guided by regulations such as GDPR and CCPA among other significant privacy regulations [8]. This is achieved by ensuring that there are good data protection policies and ensuring that users consent to the use of their information [11].

Users can be authenticated in more than one way, such as a password and a code provided to their phone, and they retain sessions. The API security is a measure that assists in data security by restricting the number of requests to execute within a specific period of time, ensuring that all the information received as inputs is secure, and preventing the use of malicious code in database queries [11]. Constant inspections and tests by the security officers assist in identifying and repairing vulnerabilities before attackers can exploit them.

3.3.4 Usability

The interface is user-friendly as it is created based on the accessibility standards (WCAG 2.1) to assist individuals with disabilities in its easy usage [12]. Its menus are easy to follow and have clear layouts that enable a new user to learn fast and it also provides higher level to those users who are advanced.

The average resumes require a maximum of 30 seconds to analyze. As the process is taking place, indicators of clear progress appear to indicate what is going on. In case of a mistake, the message must provide useful actions to correct the situation, and not complex technicalization [13].

3.4 Hardware Requirements

3.4.1 Client-Side Requirements

Minimum System Requirements:

- Modern web browser (Chrome 90+, Firefox 88+, Safari 14+, Edge 90+).
- 2GB RAM to support the browsers.
- Good internet connectivity (not less than 5 Mbps to upload files).
- Display resolution of 1024x768 or higher.

Recommended Specifications:

- 4GB or more RAM to improve the performance with large files.
- Fast internet connection (20 or more Mbps) for fast processing.
- 1920x1080 screen resolution to provide the best user experience.

3.4.2 Server-Side Infrastructure

Production Environment:

- Backend and database services are handled using Appwrite Cloud or a self-hosted instance of Appwrite, which does not require the conventional server provisioning.
- Appwrite supports processing of concurrent requests, authentication, and data storage through a managed infrastructure.
- It only needs a stable network connection and access to the standard API to integrate with the Gemini API, and compute without the use of GPUs [10].
- Cloud-based storage is automatically scalable according to usage, with no manual setup and no performance or reliability concerns.

3.5 Software Tools and Libraries

3.5.1 Frontend Dependencies

Core Framework and Libraries:

- React.js 19 for component-based UI development.
- React Router v7 for client-side routing and navigation.

Development and Build Tools:

- Webpack 5.0+ for module bundling and optimization.
- Babel for JavaScript transpilation and compatibility.
- ESLint and Prettier for code quality and formatting.
- Jest and React Testing Library for comprehensive testing.

3.5.2 Backend Dependencies

Backend Platform:

- Appwrite: Provides authentication, database management, file storage, and cloud functions, which replaces Node.js and Express.js [14].
- Appwrite SDK (JavaScript/TypeScript): Has the ability to communicate safely between the frontend and the backend.
- Appwrite Functions: Used for custom logic, API handling, or Gemini API integration [14].

Database and Storage:

- Appwrite Database: This is where the user information and analysis results will be stored in a central database.
- Appwrite Storage: Manages the upload of files, resume documents and generated reports in a safe manner.

AI/ML Integration:

- Gemini API: It is used to carry out the process of natural language processing, resume analysis, and generation of recommendations.

3.6 Architectural Overview

The system is of a cloud integrated design service oriented and based on scalability, modularity and maintenance conformity. It is a combination of a React.js frontend, an Appwrite backend platform, and external Gemini AI model integration to provide real-time resume analysis and feedback.

3.6.1 Frontend Layer

The frontend is the React.js layer that acts as the presentation layer and whose role is to deal with all the user interaction. It has an easy-to-use interface in which people can upload their resumes, analyze the results, and implement the recommendations [13]. The frontend uses its JavaScript SDK to interact with APIs of Appwrite in order to perform safe operations like authentication, database calls, and file uploads [14].

Appwrite has an in-built subscription and event system that enables real-time updates of analysis results and updates without a user having to upgrade the page.

3.6.2 Backend and Middleware Layer(Appwrite Platform)

The Appwrite platform serves as the central back-office to the system, which eliminates the need to utilize a standard server infrastructure. It offers some of its major services:

- **Authentication Service:** Authenticated registration of users, login, and management of the sessions with default security policies like email validation and access control using JWT.
- **Database Service:** Structured and semi structured data (user information, analysis metadata, and system logs) is stored [14].
- **Storage Service:** Deals with uploaded files (resumes in PDF format), analysis reports and processed text outputs.
- **Cloud Functions:** Serve as middleware to carry out asynchronous operations such as making requests to the Gemini API to analyze the input in AI, checking uploaded files, and running background services.
- **API Gateway:** Regulates the flow of information between the frontend and the backend and ensures the safe access, role-based authority, and rate limiting.

3.6.3 AI Processing Layer

The analysis of the resume is done through the Gemini model, which is retrieved through the Gemini API. On uploading a CV or a resume, the text is scraped and sent by an

Appwrite Function to the Gemini API. The model works with the content to analyze it with state-of-the-art natural-language understanding systems to analyze grammar, formatting, relevance of keywords and general industry standards of correctness [10]. The AI then gives out structured responses which are saved in the Appwrite database and presented to the user on the frontend dashboard.

3.6.4 System Deployment and Scalability

It runs on Vercel, with the frontend being built in React.js and the backend being built in Appwrite. Vercel takes care of the automatization and load balancing of scaling, so the application will not go dead during high traffic [15]. The containerized architecture of Appwrite, operates the backend services, such as the user authentication, database interaction, and storing files. With this arrangement, scalability of the setup is easy, it is highly reliable and requires minimal human intervention. The backup and data transfer are automated, fault-tolerant design, and constant performance and availability, even under peak performance [14][15].

3.6.5 Security and Data Flow

The entire communication between the frontend and the Gemini API with the Appwrite services will be through HTTPS and token-based authentication. The permission model of Appwrite gives users control over their data by preventing access to individual datasets. Resumes uploaded and reports generated are stored in a secure place with limited access, and the user sensitive information is encrypted both in storage and during transfer [11].

3.7 System Assumptions

3.7.1 User Behavior Assumptions:

- Users have basic computer literacy and are well acquainted with web browser.
- Resume uploads are real employment practices that require professional optimization.
- Users will make suggested improvements to achieve system value in the best way possible.

3.7.2 Technical Environment Assumptions:

- Internet stability during analysis processing.
- Modern web browser support for required JavaScript features.
- File uploads are valid resume files of acceptable formats.

Chapter 4

Design

4.1 Introduction

The artificial intelligence Resume Analyzer system architecture is designed in a manner that is scalable, can be managed easily, and is easy to bring up to date. It isolates various components of the system so that an individual component will be able to perform well. This architecture also ensures that the system is smooth and secure. It utilizes the current software practices such as separation of the system into small components, which operate collectively, events to induce actions, and deploying the system in a manner that is compatible with cloud computing.

Design aims at making the user experience more enhanced through interfaces that can be used across various devices, instant feedback and ensuring the processes run in a smooth manner. The system is designed to safeguard information and adhere to privacy principles through the installation of security throughout the system.

4.2 System Block Diagram

The system architecture is organized into four major layers which interact with each other to provide smooth functionalities, scale and security:

4.2.1 Presentation Layer

A frontend based on React.js which will have a user-friendly interface where users upload their resumes, view the analysis as visualized, and see the results. It communicates directly with Appwrite through SDK and REST APIs on authentication, retrieving data, and managing files.

4.2.2 Backend Service Layer

This layer is a serverless cloud functions-based layer managed with the help of Appwrite and provides user authentication, database operations, file storage, and business logic. It provides the application gateway in addition to the logic processor on the back end, which makes it less complex and more maintainable.

4.2.3 AI Processing Layer

This layer brings together the Gemini model through its API of natural-language processing and resume analysis. On uploading a resume, Appwrite Functions pass the content through the Gemini API that process the information and provides insights and improvement recommendations.

4.2.4 Data Storage Layer

The database and storage services that are inbuilt in Appwrite are used to store user data, resume data, and output of the analysis in a way that is secure. The integrity and accessibility of the data are guaranteed by the permission-based access control and automatic replication features of Appwrite.

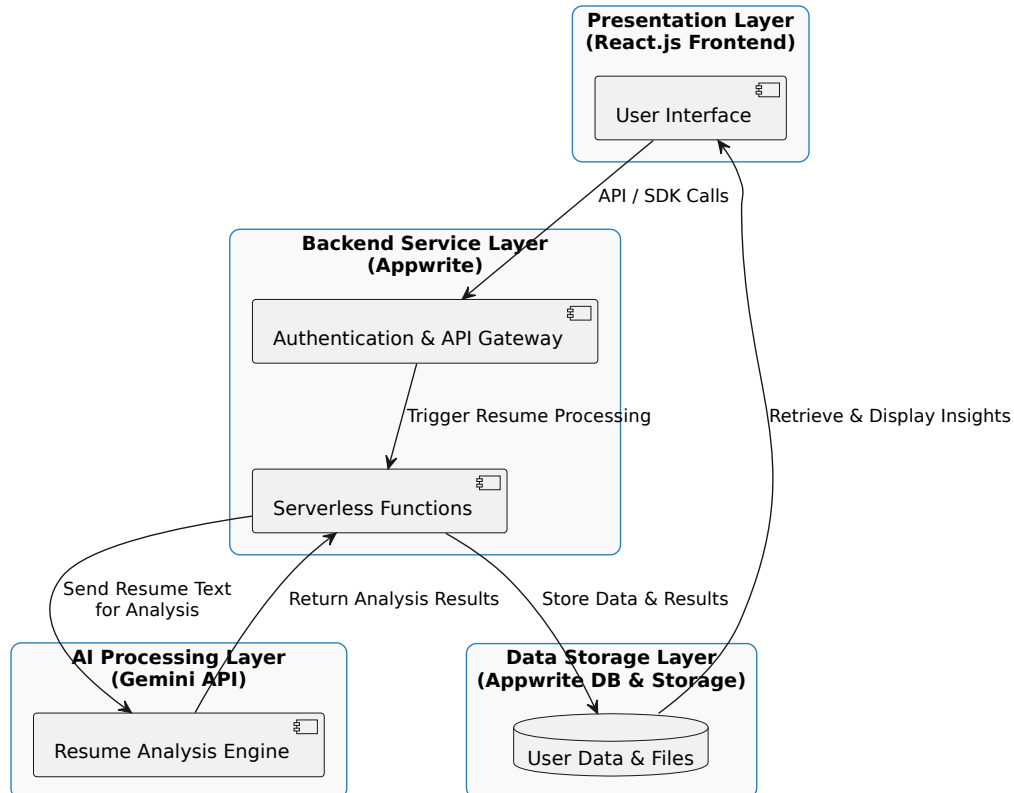


Figure 4.1: System Block Diagram

4.3 Data Flow Diagram (DFD)

4.3.1 Level 0 DFD (Context Diagram)

The system communicates most with Job Seekers, and they post their resumes to have them analyzed and optimized by the AI. The uploaded information is handled by the Appwrite backend that manages storage, request processing, and interaction between the system and outside the system [14]. Appwrite then communicates with the Gemini model to evaluate the content of the resumes, the structure, clarity, the strength of keywords as well as the industry relevance. The AI generates a scoring, optimization suggestions, and feedback after the processing.

At that point, the analysis report and personalized recommendations is returned to the job seeker as the outgoing data flow. This brings a complete loop of communication where the job seeker uploads something and receives the relevant and useful results. However, the context diagram does not expose internal complexity of the processing, but identifies the prominent data flows, external stakeholders, and explains the scope of the system in a clear manner.

