

JOB CATALYST

A Project Report
Submitted

In Partial Fulfillment of the Requirements
For the Degree of

Bachelor of Technology

in

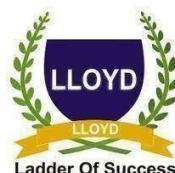
Computer Science & Engineering

by

Chayan Rai	Ashwani
2101530100051	2101530100039
Pushkar	Shruti Gupta
2101530100121	2101530100149

Riya Singh
2101530100134

Under the Supervision of
Dr. Harshita Tuli
Assistant professor



Lloyd Institute of Engineering and Technology
GREATER NOIDA



DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY,
UTTAR PRADESH, LUCKNOW
MAY 2025

Declaration

We hereby declare that the project work presented in this report entitled “JOB CATALYST”, in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering, submitted to Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh, Lucknow is based on our own work carried out at Department of Computer Science & Engineering, Lloyd Institute of Engineering and Technology, Greater Noida. The work contained in the report is true and original to the best of our knowledge and project work reported in this report has not been submitted by us for award of any other degree or diploma.

Signature:

Name: Chayan Rai

Roll No: 2101530100051

Signature:

Name: Ashwani

Roll No: 2101530100039

Signature:

Name: Pushkar

Roll No: 2101530100121

Signature:

Name: Shruti Gupta

Roll No: 2101530100149

Signature:

Name: Riya Singh

Roll No: 2101530100134

Date:

Place: Greater Noida

Certificate

This is to certify that the Project report entitled "**JOB CATALYST**" done by **Chayan rai (2101530100051), Ashwani (2101530100039), Pushkar (2101530100121), Shruti Gupta (2101530100149) and Riya Singh (2101530100134)** is an original work carried out by them in Department of Computer Science & Engineering, Lloyd Institute of Engineering and Technology, Greater Noida under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Date:

Dr. Harshita Tuli

Signature of the Supervisor

Prof. Dr. A. Kakoli Rao

Head of the Department

Acknowledgement

The merciful guidance bestowed to us by the almighty made us stick out this project to a successful end. We humbly pray with sincere heart for his guidance to continue forever.

We pay thanks to our project guide **Dr. Harshita Tuli** who has given guidance and light to us during this project. His/her versatile knowledge has helped us in the critical times during the span of this project.

We pay special thanks to our Head of Department **Prof (Dr.) A. Kakoli Rao**, who has been always present as a support and help us in all possible way during this project.

We also take this opportunity to express our gratitude to all those people who have been directly and indirectly with us during the completion of the project.

We want to thanks our friends who have always encouraged us during this project.

At the last but not least thanks to all the faculty of CSE department who provided valuable suggestions during the period of project.

Abstract

The evolving needs of students in a **competitive job** market demand **smarter tools** and **personalized guidance**. “**Job Catalyst**” is a modern **web-based platform** built with **Next.js**, **Tailwind CSS**, **Neon DB**, **Prisma ORM**, **Gemini API**, and **Clerk authentication** aimed at **bridging the gap** between **academic learning** and **industry preparedness**.

The platform is an **all-in-one career preparation hub** that enables students to:

- **Build ATS-friendly resumes**
- **Generate AI-powered cover letters**
- **Attempt mock interviews**
- **Receive performance analysis**
- **Gain real-time industry insights**

What sets **Job Catalyst** apart is its integration of **AI** to **personalize** each student's **career development journey**. The use of **Gemini API** powers **AI-driven features** such as **cover letter generation**, **resume enhancement**, and **smart recommendations**, providing **personalized** and **context-aware career support**, while Clerk provides secure, scalable authentication. Backend architecture ensures high scalability and fast performance with Neon DB and Prisma.

This project highlights the fusion of modern frontend frameworks and powerful backend technologies to build a system that not only facilitates student growth but also evolves with real-time data. The aim is to provide an intuitive, feature-rich, and interactive experience that empowers users to confidently step into the professional world.

Keywords: AI, Resume Builder, Mock Interviews, Performance Analysis, ATS, Gemini API, Clerk, Next.js, Career Platform

Table of Content

Declaration.....	2
Certificate	3
Acknowlegement	4
Abstract.....	v
Table of Content	6-7
List of Figures.....	8
List of Tables.....	9
List of Abbreviations.....	10
Chapter 1. Introduction	11-15
1.1 Preliminaries.....	11
1.2 Motivation.....	12
1.3 Project Overview/Specifications.....	13
1.4 Aim and Objectives.....	15
Chapter 2. Literature Survey	16-20
2.1 Introduction	16
2.2 Existing System	17
2.3 Benefits of the Project	19
Chapter 3. Proposed Methodology.....	21-26
3.1 Problem Formulation	21
3.2 System Analysis & Design.....	22
3.3 Proposed Work.....	24