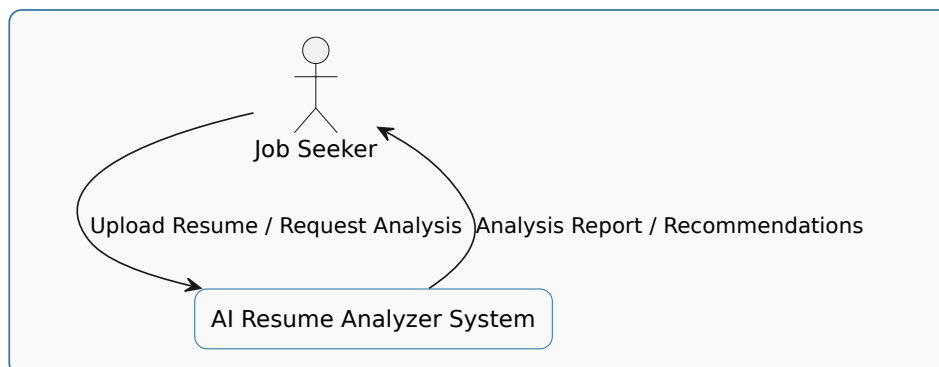


Figure 4.2: Level 0 Data Flow Diagram (Context Diagram)

4.3.2 Level 1 DFD (System Overview)

The key processes are User Authentication (done by Appwrite), Resume Upload and Storage, AI Analysis (through Gemini model), Recommendation Generation and Report Delivery. Data flows are represented by the user authentication requests, resume files uploaded, results of the analysis made by Gemini, and recommendations generated and delivered to the user in the dashboard.

In addition, the Level 1 DFD highlights the interaction of every subsystem through the well-defined data transfers. Whereas the resumes in storage are sent to the analysis procedure by AI, the authentication data streams are securely accessed. The findings of the systematic examination engineered by the Gemini model are transformed into feasible recommendations. These are then delivered to the report delivery module which

then formats them and presents them to the user through the dashboard. The diagram accentuates clear data flow, flexible processing as well as smooth integration of Appwrite services, the AI engine and the frontend interface.

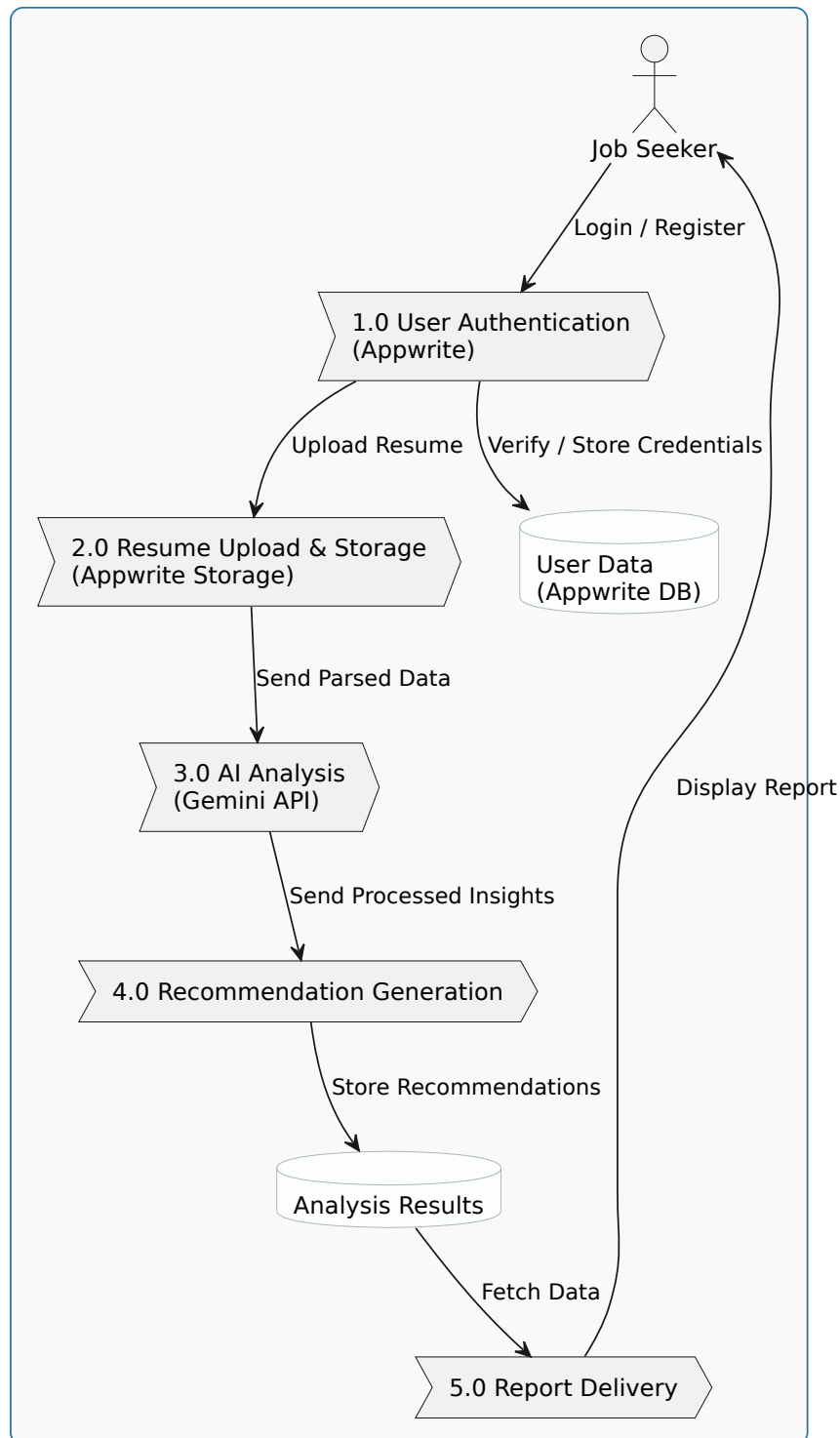


Figure 4.3: Level 1 Data Flow Diagram (System Overview)

4.3.3 Level 2 DFD (Detailed Process Flows)

The interface procedures comprise uploading and validating files, parsing and extracting their contents and analyzing the resume content conducted by the Gemini model. The model takes care of structure, grammar, keyword optimization, ATS simulation and personalized recommendations generation.

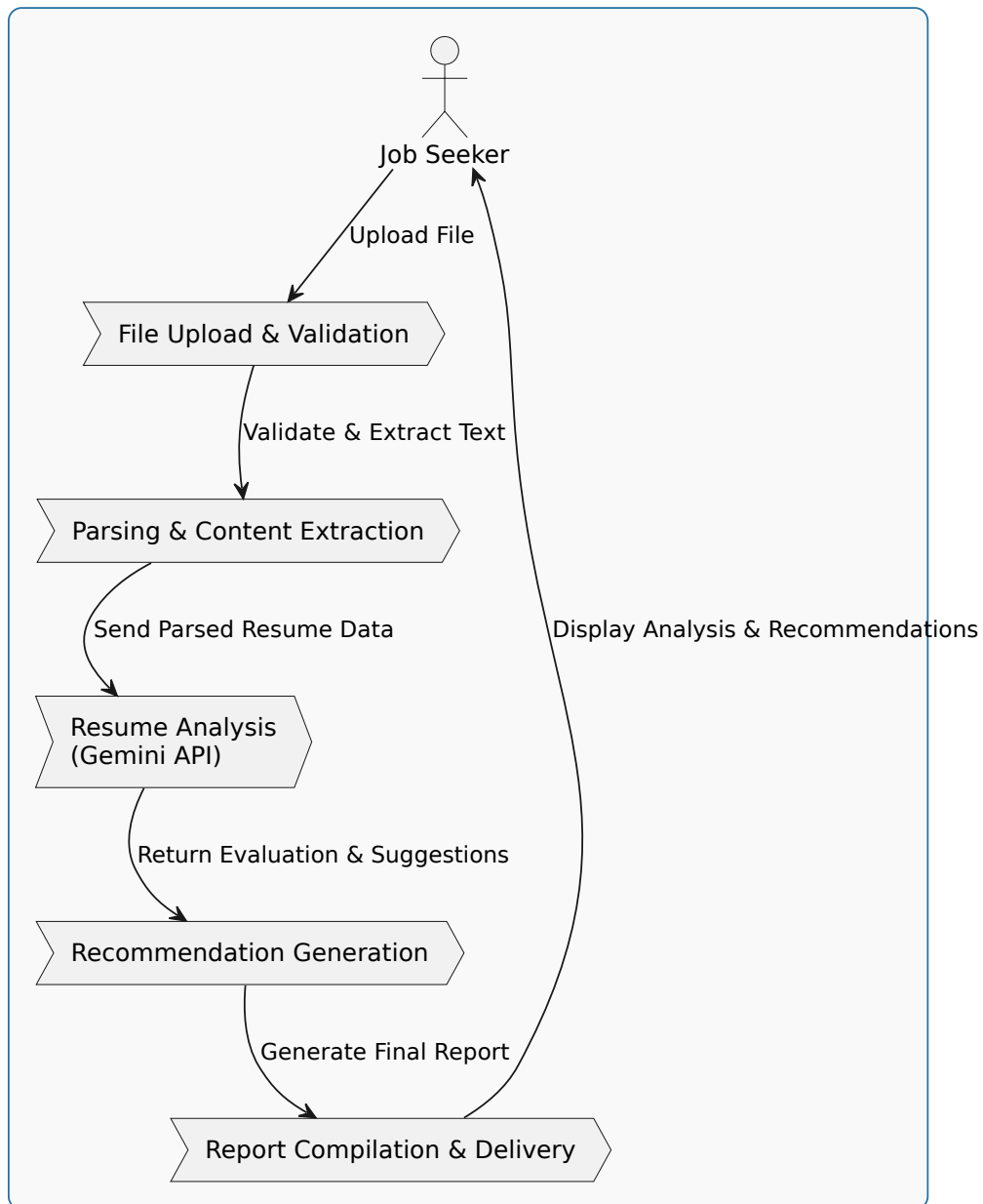


Figure 4.4: Level 2 Data Flow Diagram (Detailed Process Flows)

4.4 Design Considerations

4.4.1 Performance Optimization

Asynchronous patterns of processing blockages stop blocking operations when analyzing the resources requiring a lot of AI work. Caching plans minimize redundant calculations of similar resume files and frequent optimization plans.

4.4.2 Scalability Planning

The distributed cloud-based system of Appwrite enables scalability by scaling user load, data replication, and function execution automatically without any scaling of servers or orchestration of containers [3].

4.4.3 Security Implementation

Secure encryption between the two extremes of data communication, secure authentication, and input verification applied comprehensively. The privacy-by-design principles protect low data collection and transparent usage policy [11].

4.4.4 Maintainability Factors

Clear interface definition, modular code structure and extensive test coverage that is automated. The version control strategies facilitate the capability of safe updates and rollback.

4.5 System Flowchart

4.5.1 User Registration and Authentication Flow

1. User accesses platform and creates account.
2. Checking an email with the help of the authentication service of Appwrite.
3. The access to the dashboard is provided after the successful verification of the login.

4.5.2 Resume Analysis Workflow

1. Resume file uploaded by the user by means of web interface.
2. Checking of files and conversion of formats.
3. Text extraction and content parsing.
4. Extracted content sent to AI model for analysis.

5. Compilation of results and generation of recommendations.
6. Availability of notification and reports by the users.

4.5.3 Recommendation Implementation Process

1. Reviews are analysed and recommendations are made by the users.
2. Choice of improvements to be made.
3. Manual implementation and comparison.
4. Report download and export options.

4.5.4 System Flow Overview Explanation

The system flowchart is a visual representation of the entire operating workflow of the AI-Based Resume Analyzer at a high level. It outlines the usage of the site such as how to register, authenticate, analyze a resume, inspect recommendations and the final report. To minimize the chances of unauthorized access to the dashboard, the process begins with the creation of an account and secure authentication through Appwrite's identity service.

Upon getting into the system, the user can upload a resume file which initiates the process of resumes analysis. The backend is then used to perform file validation, format conversion, text extraction, and structured parsing. The AI model gets the processed material and evaluates the quality of the resume, relevance of the keywords, compatibility with AI, the structure, and clarity. The user is presented with the full set of recommendations based on the insights that were generated with the help of AI.

Once analyzed, the user proceeds to the recommendation implementation phase where, he or she reviews the recommended improvements, does any manual resumes edits required, and reviews the changes to make the improvement. Finally, a report is generated within the system that can be downloaded that summarizes all of the conclusions and suggestions.

4.5.5 Backend Processing Details

Once the user has submitted a resume, all the necessary functions are handled by the backend. The file is saved safely with the help of Appwrite, its format is checked, which is then converted to text, in case of the need. The extracted text is purified, systematized and prepared to be analyzed before being sent to the AI model.

The AI evaluates the clarity, structure, relevance of the keywords, and compatibility with the ATS of the resume. Its production is organized into recommendations which are classified. Once they have all these findings collected and they have been stored temporarily, the backend notifies the user that the analysis is done. Finally, it prepares the structured response as it comes to report creation and front-end presentation.

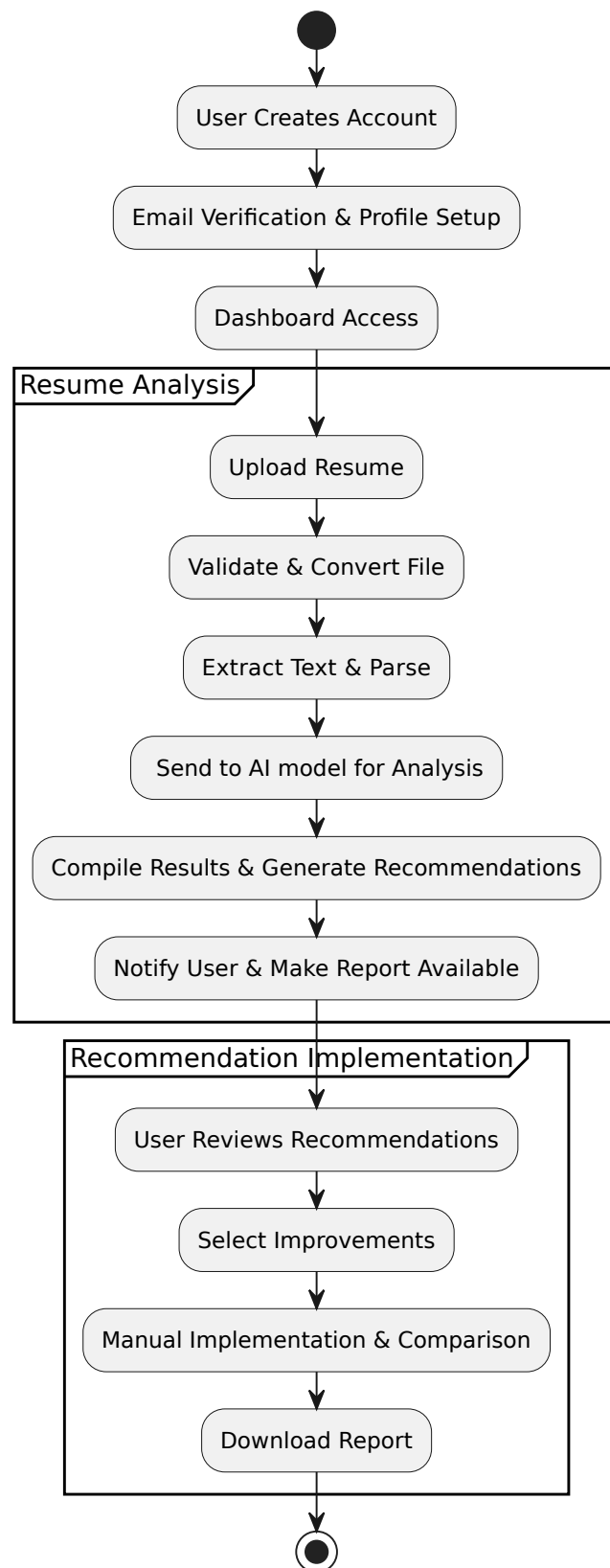


Figure 4.5: System Flowchart

4.6 UML Sequence Diagram

4.6.1 User Authentication Sequence

The authentication of users will solely depend on the secure session authentication mechanism of Appwrite.

1. The Frontend also takes the user to provide login credentials and this triggers the authentication request.
2. The credentials are uploaded to the Authentication Service of Appwrite to verify them.
3. Appwrite compares the input with account records stored and forms a secure session token in case of successful authentication.
4. This token is sent back to the front end and put in a temporary store to authorize its future interactions.
5. Once authenticated, the user will be redirected into the dashboard, and system features are accessible.
6. The frontend appends the session token to all further requests, which means that the Appwrite services can only respond to authenticated requests.

4.6.2 Resume Analysis Sequence

The interactions between the frontend, Appwrite services, and the Gemini API are several coordinated interactions that make up the resume analysis workflow. The process starts by the user choosing a resume and uploading it using the interface.

1. The Frontend Interface allows the user to initiate the upload which prepares the file to undergo processing.
2. The frontend does some basic validation and transfers the file to the Appwrite Storage making sure that it is properly handled.
3. Authentication Service is the process of verifying the user session to ensure access authorization once it has been verified.
4. When the upload is done, an Appwrite Cloud Function is fired to initiate the analysis process.
5. The extractor takes out some text data in the uploaded document and sends it in a structured request to the Gemini API.

6. The Gemini model processes the content, making grammar checks, evaluating keywords, structurally evaluating, and providing AI-generated recommendations.
7. The Cloud Function takes the results of the analysis and stores the results and metadata in the Appwrite Database.
8. The Notification Service provides the user with the message that the analysis is complete and it is possible to see the results.
9. The frontend then obtains the information stored and gives the results and the recommendations in a structured manner to the user.

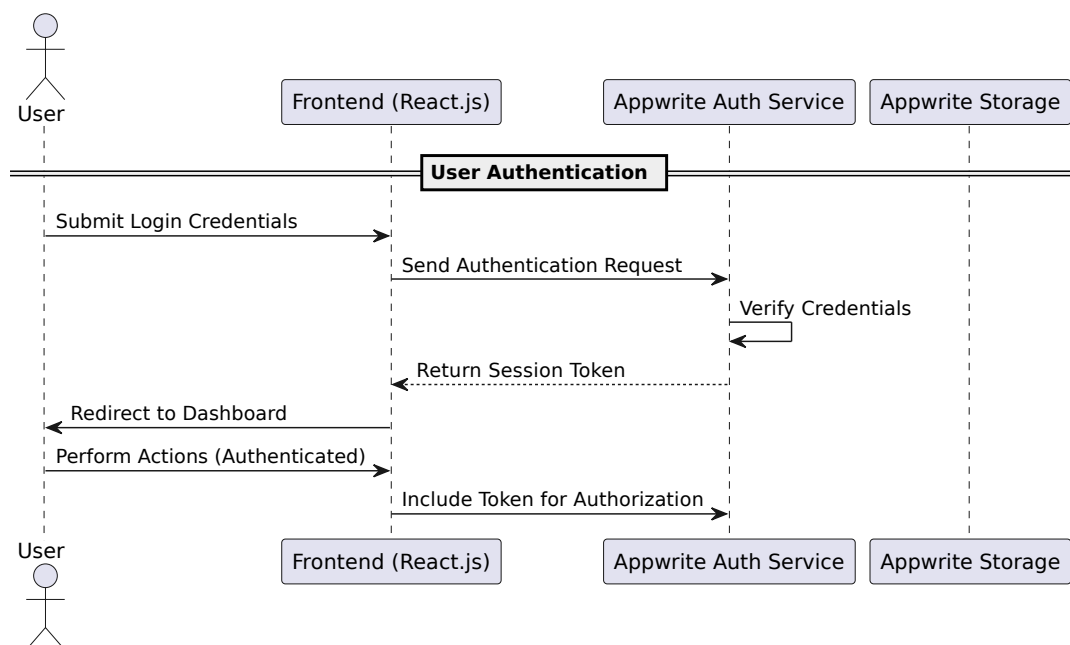


Figure 4.6: User Authentication UML Sequence Diagram

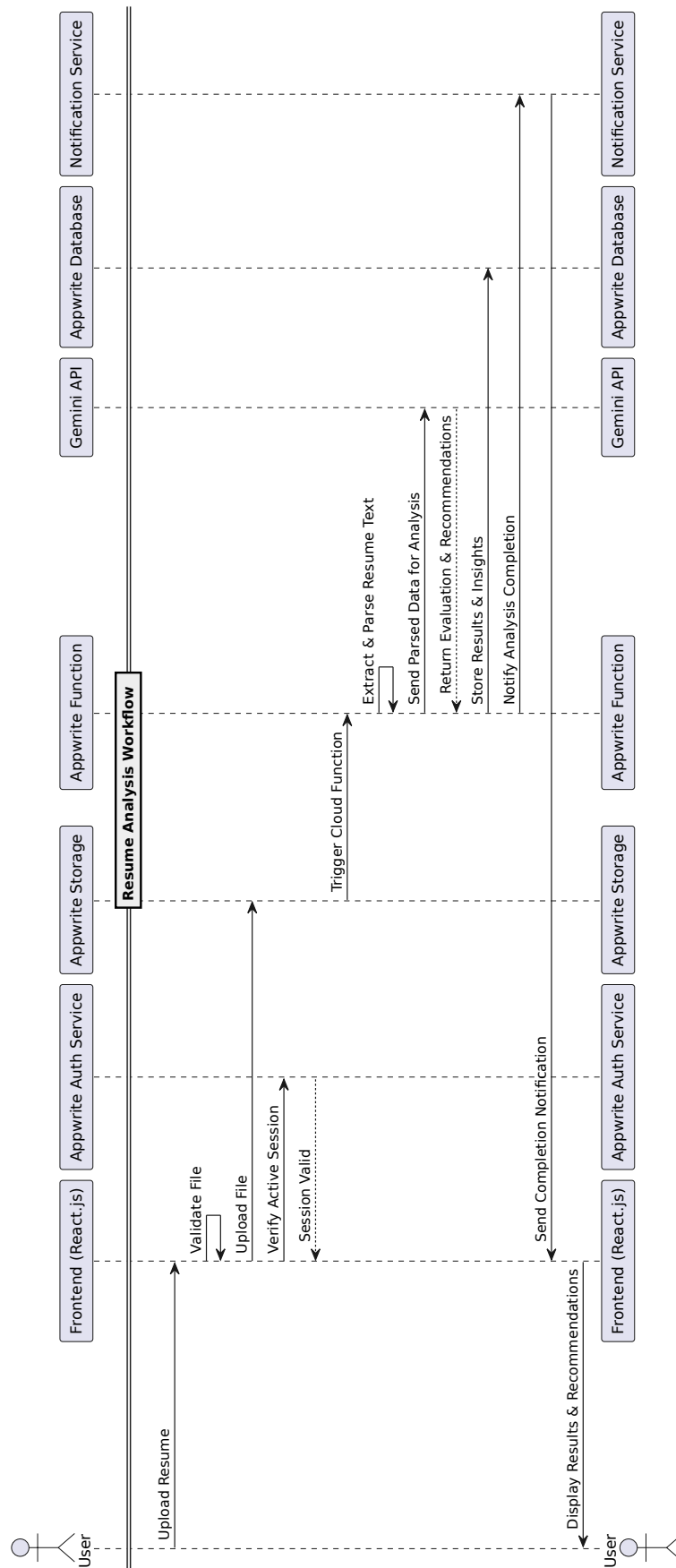


Figure 4.7: Resume Analysis UML Sequence Diagram

4.7 Use Case Diagram

4.7.1 Primary Actors

- Job Seeker: Individual users seeking resume optimization.
- Administrator: System management and user support.

4.7.2 Use Cases for Job Seeker

- Register and create account.
- Upload resume for analysis.
- View analysis and recommendations.
- Implement suggested improvements.
- Download recommendation/analysis reports.
- Access historical reports.

4.7.3 Use Cases for Administrator

- Monitor system performance and usage.
- Manage user accounts and data.
- Update AI integration settings.
- Generate analytics and usage reports.
- Handle support requests.

4.7.4 System Use Cases

- Process file uploads and parsing.
- Execute AI analysis through API integration.
- Generate and store personalized recommendations.
- Manage user sessions and authentication.
- Send notifications.