Chapter 4. Implementation	27-39
4.1 Introduction.....	27
4.2 Implementation Strategy	28
4.3 Tools/Hardware/Software Requirements	38
4.4 Expected Outcome	39
Chapter 5. Result & Discussion	40-43
5.1 Discussion	40
Chapter 6. Conclusion & Future Scope	44-45
6.1 Conclusion & Future Scope.....	44
References	46
Appendix I: Plagiarism Report of Project Report (<=15%)	47

LIST OF FIGURES

 Figure No.	 Description
Figure 1.3.1	System Overview Diagram
Figure 2.3.1	Job Catalyst Ecosystem Diagram
Figure 3.2.1	Job Catalyst Functional Flow
Figure 3.3.1	User Flow for Exploring Features in Job Catalyst
Figure 5.1.1	Sign-In Page
Figure 5.1.2	Landing Page
Figure 5.1.3	Cover Letter Page
Figure 5.1.4	Resume Builder Page
Figure 5.1.5	Interview Prep Page

LIST OF TABLES

 Table No.	 Description
Table 2.2.1	Comparison of Existing Career Readiness Platforms vs. Job Catalyst
Table 4.3.1	Technology Stack for Job Catalyst
Table 5.1.1	Key Functional Outcomes

LIST OF ABBREVIATION

 Abbreviation	 Full Form	 Where it Appeared
ATS	Applicant Tracking System	Introduction, Resume Builder
AI	Artificial Intelligence	Cover Letter, Interview, Analytics
API	Application Programming Interface	Gemini API, Clerk API
UI	User Interface	Tailwind UI, Frontend Section
ORM	Object Relational Mapping	Prisma ORM Usage
DB	Database	NeonDB, Prisma Database Layer
PDF	Portable Document Format	Resume & Cover Letter Export
MCQ	Multiple Choice Questions	Interview Mock Section
OAuth	Open Authorization	Clerk Authentication
JSON	JavaScript Object Notation	Gemini Output Formatting
SSR	Server-Side Rendering	Next.js Framework
RAM	Random Access Memory	Hardware Requirements Section

Chapter 1: Introduction

1.1 Preliminaries

In the current competitive landscape, the demand for **industry-ready graduates** has grown significantly. Employers are seeking candidates who not only possess academic knowledge but also demonstrate practical skills, professional communication, and adaptability to modern recruitment practices. However, a substantial number of students continue to face challenges in transitioning from academia to the corporate world.

One of the key obstacles is the **lack of structured, personalized support** in areas such as **resume building, cover letter writing, and interview preparation**. Although most educational institutions provide placement assistance through their placement cells, these efforts are often **generic in nature** and fail to address the **individual career development needs** of students. Furthermore, the **rapid evolution of recruitment processes**, especially with the widespread adoption of **Applicant Tracking Systems (ATS)**, has introduced a new layer of complexity. Resumes are now evaluated algorithmically, making it essential for students to submit **ATS-optimized documents** that align with specific job roles and keyword criteria [1].

In parallel, advancements in **Artificial Intelligence (AI)** have revolutionized content generation, user personalization, and performance evaluation. These technologies provide a unique opportunity to develop intelligent platforms that can offer **real-time, customized career guidance** [2].

Recognizing this gap between student capabilities and industry expectations, as well as the transformative potential of AI technologies, this project introduces **Job Catalyst**—a comprehensive, web-based platform designed to enhance student career readiness. By integrating advanced technologies such as the **Gemini API** [2] for AI-driven content generation and **Clerk** [3] for secure user authentication, along with scalable backend systems powered by **Neon DB** and **Prisma ORM** [4], the platform offers a seamless and intuitive experience. Students are empowered to build **ATS-compliant resumes**, generate **tailored cover letters**, engage in **mock interviews**, and receive **automated performance feedback**, all within a single ecosystem.

The development of Job Catalyst marks a significant step toward addressing the limitations of traditional career support models by leveraging technology to provide **accessible, personalized, and effective career preparation tools** for every student.

1.2 Motivation

As a final-year engineering student preparing to transition into the professional world, I have personally encountered the **challenges and limitations** in the tools and guidance available for **resume building, interview readiness, and personalized performance analysis**. Despite the presence of widely used platforms such as **LinkedIn**, online **job portals**, and various resume creators, the services they offer are often **fragmented, generic**, and not tailored to the **unique needs of students**.

There remains a clear lack of a unified platform that specifically addresses:

- **Instant, intelligent resume analysis** based on ATS standards [1].
- **Context-aware cover letter generation** that reflects a candidate's strengths and role-specific relevance [2].
- **Skill-specific mock interviews**, dynamically curated based on the user's input
- **Detailed performance analytics** that help students identify areas for improvement and build confidence [5].
- **Real-time industry insights** to keep students updated with emerging job trends, in-demand skills, and sector-specific opportunities [6].