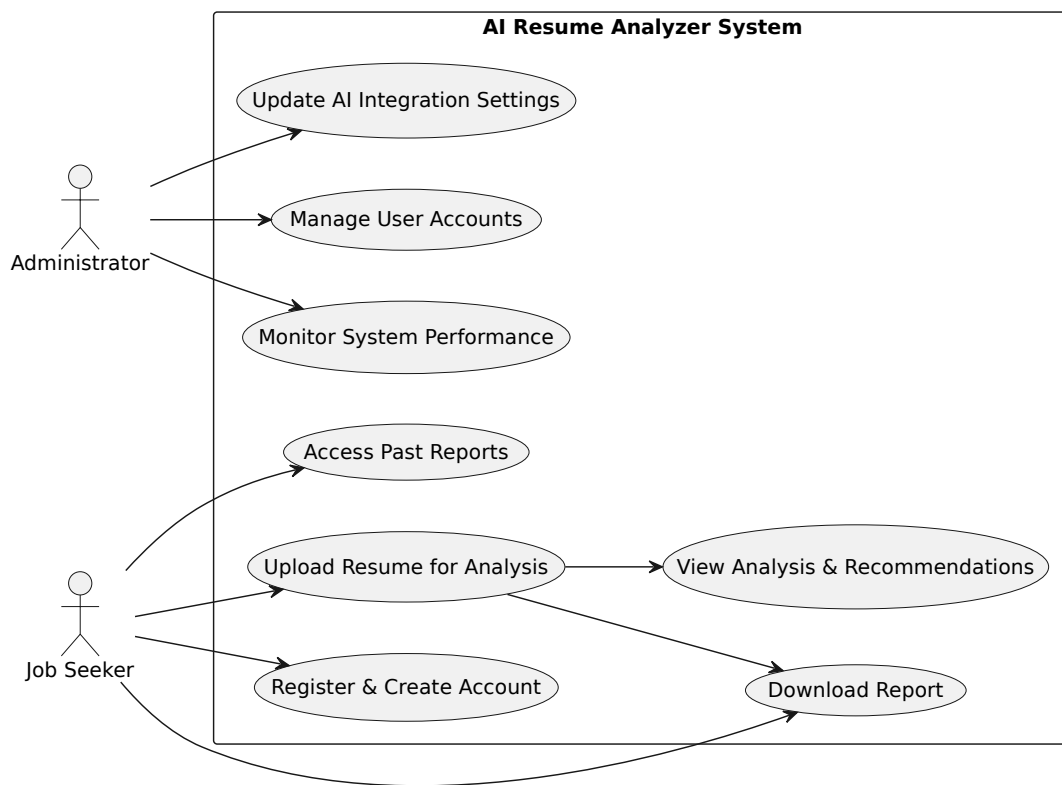


Figure 4.8: Use Case Diagram

4.8 UML Component Diagram

4.8.1 Frontend Components

- **UserInterface:** Offers the general layout of the application, navigation pattern, and the general structure. It facilitates view switching and stores consistency throughout the user interface.
- **FileUpload:** Oversees the resume selection process, the validation of the format and the error management, as well as the communication with the Appwrite Storage to upload the files safely.
- **AnalysisDisplay:** This shows the results of the AI analysis, by showing insights, feedback, and a general assessment of the resume. It also controls user interplay on the presented information.
- **RecommendationManager:** This shows specific improvement recommendations and users can interact or implement the recommendations. It assists in comparison perspectives and procedural advice when necessary.

- **UserProfile:** Deals with account-related functions including profile information, user settings and displaying analysis history stored in the backend.

4.8.2 Backend Components

- **AuthenticationService:** In charge of user registration, logging in, session management, and controlled access with the in-house authentication functions of Appwrite.
- **DatabaseService:** Stores structured information such as user profiles, the results of analysis or timestamps and system generated metadata. It guarantees a steady and safe call of information.
- **StorageService:** This service controls the upload, storage and retrieval of resume files. It ensures that files remain intact and that accessibility controls are enforced to secure documents of users.
- **FunctionService:** Invokes serverless process like the sending of resume text to the Gemini API, the analysis results, and updating the corresponding database entries.
- **NotificationService:** This is a user notification service that delivers user notifications as analysis completion messages via the event system in Appwrite, enhancing user responsiveness and awareness.

4.8.3 Integration Components

- **APIConnector:** Communication with the Gemini model, both to and from structured resume content and AI-generated feedback. It also makes the communication during the analysis requests reliable and secure.
- **ReportGenerator:** This process converts an AI output into a structured readable report of an analysis. It structures the results to be displayed on dashboard and ready them to be exported or used in the future.

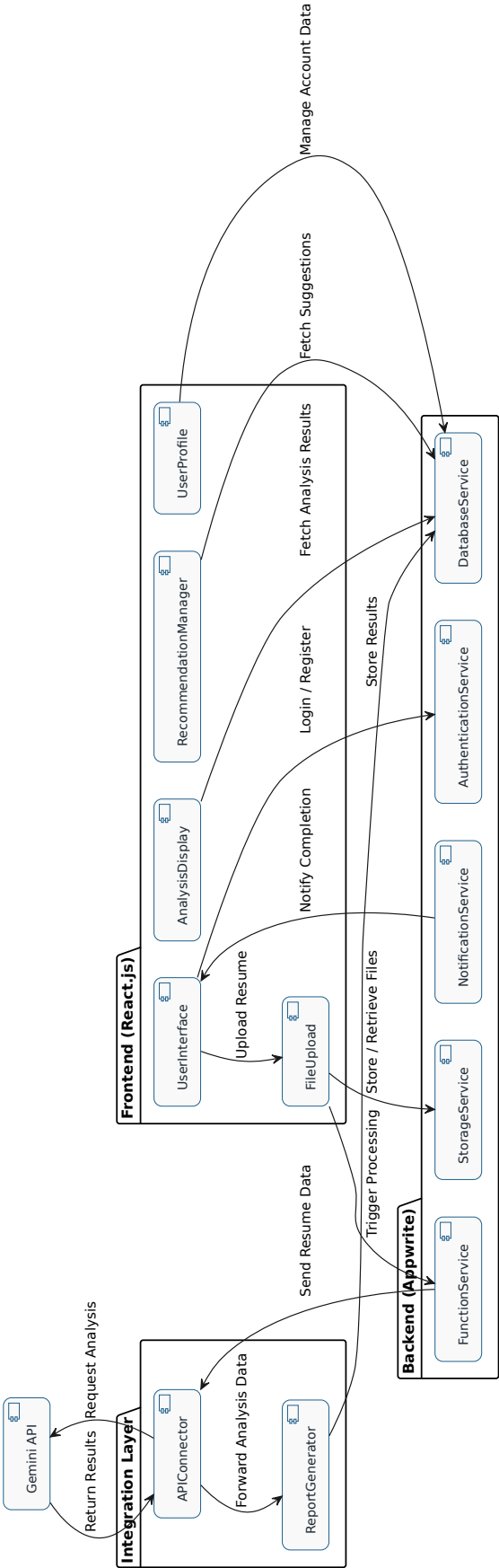


Figure 4.9: UML Component Diagram

Chapter 5

System Implementation

5.1 Introduction

The implementation stage is the process defined by the shift between the theoretical design of the AI-Based Resume Analyzer and a workable Web Application. It pays attention to converting system requirements and specifications into a working system. The modules (frontend, backend, AI integration, database) were developed based on modern web development technologies and cloud-based services to meet the requirements of scalability, security, and performance. The implementation occurred in cyclical cycles with agile development principles, and the functional and non-functional requirements were enforced.

5.2 Implementation Environment

The implementation of the system was carried out in a cloud-based development environment, using the following software and hardware tools:

- **Frontend:** React.js 19, HTML5, Tailwind CSS, JavaScript (ES6+).
- **Backend:** Appwrite Cloud (Authentication, Database, Storage, Cloud Functions).
- **AI Integration:** Gemini API (NLP-based Resume Analysis).
- **Database:** Appwrite's Built-in Database & Storage Services.
- **IDE:** Visual Studio Code.
- **Version Control:** Git & GitHub.
- **Deployment:** Vercel
- **OS/Hardware:** Windows 10, Intel i7 Processor, 8GB RAM.

5.3 Frontend Implementation

The Frontend was deployed with React.js, as it is a component-based framework with a powerful state management system. The interface enables users to be basically involved with the system with the use of intuitive navigation and real time reaction.

5.3.1 User Interface Design

All interface components were designed with Tailwind CSS to ensure that all interfaces are responsive to any screen size. There are the following primary elements of the UI:

- **Upload Interface:** It gives the user the opportunity to drag and drop PDF resume files.
- **Dashboard:** shows the use of ATS compatibility score, the analysis report, and recommendations provided by AI.
- **History:** Displays all of the historical notes of the user.

UI is dynamically connected to Backend APIs, which is a single-page application providing a smooth experience.

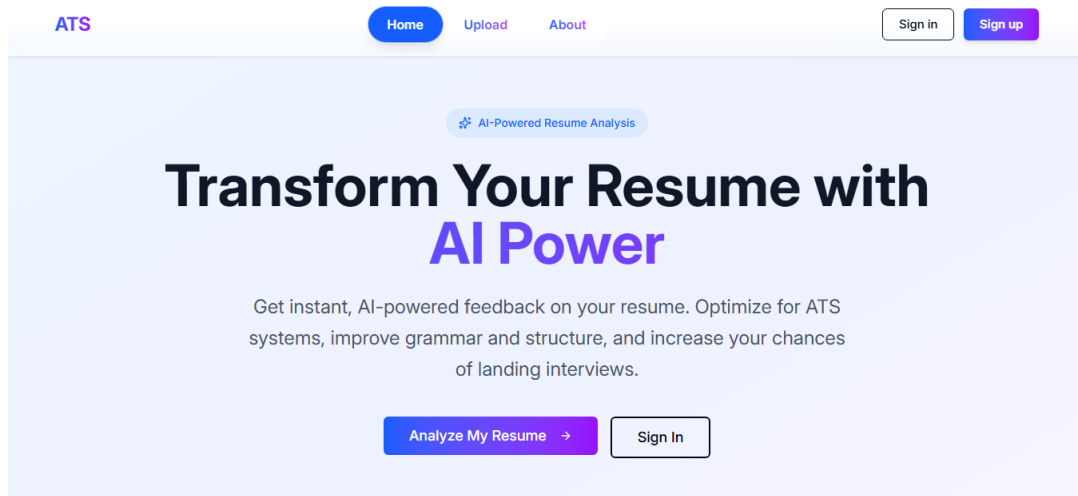


Figure 5.1: Home Page (1)

Description: The Home Page of the system that shows the landing layout with the links to the navigation, summary of the features, and the action buttons for resume analysis and user authentication

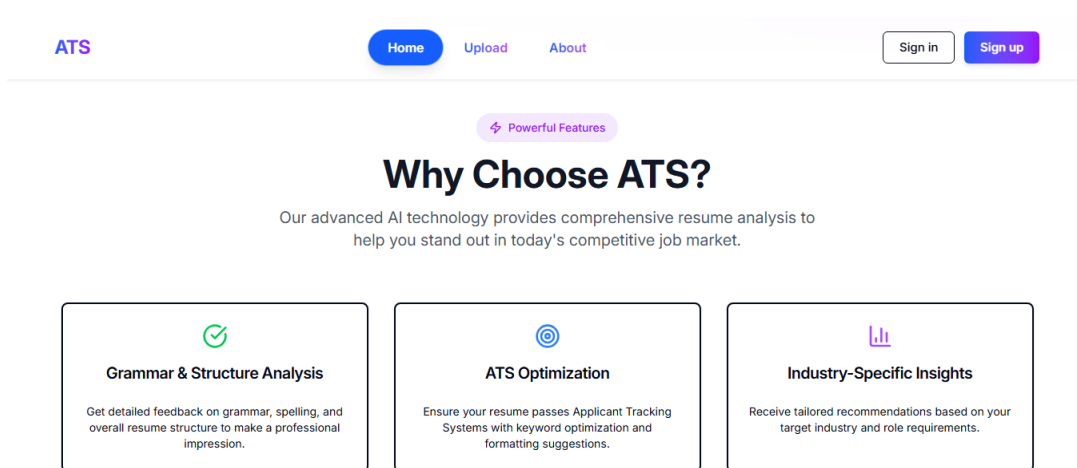


Figure 5.2: Home Page (2)

Description: This figure displays the “Why Choose ATS?” section, which focuses on the main aspects, which can be the grammar analysis, ATS optimization, and industry-specific insights.

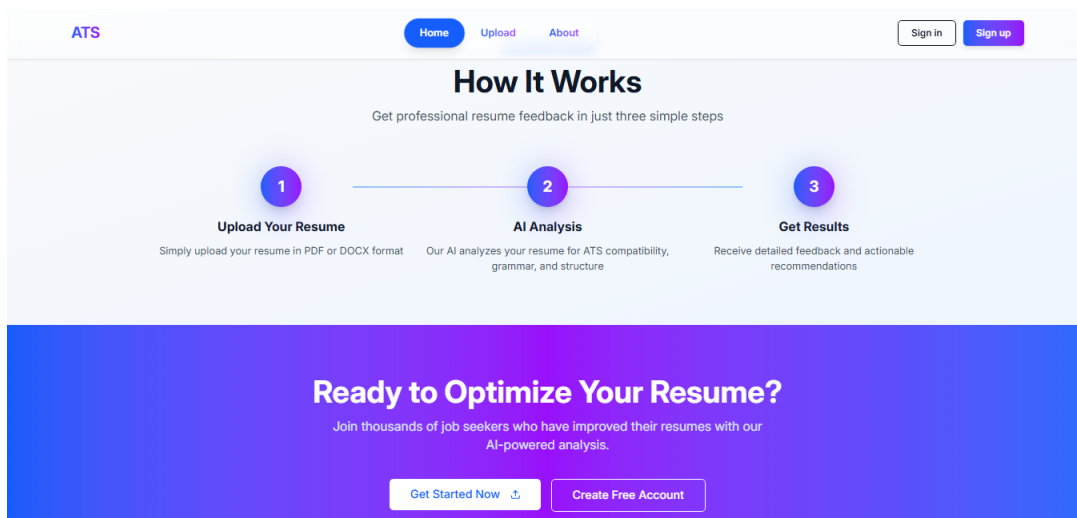
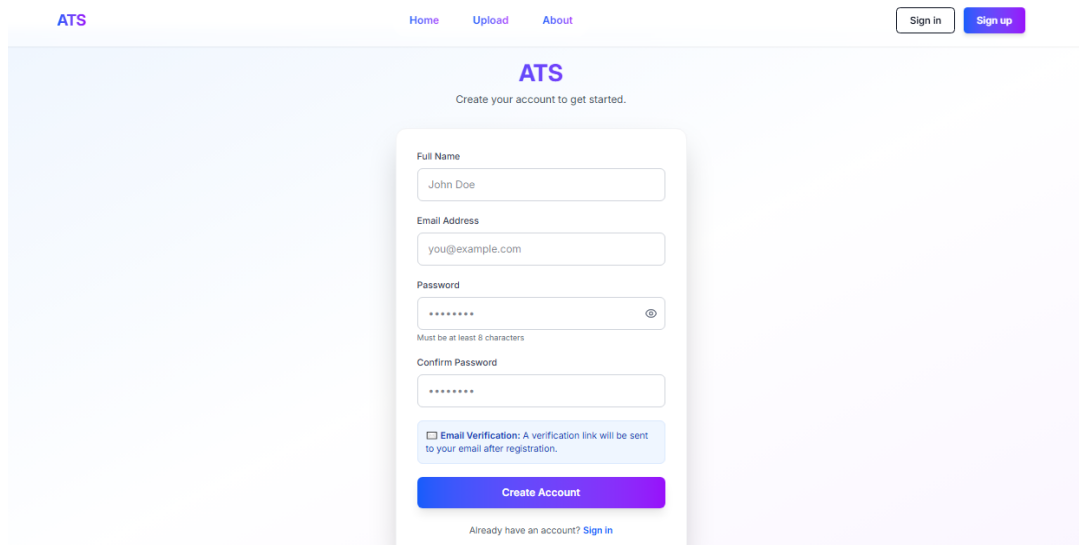


Figure 5.3: Home Page (3)

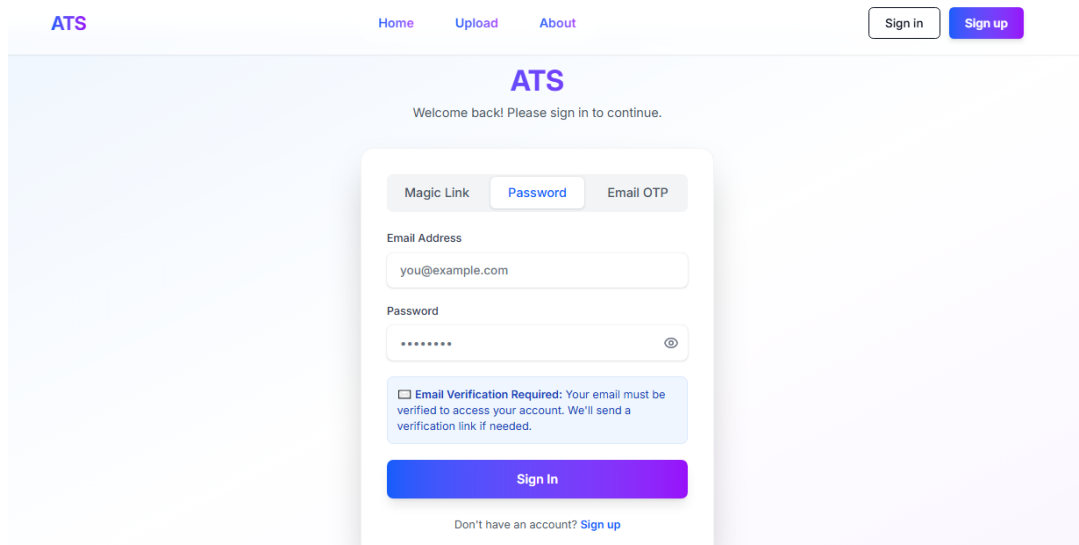
Description: The figure illustrates the “How It Works” section and presents the three-step process, namely, up- loading a resume, AI analysis, and receiving results.



The image shows the 'Sign Up' page of the ATS application. The header includes the 'ATS' logo, navigation links for 'Home', 'Upload', and 'About', and buttons for 'Sign in' and 'Sign up'. The main content area features the 'ATS' logo and the text 'Create your account to get started.' Below this is a registration form with the following fields: 'Full Name' (containing 'John Doe'), 'Email Address' (containing 'you@example.com'), 'Password' (with a strength indicator and a note 'Must be at least 8 characters'), and 'Confirm Password'. A checkbox for 'Email Verification' is present, with a note that a verification link will be sent. A large purple 'Create Account' button is at the bottom of the form, followed by a link for users who 'Already have an account? Sign in'.

Figure 5.4: Sign Up

Description: This figure demonstrates the Sign-Up page, in which the user can create an account by providing the name, email, password, and agreement information.



The image shows the 'Sign In' page of the ATS application. The header is identical to the Sign Up page. The main content area features the 'ATS' logo and the text 'Welcome back! Please sign in to continue.' Below this are three tabs: 'Magic Link', 'Password' (which is selected), and 'Email OTP'. The 'Password' tab contains an 'Email Address' field (with 'you@example.com') and a 'Password' field. A checkbox for 'Email Verification Required' is present, with a note that the email must be verified. A large purple 'Sign In' button is at the bottom of the form, followed by a link for users who 'Don't have an account? Sign up'.

Figure 5.5: Sign In

Description: This figure shows the Sign-In page where the Sign in button can be used to log in using email and password, with options for magic link and email OTP log in.

Figure 5.6: Upload Page

Description: This figure demonstrates the Resume Upload page where individuals choose the job role and upload a resume file to be analyzed by AI.

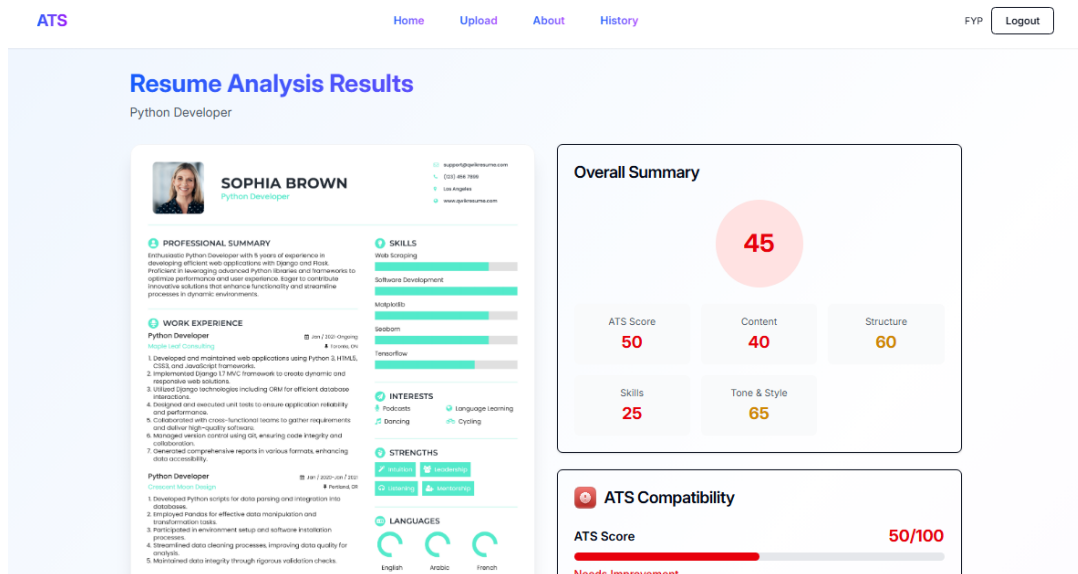


Figure 5.7: Result (1)

Description: This figure presents the upper section of the results page, which has the preview of the analyzed resume and the total score and the ATS compatibility rating.

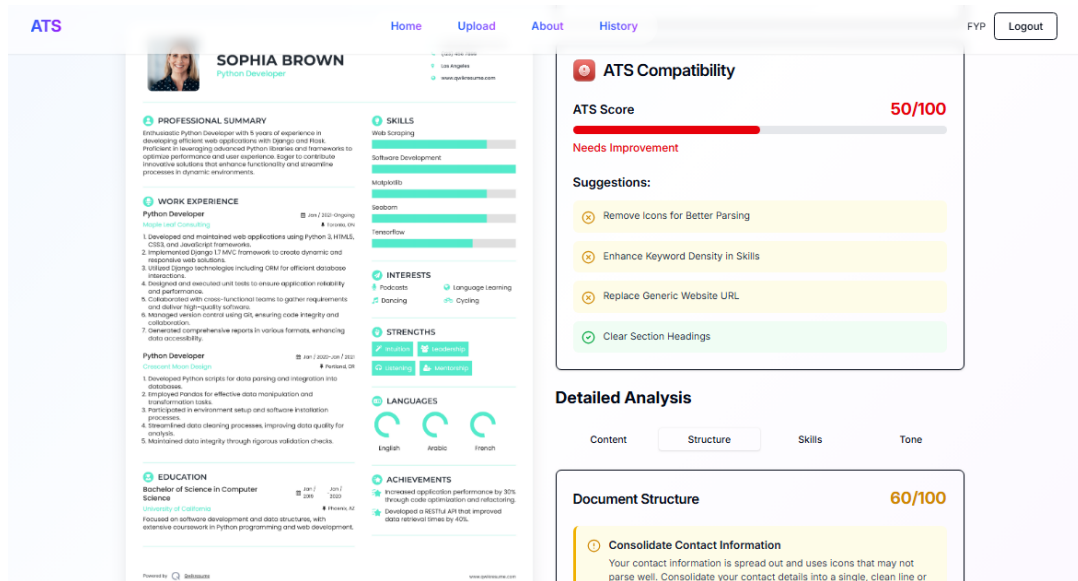


Figure 5.8: Result (2)

Description: This screenshot represents the resume analysis platform with the resume preview on the left and ATS compatibility score with detailed information on the right.