These limitations, coupled with the increasing use of **AI** in education and employment tools [2], inspired me to **conceptualize and build a solution from the ground up**—a platform that not only simplifies but also **enhances** the career preparation journey for students.

This led to the creation of “**Job Catalyst**”, an AI-powered, student-focused web application that consolidates all essential career-building tools in a **single, intuitive platform**. Unlike generic services, Job Catalyst is designed **by a student, for students**, with the aim of making career readiness **smarter, faster, and more personalized**, while keeping students informed about the **current demands of the job market**.

1.3 Project Overview / Project Specifications

Job Catalyst is a full-stack, AI-powered career assistance web application designed to support students in preparing for professional opportunities with confidence and precision. Built with modern technologies, the platform aims to bridge the academic-to-industry transition by offering personalized, intelligent, and real-time career development tools.

Technology Stack

- **Next.js** – Utilized for frontend development and client-side routing, enabling fast rendering and improved performance through server-side capabilities [7].
- **Tailwind CSS** – Powers the application's responsive and clean UI design, ensuring an intuitive user experience across devices [8].
- **NeonDB & Prisma ORM** – NeonDB serves as a scalable cloud-based PostgreSQL database, while Prisma ORM offers efficient and type-safe interaction between the application and the database layer [9].
- **Gemini API** – Integrates advanced AI capabilities to generate context-aware cover letters, analyze resume data, and support personalized feedback mechanisms [2].
- **Clerk** – Handles user authentication and session management, ensuring secure, scalable access control with seamless onboarding [3].

Core Functional Modules

1. Resume Builder

- Provides **ATS-optimized templates** [1]
- Supports **LinkedIn data import** for auto-filling content
- Offers **real-time resume improvement suggestions**

2. AI Cover Letter Generator

- Generates context-relevant, professional cover letters
- Takes **user input and job details** as prompt
- Powered by the **Gemini API** for coherent and personalized outputs [2]

3. Mock Interview System

- Contains **objective (MCQ)** and **subjective** questions
- Offers **skill-based mock interviews**
- Delivers **AI-driven feedback and scoring**

4. Performance Tracker

- Displays visual analytics (graphs and charts) of user progress [10]
- Shares **personalized tips**, resume scores, and interview metrics

5. Industry Insights Feed

- Curates **real-time trends and job demands** from various industries [6]
- Personalized based on user-selected **fields of interest**
- Helps users stay informed about **emerging technologies and skills**

Platform Features

- Fully responsive design for desktops, tablets, and mobiles [8]
- Real-time AI suggestions throughout the resume and interview process [2]
- Secure authentication with Clerk and session handling [3]
- User-centric dashboard with a personalized and modular layout

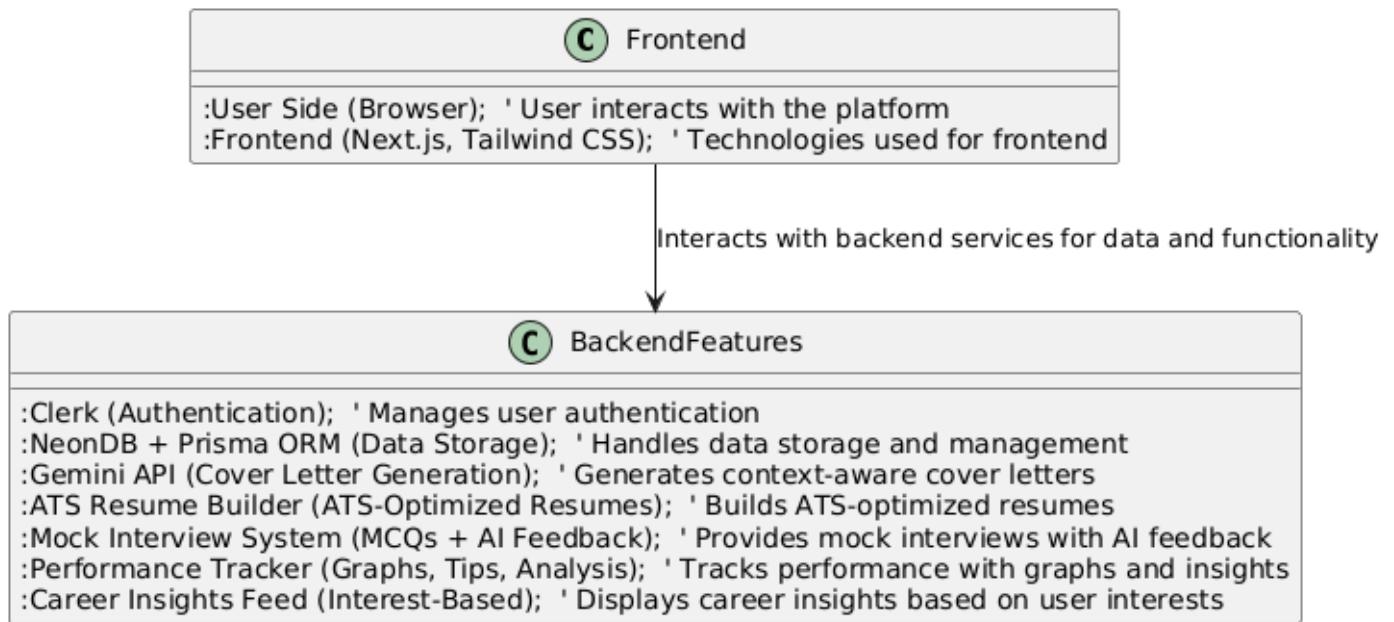


Figure 1.3.1 System Overview Diagram

1.4 Aim and Objectives

Aim:

To develop an AI-integrated platform that not only guides students through resume creation, cover letter generation, interview preparation, and post-performance analysis, but also provides valuable industry insights, trends, and salary information, equipping students with the tools and knowledge to make informed career decisions.

Objectives:

- **Design an intuitive and student-centric interface** that simplifies and enhances the process of career preparation, ensuring ease of use and accessibility for all users [7] [8].
- **Develop a secure, scalable, and high-performance web application** using modern technologies such as Next.js, Prisma ORM, and Tailwind CSS to ensure a robust user experience [7] [9] [8].
- **Integrate the Gemini API** to automate cover letter generation and provide personalized performance analysis, helping students craft impactful documents and gain insightful feedback [2].
- **Enable real-time feedback mechanisms** to evaluate resume structures and mock interview performance, offering actionable insights for continuous improvement [1] [2] [5].
- **Provide industry-specific career tips and job search guidance** tailored to the student's field of interest, helping them stay up-to-date with current trends and market demands [6].
- **Deliver real-time industry insights, including salary trends, job market updates, and the most in-demand skills**, empowering students to align their career aspirations with industry realities.
- **Offer salary trend analysis** for various job roles across industries, allowing students to understand compensation expectations and make informed decisions regarding their career paths.

Chapter 2: Literature Survey

2.1 Introduction

Over the past decade, career development platforms and resume-building tools have garnered significant attention as essential resources for individuals seeking to enter the professional world. The landscape is increasingly populated by AI-powered solutions, job portals, and other digital tools designed to aid users in crafting professional profiles. Despite the wide availability of these platforms, many are commercialized, overly complex, and fail to address the specific needs of students.

With the growing adoption of automation in recruitment processes, particularly through Applicant Tracking Systems (ATS), traditional resumes are becoming obsolete. As a result, students are faced with the challenge of navigating a competitive job market without the necessary tools to create effective, ATS-friendly resumes. To bridge this gap, there is a clear demand for platforms that can:

- Provide comprehensive, step-by-step guidance for optimizing resumes.
- Offer personalized, industry-specific suggestions that align with current market trends.
- Leverage AI to enhance and tailor every aspect of the job preparation process, from resume creation to interview practice.

This chapter delves into the existing literature and explores platforms that have laid the groundwork for **Job Catalyst**, while also identifying the persisting gaps in the market that the platform seeks to address.

2.2 Existing System

Several online platforms aim to facilitate career readiness, yet many fall short in offering a holistic, student-centred experience:

- **Zety & ResumeGenius:** These platforms provide user-friendly, drag-and-drop resume builders with customizable templates. However, they lack advanced Applicant Tracking System (ATS) optimization and fail to deliver personalized recommendations, which are crucial in today's competitive recruitment landscape.
- **Rezi:** Rezi is notable for its ATS tracking functionality, yet it is limited by the absence of mock interview preparation and personalized feedback features, both of which are essential for students actively preparing for job opportunities.
- **LinkedIn Resume Assistant:** While LinkedIn offers a basic resume-building tool, it is constrained in terms of flexibility and does not include dynamic AI-driven cover letter generation or integrated feedback systems, making it insufficient for comprehensive job preparation.
- **Other Platforms:** Websites such as Glassdoor, LeetCode, and InterviewBit offer fragmented career preparation services. While they provide job listings, interview preparation, or community-driven forums, none of these platforms offer a unified, student-first solution that integrates resume building, interview practice, and post-performance analysis into one seamless experience.