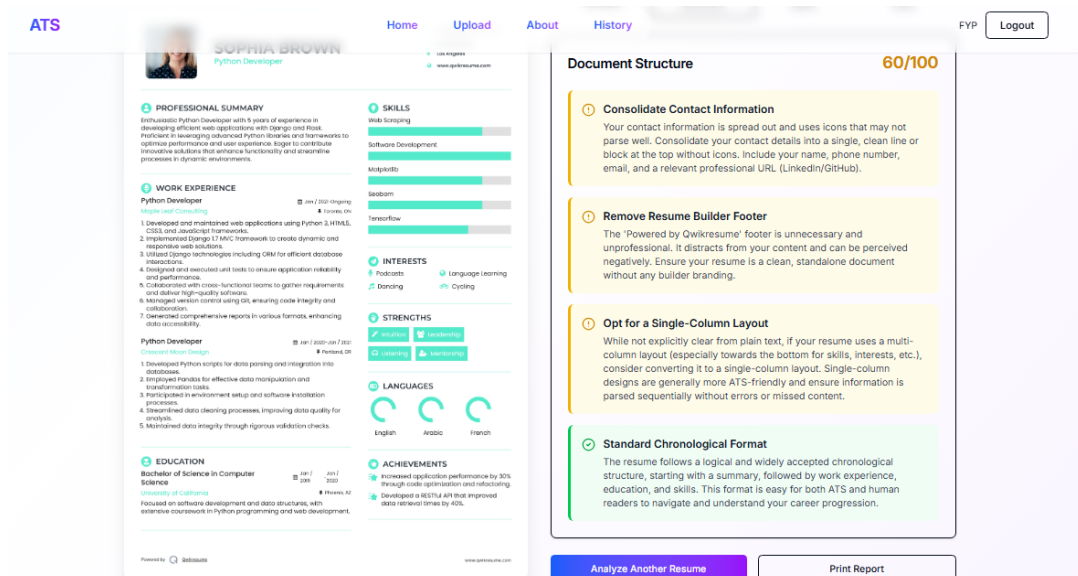


Figure 5.9: Result (3)

Description: In this figure, the resume analysis interface is used to display the resume on the left side and a suggestion overview panel on the right side with tips on how the resume may be improved.

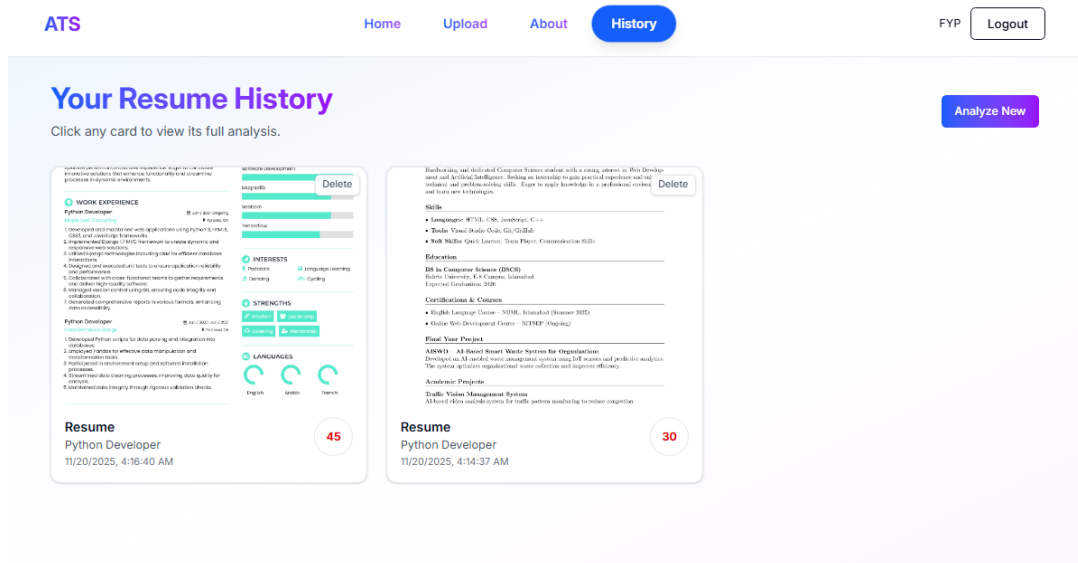


Figure 5.10: History Page

Description: This illustrates the resume history page where the already examined resumes are presented with preview cards and a button to see the details.

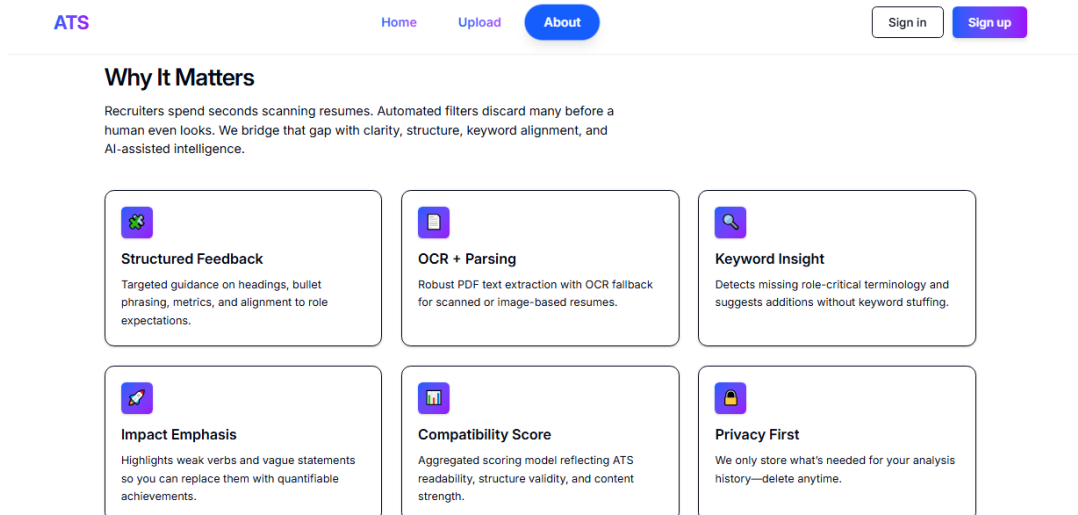


Figure 5.11: About Page (1)

Description: This figure shows the about page which highlights the key features of the system by informational cards, showing their importance and functionality.

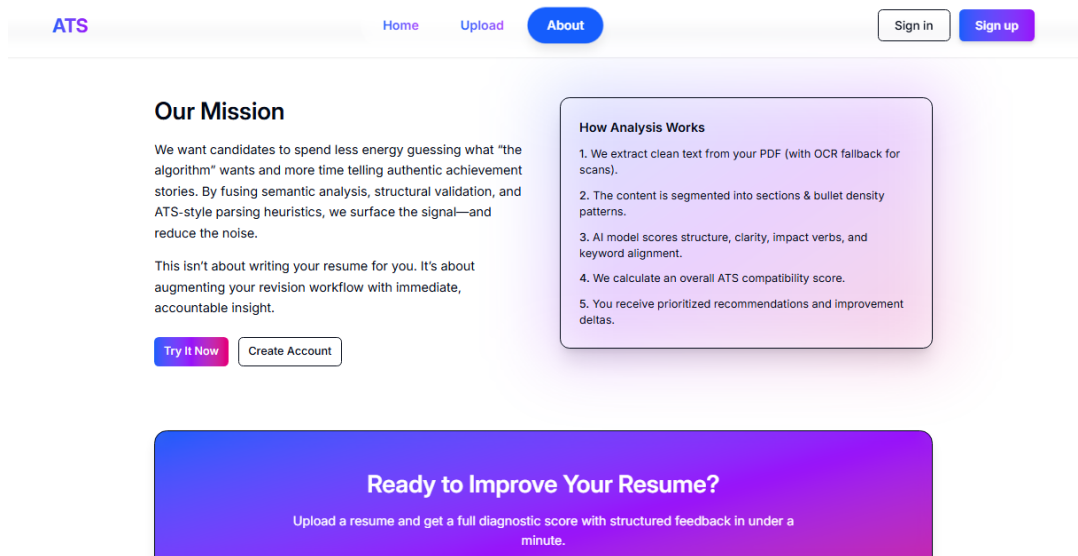


Figure 5.12: About Page (2)

Description: In this figure, the mission statement of the system and the highlights of the features of the system are displayed, as well as a call-to-action section where the users are invited to post their resumes.

5.3.2 Routing and State Management

The application uses React Router v7 and React Hooks for client-side routing and managing state and component lifecycle respectively. Appwrite authentication tokens are saved in memory, which guarantees authenticated users secure routes.

5.4 Backend Implementation

The backend of AI-Based Resume Analyzer was built using Appwrite, which is a backend-as-a-service (BaaS). It is a platform offering in-built authentication, data management, file storage, and cloud functions modules.

5.4.1 Authentication and Authorization

The Appwrite Authentication Service was used to provide user registration, login and handling of sessions. JWT-based tokens and multi-step verification ensure safer user access. The frontend was directly connected to authentication APIs, by the use of the JavaScript SDK of Appwrite.

5.4.2 Cloud Functions for Business Logic

Cloud Functions in Appwrite are used to process server-side code, such as:

- Sending content from the uploaded resume to the Gemini API.
- Analyzing and storing AI generated analysis.
- The control of asynchronous processes like notifications and file authentications.

These are JavaScript functions that are executed in the secure sandboxed environment of Appwrite, removing the need for manual configuration of a server.

5.5 AI Model Integration

The fundamental component of the system is the AI integration that allows analyzing resume intelligently and providing feedback. The Gemini API is the AI application that analyzes resumes regarding grammar, keyword match, format, and ATS compatibility.

5.5.1 Data Processing Workflow

Following is the workflow of data processing:

1. Appwrite extracts the file and parses the text in it.
2. The text is passed to the Gemini API in a parsed form through a secure call.
3. Gemini examines the contents and it gives organized responses such as:
 - Evaluation of grammar and syntax.
 - Keyword density analysis.
 - ATS compatibility score.
 - Recommendations on a section-by-section basis.
4. The data is stored in Appwrite database and displayed in the frontend dashboard.

5.5.2 ATS Simulation

ATS scoring simulation model can be compared to real-life systems such as Workday and Taleo. It analyzes alignment of key words, structure of the document and relevance of the job in relation to the candidate using the NLP features of Gemini to give the user a realistic ATS Score.

5.6 Database and Storage Configuration

The database layer of the system is fully controlled by Database and Storage Services of Appwrite.

- User profiles, resume data, and the results of AI analysis are stored in a structured JSON format in Appwrite Database.
- Appwrite Storage manages upload of files, resume documents, as well as downloadable reports.

The role-based access control implemented by Appwrite will ensure that each user has access to their own data only, and it will be confidential and compliant with the data protection standards.

5.7 System Deployment

Deployment had been done using Vercel which offers serverless hosting with automatic scaling and load balancing for the React.js frontend.

5.7.1 Key Deployment Steps

The key deployment steps are as follows:

1. Setting up Appwrite project to use as the backend services (e.g. authentication, database) and connecting the frontend using SDK credentials.
2. Hosting the React.js frontend on Vercel, making use of the Content Delivery Network (CDN) to achieve the best possible performance.
3. Setting up of a custom domain and encryption on Vercel to access the React.js application safely.
4. Adding Appwrite API keys in Vercel using environment variables in order to securely communicate with Appwrite backend.
5. The continuous integration and deployment (CI/CD) is controlled through GitHub Actions, by guaranteeing automated deployment and build pipelines to Vercel.

Chapter 6

System Testing and Evaluation

System testing and evaluation will make sure that AI-Based Resume Analyzer will operate as per the system specifications that are established during design and implementation. This step would ensure that all the integrated modules (frontend, backend, AI processing, and data storage) work well as one complete system. In this phase, the system is also tested in the real-life situations of using it in order to ensure it works correctly not only when isolated, but also when all the elements are interacting in the case of the normal user working processes.

System testing is done to find out the problems with bottleneck in performance, integration failure, limitations in user experience, and the behaviour that would otherwise not be noticed until the whole system is operational. This phase is also useful to ensure that the platform is capable of managing various resume forms, user operations, and external requirements like Appwrite services and Gemini API. System testing performs the validation of such interactions and so, the end product will be reliable, stable, and fit to be used in the real world.

System evaluation also investigates the quality of generated AI model responses, the accessibility of system responses to human users, and the overall resilience of the platform to edge cases e.g. invalid input, high load or slow response of external services. The information acquired during this process helps to make final adjustments and to point out the aspects that need to be improved in the next iterations of the system.

6.1 Testing Overview

Objectives: Check the correct functionality, performance, security and usability, check the integrations with the Appwrite and Gemini API, and verify the quality of AI output.

Test Environment & Tools: React.js frontend, Appwrite backend (cloud) and Gemini API (AI), manual browser test on Chrome/Firefox, and simple load simulation tools.

6.2 Functional Testing

The core application functionality was checked with specifications using functional testing. Tests were done on user registration, authentication, resume uploading and parsing, and report generation. Most functional flows worked well in a normal condition. Exception: invalid/corrupted file operation did not give clear error and made the interface to freeze. All the functional test information is provided in the test-case tables which follow.

6.3 Integration Testing

End to end communications between the React frontend and Appwrite backend services and the Gemini API were tested correctly by integration tests. These tests comprised storage / DB consistency tests and a simulated API failure test. Integration points also tended to work as anticipated. Mock responses to test error-handling paths were tested to confirm the retries and graceful failure handling.

6.4 Performance Testing

Single-request latency and performance was tested at concurrent load on performance. Light load was within the target of single- resume processing. In concurrent-upload conditions the system became congested. There were queue under loads and timeouts at simultaneous uploads, p95 latency was higher than acceptable limits on repeat API requests, and stress testing revealed that it was unstable at some point beyond the number of simultaneous users with taxing timeouts. These findings suggest that cloud execution of functions should be tuned, retries conducted, and focused load-optimization done.

6.5 Security Validation

Security validation was based on input sanitization, session management and access control. Role-based access control, correct session expiry, and barring unauthorized access to data were all verified using tests. The API endpoints as well as the input validation mechanisms performed as desired in the test scenarios.

6.6 Usability and Error Messaging

The usability tests were conducted to determine the dashboard clarity, navigation, progress indicators, and mobile responsiveness. The interface is easy to use and receptive among the supported devices. There are also good progress messages and poor error messages. Informative feedback in exceptional failure modes (e.g. when analysis remains frozen after invalid uploads) is the most significant usability gap.

6.7 AI Evaluation and Consistency

Gemini-driven outputs repeatability and relevance were measured with AI-specific tests. The numerous repetitions of the same inputs yielded the same results within the acceptable deviations. Job-description expectations, industry-sensitive recommendations were made on sampled resumes. The general AI applicability and accuracy are good to the present extent.

Table 6.1: Test Case for User Registration (ID: TC-1)

Test Case ID	TC-1
Test Case Name	User Registration
Related Use Case	Register and create account
Pre-Condition	User on registration page
Test Steps	1. Open the registration page. 2. Enter valid email, password, name 3. Submit form 4. Check email for verification link
Expected Result	Account created, verification email sent, user record saved
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	User awaits email verification

Table 6.2: Test Case for Email Verification (ID: TC-2)

Test Case ID	TC-2
Test Case Name	Email verification
Related Use Case	Confirm account via email
Pre-Condition	Registration completed, verification email received
Test Steps	1. Click verification link in email 2. Observe confirmation page 3. Attempt login
Expected Result	Account status changes to verified, login allowed
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	User account verified

Table 6.3: Test Case for User Login (Valid) (ID: TC-3)

Test Case ID	TC-3
Test Case Name	User Login (Valid)
Related Use Case	Login to dashboard
Pre-Condition	User registered and verified
Test Steps	1. Navigate to login page 2. Enter valid credentials 3. Click login
Expected Result	Successful login, redirected to dashboard, session token issued
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Authenticated session active

Table 6.4: Test Case for User Login (Invalid) (ID: TC-4)

Test Case ID	TC-4
Test Case Name	User Login (Invalid)
Related Use Case	Handle invalid credentials
Pre-Condition	User exists
Test Steps	1. Enter incorrect password or email 2. Click login
Expected Result	Login denied, clear error message, no session created
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	No session created

Table 6.5: Test Case for Resume Upload (Valid Formats) (ID: TC-5)

Test Case ID	TC-5
Test Case Name	Resume Upload (Valid Formats)
Related Use Case	Upload resume for analysis
Pre-Condition	User logged in
Test Steps	1. Open upload dialogue 2. Select PDF under 10MB 3. Click upload 4. Observe progress and completion
Expected Result	File accepted, stored in Appwrite Storage, parsing job queued
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Resume available in user's files

Table 6.6: Test Case for Resume Upload (Invalid/Corrupted File) (ID: TC-6)

Test Case ID	TC-6
Test Case Name	Resume Upload (Invalid/Corrupted File)
Related Use Case	Handle bad uploads
Pre-Condition	User logged in
Test Steps	1. Attempt to upload unsupported file or corrupted PDF 2. Click Upload
Expected Result	Upload rejected, informative error message, no storage entry created
Actual Result	Upload accepted, no error message
Pass/Fail	Fail
Post-Condition	System stuck on loading screen

Table 6.7: Test Case for Resume Parsing Accuracy (ID: TC-7)

Test Case ID	TC-7
Test Case Name	Resume parsing accuracy
Related Use Case	Extract text and sections
Pre-Condition	Valid resume uploaded
Test Steps	1. Upload resume with multiple sections and formatting 2. Trigger Parsing 3. Inspect parsed JSON for sections (Education, Experience, Skills)
Expected Result	Sections recognized correctly, main text extracted with minimal loss of formatting context
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Parsed content stored

Table 6.8: Test Case for Analysis Report Generation (ID: TC-8)

Test Case ID	TC-8
Test Case Name	Analysis Report Generation
Related Use Case	Produce analysis and suggestions
Pre-Condition	Parsed content available
Test Steps	1. Send parsed text to Gemini via Appwrite Function 2. Receive AI response 3. Render report on dashboard
Expected Result	Report shows grammar checks, keyword suggestions, ATS score, and recommendations
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Report saved in database

Table 6.9: Test Case for Single Resume Processing Time (ID: TC-9)

Test Case ID	TC-9
Test Case Name	Single Resume Processing Time
Related Use Case	Performance SLA
Pre-Condition	System idle, AI and Appwrite reachable
Test Steps	1. Upload a standard 1-page resume 2. Start timer at upload 3. Stop timer when report displays
Expected Result	End-to-end processing $\leq$ 45 seconds (target)
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Processing time logged

Table 6.10: Test Case for Concurrent Uploads (Load) (ID: TC-10)

Test Case ID	TC-10
Test Case Name	Concurrent Uploads (Load)
Related Use Case	Scalability under load
Pre-Condition	Load testing tool configured
Test Steps	1. Simulate 50-100 concurrent resume uploads 2. Monitor CPU, memory, Appwrite function queue, and response errors
Expected Result	System handles load with acceptable latency and error rate under defined thresholds
Actual Result	System slowed with 25–30 simultaneous uploads, Appwrite queue delays caused timeouts
Pass/Fail	Fail
Post-Condition	Increased latency observed

Table 6.11: Test Case for API Latency Monitoring (ID: TC-11)

Test Case ID	TC-11
Test Case Name	API Latency Monitoring
Related Use Case	Gemini & Appwrite API response time
Pre-Condition	APIs reachable
Test Steps	1. Send repeated API requests to Appwrite and Gemini 2. Measure average and p95 latency
Expected Result	Average API latency within expected limits, p95 acceptable
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Latency reports generated

Table 6.12: Test Case for Stress Test Until Failure (ID: TC-12)

Test Case ID	TC-12
Test Case Name	Stress Test Until Failure
Related Use Case	Find breaking point
Pre-Condition	Monitoring in place
Test Steps	1. Gradually increase concurrent users until system errors increase 2. Record the threshold and failure modes
Expected Result	Documented capacity limit and graceful degradation behavior
Actual Result	System performance degraded as concurrent users increased, No graceful degradation
Pass/Fail	Fail
Post-Condition	Failure threshold data recorded

Table 6.13: Test Case for Access Control Enforcement (ID: TC-13)

Test Case ID	TC-13
Test Case Name	Access Control Enforcement
Related Use Case	Protect user data
Pre-Condition	Multiple user accounts exist
Test Steps	1. Log in as User A 2. Attempt to access User B's report via URL manipulation or API 3. Observe response
Expected Result	Access denied (403) and no data leakage
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Unauthorized access blocked

Table 6.14: Test Case for Session Expiry and Logout (ID: TC-14)

Test Case ID	TC-14
Test Case Name	Session Expiry and Logout
Related Use Case	Session management
Pre-Condition	User logged in
Test Steps	1. Remain idle until session timeout 2. Attempt to access protected resource 3. Click logout manually
Expected Result	Session expires and requires re-authentication, logout invalidates token
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	No active session

Table 6.15: Test Case for Dashboard Clarity and Navigation (ID: TC-15)

Test Case ID	TC-15
Test Case Name	Dashboard Clarity and Navigation
Related Use Case	User views and acts on recommendations
Pre-Condition	User has at least one analysis report
Test Steps	1. Open dashboard 2. Find analysis summary, detailed recommendations, and download button 3. Attempt to apply a suggested change (manual flow)
Expected Result	Elements are easy to find, instructions clear, download works
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Users can follow recommended steps

Table 6.16: Test Case for Progress & Error Messaging (ID: TC-16)

Test Case ID	TC-16
Test Case Name	Progress and Error Messaging
Related Use Case	Upload and analysis feedback
Pre-Condition	Upload in progress or error scenario triggered
Test Steps	1. Upload a large file or induce a parsing error 2. Watch for progress indicators and error messages
Expected Result	Progress is shown, errors are clear and actionable (e.g., 'File too large' or 'Parsing failed')
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	User informed of status

Table 6.17: Test Case for Appwrite Storage + Database Consistency (ID: TC-17)

Test Case ID	TC-17
Test Case Name	Appwrite Storage + Database Consistency
Related Use Case	File and metadata storage
Pre-Condition	Appwrite project configured
Test Steps	1. Upload file 2. Verify file exists in storage 3. Verify metadata and analysis record exist in database and reference file ID
Expected Result	Storage and DB records consistent and linked
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Data integrity confirmed

Table 6.18: Test Case for Gemini API Error Handling (ID: TC-18)

Test Case ID	TC-18
Test Case Name	Gemini API Error Handling
Related Use Case	AI integration robustness
Pre-Condition	Gemini API key active; simulate API failure or timeout
Test Steps	1. Simulate Gemini API timeout or 5xx error 2. Observe Appwrite function and frontend behavior
Expected Result	Request retries or graceful failure with user-friendly message, no unhandled exceptions
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Errors handled gracefully

Table 6.19: Test Case for Cross-Browser Functionality (ID: TC-19)

Test Case ID	TC-19
Test Case Name	Cross-Browser Functionality
Related Use Case	Front-end compatibility
Pre-Condition	App deployed to a public URL
Test Steps	1. Open app in Chrome, Firefox, Edge, Safari 2. Execute main flows (register, upload, analyze, download)
Expected Result	Core flows work similarly across supported browsers
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Browser compatibility matrix updated

Table 6.20: Test Case for Responsive Layout on Mobile (ID: TC-20)

Test Case ID	TC-20
Test Case Name	Test Case for Responsive Layout on Mobile
Related Use Case	Mobile usability
Pre-Condition	App responsive styles included
Test Steps	1. Open app on various screen sizes and mobile devices 2. Check layout, buttons, upload flow, and report readability
Expected Result	Interface adapts and remains usable on mobile and tablets
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	Responsive issues logged

Table 6.21: Test Case for AI Output Consistency (Repeatability) (ID: TC-21)

Test Case ID	TC-21
Test Case Name	AI Output Consistency (Repeatability)
Related Use Case	Stable AI recommendations
Pre-Condition	Same resume used multiple times
Test Steps	1. Run identical resume through analysis 3 times in different short intervals 2. Comparing key outputs: ATS score, main suggestions, keyword list
Expected Result	Outputs are consistent or show explainable minor variations
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	AI repeatability assessed

Table 6.22: Test Case for AI Relevance and Precision (ID: TC-22)

Test Case ID	TC-22
Test Case Name	AI Relevance and Precision
Related Use Case	Quality of recommendations
Pre-Condition	Resumes from different industries available
Test Steps	1. Analyze resumes for Software, Marketing, and Finance 2. Review keyword suggestions and industry-specific tips against known job descriptions
Expected Result	Suggestions align with industry expectations and improve keyword-match for job descriptions
Actual Result	As expected
Pass/Fail	Pass
Post-Condition	AI relevancy evaluated

Chapter 7

Conclusions

7.1 Conclusion

The AI-Based Resume Analyzer was able to fulfill its main objective, which implied creating an intelligent, user-friendly, and scalable web based system that helps job applicants enhance their resumes by automated AI analysis. The system has real-time feedback on resume quality, keyword optimization, grammar, and Applicant Tracking System (ATS) compatibility by leveraging the Gemini API. The solution of using Appwrite as a backend platform has removed the concept of having a traditional server that has to be managed and has introduced secured authentication, structured data handling, and the ability to have all the components of the system to interact without any issues.