Key Drawbacks of Existing Systems:

- Reliance on manual input without context-aware suggestions or enhancements, which limits their adaptability to individual user needs.
- Absence of integration between mock interview systems and skill-specific tracking, preventing a comprehensive approach to interview readiness.
- Lack of personalized performance analytics tailored to track student progress and learning outcomes effectively.
- Limited interoperability between modules related to resume building, cover letter generation, and interview preparation, making the process disjointed and inefficient.
- Lack of performance analytics tailored to student learning levels
- Poor interoperability between resume, cover letter, and interview modules

Feature	Job Catalyst	Top Resume Builder (e.g., Novoresume)	Interview Prep (e.g., InterviewBit)	Cover Letter Generator (e.g., Jobscan)
ATS Optimization	Yes	Yes (Novoresume)	No	No
Personalized Suggestions	Yes	Yes (Novoresume)	Yes (InterviewBit)	Yes (Jobscan)
Cover Letter Generation	Yes	No	No	Yes (Jobscan)
Mock Interviews	Yes	No	Yes (InterviewBit)	No
Performance Analytics	Yes	No	Yes (InterviewBit)	No
Industry Insights	Yes	No	No	No

Table 2.2.1 Comparison of Existing Career Readiness Platforms vs. Job Catalyst

2.3 Benefits of the Project

Job Catalyst addresses the limitations of existing career readiness platforms by offering a comprehensive, AI-driven solution tailored specifically for students. The platform goes beyond traditional tools, providing a unified, seamless, and personalized experience that supports students through every stage of their career journey. The key benefits include:

Unified Career-Building Experience

Job Catalyst integrates all core features—resume building, cover letter generation, mock interview preparation, and performance analytics—into a cohesive platform. This creates a seamless workflow that mirrors real-world job application processes, ensuring that students don't have to juggle multiple platforms.

AI-Powered Content Generation

Leveraging the Gemini API, Job Catalyst provides:

- **Tailored Cover Letters:** Automatically generates cover letters by analyzing resumes and job descriptions, ensuring personalized, job-specific content.
- **Smart Interview Questions:** Generates interview questions based on selected skills and job requirements to simulate realistic interview scenarios.
- **Performance Analysis:** Provides detailed performance feedback after mock interviews and tests, with actionable improvement suggestions.

Secure and Seamless Authentication

Job Catalyst uses **Clerk** for secure, easy sign-up and login processes. Session management is robust, ensuring users' data is kept safe. The platform also supports extended admin dashboard capabilities for training coordinators to manage and monitor students' progress.

Data-Driven Performance Insights

Job Catalyst helps students track their progress and improve by offering:

- **Visual Dashboards:** Displays performance metrics, giving students a clear view of their strengths and areas for improvement.
- **Mock Test Results:** Helps identify weaknesses and suggests focused improvement areas based on test outcomes.
- **Milestone Achievements:** Encourages growth through a gamified approach with milestone-based achievements and progress tracking.

Modern and Scalable Tech Stack

Job Catalyst is built on a fast, responsive, and scalable tech stack, ensuring a smooth user experience:

- **Next.js** (for frontend rendering and routing)
- **Tailwind CSS** (for modern, responsive UI design)
- **NeonDB + Prisma ORM** (for data management)

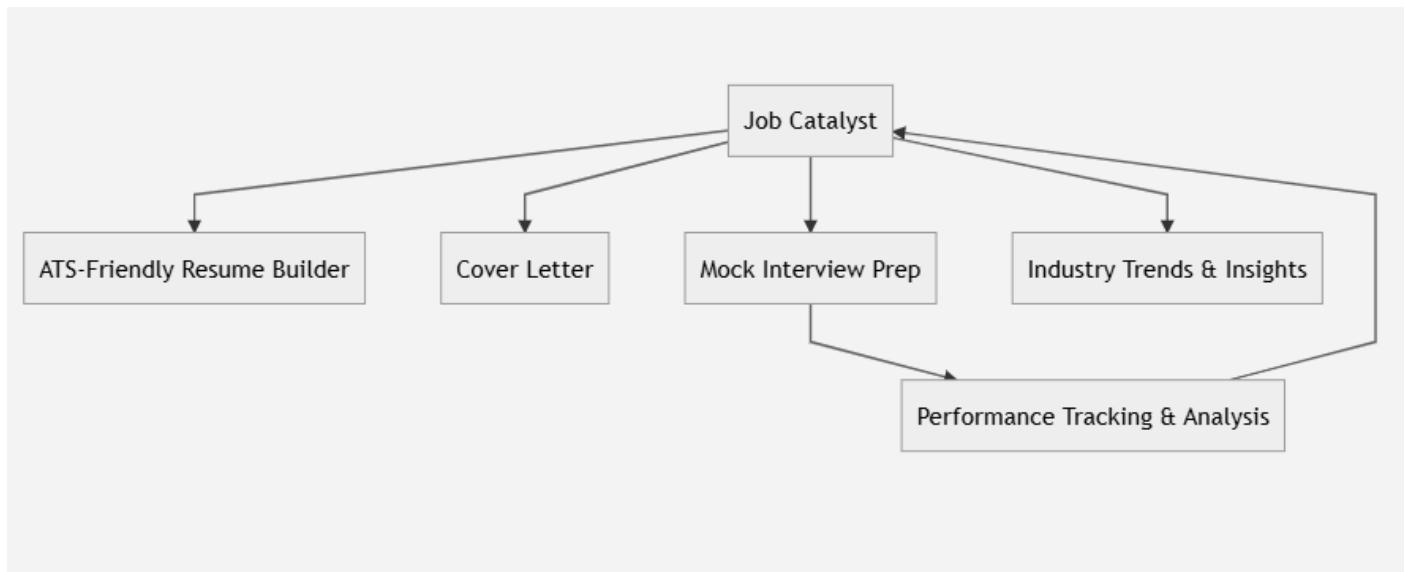


Figure 2.3.1 Job Catalyst Ecosystem Diagram

Chapter 3: Proposed Methodology

3.1 Problem Formulation

In today's highly competitive job market, students face numerous challenges as they navigate their career preparation journey, including:

- Creating ATS-friendly resumes that meet industry standards.
- Writing tailored, impactful cover letters for each job application.
- Preparing for interviews that align with their chosen career paths.
- Analysing and understanding performance metrics and feedback to improve their job-readiness.

Although there are numerous tools available to address these challenges, many are fragmented, overwhelming, or require expensive subscriptions. Most platforms only offer isolated solutions, without a unified approach or real-time guidance, leaving students without the holistic support they need.

Core Problem Statement: To develop a comprehensive, AI-powered platform that seamlessly integrates all aspects of job preparation — from resume building to cover letter generation, interview readiness, and performance tracking — empowering students to enhance their job-readiness with a single, easy-to-use, and cost-effective solution built with modern web technologies.

3.2 System Analysis & Design

Functional Requirements

The system must deliver a set of essential features to ensure a seamless, efficient, and personalized experience for the user. These functionalities include:

- **Resume Creation & Editing:** Users should be able to create, edit, and customize their resumes to be ATS-friendly, with smart suggestions for optimization based on job market standards.
- **AI-Powered Cover Letter Generation:** The platform should dynamically generate personalized cover letters using AI, based on user-provided information (like resume data and job descriptions) to make each application unique and tailored.
- **Mock Interview Preparation:** Users should be able to take mock interviews, which could include both multiple-choice questions (MCQs) and subjective questions, with AI-driven feedback on their answers to simulate a real-life interview scenario.
- **Personalized Feedback & Analytics:** After completing mock interviews or resume submissions, users should receive comprehensive performance analysis, including strengths, areas for improvement, and actionable suggestions to enhance their profile and interview readiness.
- **Secure Authentication:** The system should allow users to securely sign up and log in through Clerk for smooth session management, ensuring privacy and security of personal data.

Non-Functional Requirements

In addition to the core functionalities, the system should adhere to the following non-functional requirements to ensure a high-quality user experience:

- **Responsive & Intuitive User Interface (UI):** The platform should provide an easy-to-use interface that adapts seamlessly to any device (desktop, tablet, mobile) and ensures a smooth user journey.
- **Fast Query Handling & Data Management:** The system should leverage **NeonDB** for fast data storage and retrieval, enabling quick response times even when handling large datasets, like resumes, cover letters, and user performance history.
- **Scalable & Low-Latency System:** The use of **Next.js** ensures high scalability and performance, with low latency during user interactions. The platform should be able to handle an increasing number of users without compromising speed or quality.
- **Seamless Integration Across Modules:** All the modules (resume builder, cover letter generator, mock interview system, and analytics dashboard) should be integrated to provide a fluid experience. Users should be able to navigate between different modules with minimal friction, making their journey from resume creation to interview preparation and performance analysis efficient and cohesive.
- **High Availability & Reliability:** The system should be designed for high availability, ensuring it is accessible to users at all times. It should also handle system errors gracefully, minimizing downtime and providing fallback mechanisms in case of failure.

By addressing these functional and non-functional requirements, **Job Catalyst** will deliver a comprehensive, user-friendly, and high-performance platform that empowers students to achieve success in the job market.