The architecture of the project, which was created using the current web technologies, React.js on the frontend side and Appwrite Cloud Functions on the backend side, also made the system modular, responsive, and easily maintainable. During the development, scalability, accessibility by users, and data security were given high priority. The system was also put to test in various resume formats, file types and job sectors to make sure that the AI-based analysis generates uniform and reliable feedback.

Using ATS simulation module is a significant improvement over the traditional types of platforms of matching keywords. Rather than just pointing out the words or phrases that are missing, the analyzer assesses the organization, legibility, and professionalism of resumes using sophisticated NLP-based logic. This will make the feedback that is given to the users well matched with the way the real-world recruitment systems process and screen the applications. The findings can be used to prove that the application of AI and machine learning technologies to resume analysis can greatly enhance the general procedure of filling a job position. The system assists the user in producing high industry and recruiter expectations professional and targeted resumes through automating complex content

analysis. Moreover, the effective implementation of this system provides an excellent basis to further AI-based career-support products, indicating the opportunity of integrating cloud-based computer applications, natural language processing, and automation to enhance the employability performance.

7.2 Future Work

Although the AI-Based Resume Analyzer may reach its objectives, the further development and improvement of the subsequent versions still have much space. The system should evolve to become pertinent and useful as the process of hiring, employee demands, and AI technologies are constantly evolving. It can also be enhanced in its utility through broadened scalability, enhanced intelligent automation, and improved user experience. More collaboration and integration with external recruiting tools in the industry and absorption of emerging trends in AI research could all be developed in the future. Such advancements would increase the scope of the system in the hiring and career development system as well as enhance its analytical skills. It is suggested that future work should be improved in the following ways:

7.2.1 Enhanced Machine Learning Capabilities

The next generation can have domain specific and refined AI models that work with bigger amounts of resumes and job descriptions. It would enable the presentation of more context-sensitive recommendations, including rating soft skills, tone, and consistency of career progression. The inclusion of self-learning algorithms may allow the system to change automatically with the changing trends of hiring.

7.2.2 Integration with Job Portals and LinkedIn

To provide users with the opportunity to automatically connect their resumes with available job opportunities, it would be beneficial to integrate the analyzer with big job boards such as LinkedIn, Indeed, or Glassdoor. This would establish an entire ecosystem, resume optimization to job recommendation in one platform.

7.2.3 Improved User Interface and Personalization

Individualization AI-powered dashboards, tracking progress, and customized recommendations may enhance the user experiences. Defining how the system might leverage the data analytics to track the improvements of the users over a period of time and provide the access to the custom-made learning paths or the fields of the course requiring improvement may be proposed.

7.3 Comparison With Existing Platforms

The following tables show the comparison between our system and existing platforms. This includes the comparison of both things, the feature present in our system and absent in the existing ones, as well as the features absent in our system but present in the existing ones.

Table 7.1: Core Feature Comparison Between Our System and Existing Platforms

Feature	Our System	Jobscan	Resume.io	ResumeGenius	Zety	Resume.com
AI contextual analysis	Yes	No	No	No	No	No
ATS simulation	Yes	Yes (basic)	No	No	No	No
Industry-specific profiles	Yes	No	No	No	No	No
Resume quality checks (grammar, keywords, structure)	Yes	Yes	Yes	Yes	Yes	Yes
PDF parsing	Yes	Yes (partial)	Yes (partial)	Yes (partial)	Yes (partial)	Yes (partial)
Real-time feedback	Yes	No	No	No	No	No
Job description matching	Yes	Yes (basic)	No	No	No	No

Table 7.2: Features Present in Competitor Platforms but Not in Our System

Feature (Competitor Advantage)	Our System	Jobscan	Resume.io	ResumeGenius	Zety	Resume.com
Resume templates & design builder	No	No	Yes	Yes	Yes	Yes
Cover letter builder	No	No	Yes	Yes	Yes	Yes
Multiple resume format exports (PDF/DOCX/TXT/HTML)	No	Yes	Yes	Yes	Yes	Yes

References

- [1] J. Sánchez-Monedero, L. Dencik, and L. Edwards. What does it mean to 'solve' the problem of discrimination in hiring?: social, technical and legal perspectives from the uk on automated hiring systems. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2019. Cited on pp. [1](#), [4](#), and [7](#).
- [2] K. L. Abhishek, M. Niranjana Murthy, Shonit Aric, Syed Immamul Ansarullah, Anurag Sinha, G. Tejani, and Mohd Asif Shah. Developing an intelligent resume screening tool with ai-driven analysis and recommendation features. *Applied AI Letters*, 2025. Cited on pp. [2](#), [4](#), [5](#), and [8](#).
- [3] Aslam B Nandyal and Mohammed Rafi. Determining feature gaps of open source cloud platforms for mobile backend as service (mbaas) in enterprise mobile applications. *2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pages 200–205, 2020. Cited on pp. [3](#), [10](#), and [20](#).
- [4] Yong Luo, Huaizheng Zhang, Yongjie Wang, Yonggang Wen, and Xinwen Zhang. Resumenet: A learning-based framework for automatic resume quality assessment. *CoRR*, abs/1810.02832, 2018. Cited on pp. [4](#), [5](#), [6](#), and [7](#).
- [5] Oihab Allal-Chérif, Alba Yela Aránega, and Rafael Castaño Sánchez. Intelligent recruitment: How to identify, select, and retain talents from around the world using artificial intelligence. *Technological Forecasting and Social Change*, 169:120822, 2021. Cited on pp. [4](#) and [7](#).
- [6] Vidhita Jagwani, Smit Meghani, K. Pai, and Sudhir Dhage. Resume evaluation through latent dirichlet allocation and natural language processing for effective candidate selection. *ArXiv*, abs/2307.15752, 2023. Cited on pp. [5](#) and [6](#).
- [7] Changmao Li, Elaine Fisher, Rebecca Thomas, Steve Pittard, Vicki Hertzberg, and Jinho D. Choi. Competence-level prediction and resume & job description matching using context-aware transformer models. *CoRR*, abs/2011.02998, 2020. Cited on pp. [5](#) and [6](#).
- [8] Sihang Li, Kuangzheng Li, and Haibing Lu. National origin discrimination in deep-learning-powered automated resume screening. *ArXiv*, abs/2307.08624, 2023. Cited on pp. [6](#) and [10](#).
- [9] Mehtap Saatci, Rukiye Kaya, and Ramazan Ünlü. Resume screening with natural language processing (nlp). *Alphanumeric Journal*, 2024. Cited on p. [6](#).

- [10] Atharva Joshi, Shailaja V. Pede, Prathmesh Kolekar, and Aditya Metkar. Resume parsing and skill extraction using custom pattern matching algorithm and gemini api. *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pages 1–6, 2024. Cited on pp. 7, 9, 11, and 14.
- [11] Yinjin Xu. Evaluation of the impact of privacy-enhancing technologies on data security compliance in regulated frameworks. *International Journal of High Speed Electronics and Systems*, 2025. Cited on pp. 10, 14, and 20.
- [12] J. Cheoh, Behzad Beigpourian, Siqing Wei, D. Ferguson, and M. Ohland. Examining the perceptions of people with disabilities on the use of accessibility standards in web interface design. *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–4, 2020. Cited on p. 11.
- [13] Vijay Bhasker Reddy Bhimanapati, Pandi Kirupa Gopalakrishna Pandian, and Prof.(Dr.) Punit Goel. Ui/ux design principles for mobile health applications. *International Journal for Research Publication and Seminar*, 2024. Cited on pp. 11 and 13.
- [14] Appwrite. Appwrite documentation. <https://appwrite.io/docs>. Cited on pp. 12, 13, 14, and 17.
- [15] Vercel. Vercel documentation. <https://vercel.com/docs>. Cited on p. 14.

Appendix A

User Manual

A.1 Introduction

This user manual provides detailed instructions for using the AI-Based Resume Analyzer. The system is designed to assist job seekers in evaluating and improving their resumes using artificial intelligence. It analyzes resume content, structure, keyword relevance, and overall compatibility with Applicant Tracking Systems (ATS), and provides actionable recommendations.

This manual is intended for end users (job seekers) with basic computer and web browser knowledge. No prior technical expertise is required.

A.2 System Overview

The AI-Based Resume Analyzer is a web-based application consisting of a React.js frontend for user interaction, an Appwrite backend for authentication, database management, file storage, and serverless functions, and the Gemini AI model for resume content analysis and recommendation generation.

Users interact with the system through a web browser to upload resumes, trigger analysis, and view results.

A.3 System Requirements

A.3.1 Hardware Requirements

- Desktop, laptop, tablet, or smartphone
- Minimum 2 GB RAM (4 GB recommended)

A.3.2 Software Requirements

- Modern web browser, preferably updated versions:
 - Google Chrome
 - Mozilla Firefox
 - Microsoft Edge
 - Safari
- Stable internet connection

A.4 User Roles

The **Job Seeker** is the primary user of the system. This role allows users to:

- Register and log into the system
- Upload resumes for analysis
- View AI-generated feedback and recommendations
- Download optimized resumes
- Manage account and subscription details

A.5 Accessing the System

1. Open a supported web browser.
2. Navigate to the deployed application URL.
3. Choose to register a new account or log in.

A.6 User Registration

1. Click **Register** on the homepage.
2. Enter full name, email address, and password.
3. Submit the registration form.
4. Verify the account using the link sent to the registered email.

A.7 User Login

1. Click **Log In**.
2. Enter registered email and password.
3. Submit credentials.

Successful authentication redirects the user to the dashboard.

A.8 Dashboard Overview

The dashboard provides access to resume uploads, previously analyzed resumes, analysis status indicators, downloadable reports, and account management options.

A.9 Resume Upload

A.9.1 Supported Formats

- PDF

A.9.2 Upload Procedure

1. Click **Upload Resume**.
2. Select a supported file (maximum size: 10 MB).
3. Confirm upload.

A.10 Resume Analysis

After uploading the resume:

1. Click **Analyze Resume**.
2. The system displays a progress indicator.
3. AI processing is performed using the Gemini model.

The analysis evaluates grammar, structure, keyword relevance, ATS compatibility, and industry alignment.

A.11 Viewing Analysis Results

Once analysis is complete, users can view:

- Resume evaluation summary
- Strengths and weaknesses
- ATS compatibility feedback
- Keyword optimization suggestions
- Section-wise improvement recommendations

A.12 Implementing Recommendations

Users can manually apply suggested improvements using an external resume editor. The system provides guidance and comparison support while maintaining full user control over final content.

A.13 Downloading Reports

1. Open the analysis report.
2. Click **Download**.
3. Save the report locally.

A.14 Account Management

Users can view profile details and securely log out from the system.

A.15 Error Handling and Feedback

The system provides clear messages for unsupported file formats, upload failures, analysis delays, and authentication issues. Progress indicators inform users about processing status.

A.16 Security and Privacy

All communication uses HTTPS with token-based authentication. Resume files and reports are stored securely with user-specific access control. Sensitive data is encrypted both in transit and at rest.

A.17 Logout Procedure

1. Click **Logout** from the dashboard.
2. The session is terminated immediately.

A.18 System Limitations

- Optimized for English-language resumes.
- ATS scores are indicative, not official.
- Continuous internet access is required.

A.19 Support and Troubleshooting

Users are advised to refresh the browser, re-login, or re-upload resumes if issues occur. Persistent issues should be reported to the system administrator.