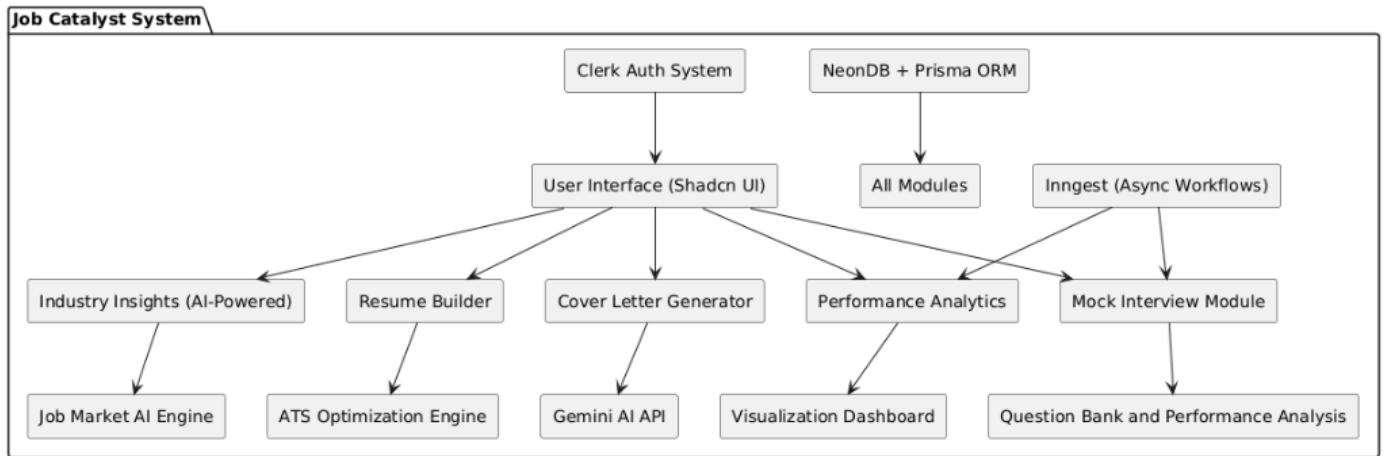


Figure 3.2.1 Job Catalyst Functional Flow

3.3 Proposed Work

The **Job Catalyst** platform is envisioned as a modular, scalable, and intelligent system focused on enhancing student employability through real-time feedback, AI-powered assistance, and seamless user experience. It is built using modern web technologies including **Next.js**, **NeonDB**, **Tailwind CSS**, **Prisma ORM**, **Shadcn UI**, **Inngest**, **Clerk**, and the **Gemini API**. The key components of the proposed system are:

Resume Builder Module

- **Input:** User-provided personal details, education, experience, and skills.
- **Output:** Professionally designed, **ATS-compliant resumes**.
- Built with **Shadcn UI** for clean form elements and real-time preview.
- Drag-and-drop section arrangement and editable templates.
- One-click **PDF export** for finalized resumes.

AI-Powered Cover Letter Generator

- Uses the **Gemini API** to generate cover letters based on the resume content and selected job descriptions.
- Features **real-time generation, editing, and version tracking**.
- Personalized content ensures alignment with the job and industry.
- Automatically adapts tone and format for Indian job market standards.

Mock Interview Preparation Module

- Users select **skill tags** (e.g., React, DBMS, Python).
- AI generates relevant **interview questions** dynamically.
- Responses accepted via **text**.
- Scoring logic integrates Gemini API + prompt engineering to give:
 - Real-time **performance ratings**.
 - Intelligent **feedback with improvement tips**.

Performance Analytics & Insights

- Tracks each user's mock test performance over time.
- Uses **Chart.js** to visualize data with **bar graphs, line charts, and radar diagrams**.
- Tracks **growth milestones** and builds a career progress report.

Authentication & User Management

- **Clerk** handles secure **email/password login, OAuth, and session management**.
- Role-based user access: Student, Admin, and Career Coach.
- Tracks user metadata including history, attempts, and preferences.

Market Insights & Industry Trends

- Uses **AI scraping models** to analyze **live job market trends**.
- Provides users with:
- **Salary benchmarks** based on role & location.
- **In-demand skillsets** and market shifts.
- Job role recommendations matching their resume content.

Event-Driven Automation (via Inngest)

- Background tasks like **email reminders, cover letter regeneration, and data backups** are handled using **Inngest**.
- Ensures reliability with **retry queues** and **fail-safe logic**.
- Helps decouple time-consuming processes from the main user flow, ensuring responsiveness.

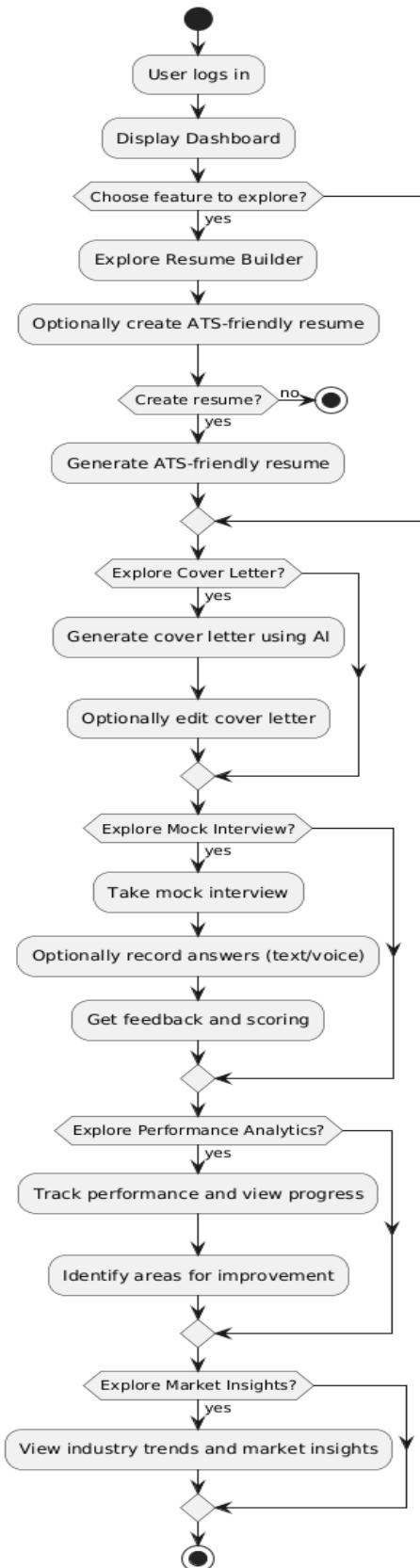


Figure 3.3.1 User Flow for Exploring Features in Job Catalyst

Chapter 4: Implementation

4.1 Introduction

The implementation phase of **Job Catalyst** emphasized a modular, scalable, and performance-oriented approach. Each core feature—**Resume Builder**, **AI-Powered Cover Letter Generator**, **Mock Interview Module**, and **Performance Analytics Dashboard**—was developed as an independent module to ensure maintainability and scalability. These modules were later integrated into a unified and seamless user workflow.

The platform leverages modern technologies to enhance both user experience and functionality:

- **Next.js** – A React-based framework enabling server-side rendering (SSR) and optimized performance.
- **Tailwind CSS** – For building a responsive, clean, and modern UI efficiently.
- **NeonDB + Prisma ORM** – NeonDB handles scalable database operations, while Prisma ORM simplifies schema modeling and data fetching.
- **Clerk** – Provides secure and streamlined user authentication, including OAuth and session management.
- **Gemini API** – Powers AI-driven features such as resume-tailored cover letter generation, intelligent interview question creation, and personalized performance feedback.

This modular architecture ensures that each feature functions independently while contributing to the overall ecosystem, resulting in a powerful and intuitive career support platform for students.

4.2 Implementation Strategy

1. Project Initialization

Setup Environment

- **Goal:** Set up the core structure of your Next.js app.
- **Theory:** To start the project, Next.js will be used for server-side rendering and static page generation. Tailwind CSS will help with styling, and Clerk will manage authentication. Prisma and NeonDB will handle the database connection.

Steps:

- **Initialize Next.js App:**

```
bash
➤ npx create-nextapp@latest job-catalyst
```

- **Install Essential Packages:**

```
bash
➤ npm install next react react-dom tailwindcss postcss autoprefixer @shadcn/ui
      @clerk/nextjs prisma @prisma/client axios recharts chart.js @react-
      pdf/renderer react-dnd react-dnd-html5-backend husky lint-staged eslint
      prettier inngest neon && npx prisma init
```

Configure Development Tools

- **Goal:** Set up a clean development environment with linters and formatters.
- **Theory:** For better code quality, ESLint and Prettier help enforce coding standards and formatting rules. Git hooks automate checks before committing.

Steps:

- **Setup .eslintrc and .prettierrc files.**
- **Add Prettier and ESLint hooks using Husky:**

2. Authentication & User Management (CLERK)

Integrate Clerk

- **Goal:** Implement user authentication using Clerk.
- **Theory:** Clerk provides seamless authentication and session management, allowing users to securely log in, sign up, and manage profiles.

Steps:

- **Set up ClerkProvider in _app.tsx:**

```
tsx

import { ClerkProvider, RedirectToSignIn } from '@clerk/nextjs';

function MyApp({ Component, pageProps }) {
  return (
    <ClerkProvider>
      <Component {...pageProps} />
    </ClerkProvider>
  );
}
```

- **Create protected routes with auth guards:**

```
tsx

import { withAuth } from "@clerk/nextjs";

const ProtectedRoute = () => {
  return (
    <div>
      <h1>Protected Content</h1>
    </div>
  );
}
```

User Schema

- **Goal:** Define the User model in Prisma to store user details.
- **Theory:** Using Prisma, define the user schema and link it to the Resume and Interview models to manage resumes and mock interviews.

Steps:

- **Define User model:**

```
ts

model User {
    id      String @id @default(cuid())
    email   String @unique
    name    String
    role    String @default("student")
    resumes Resume[]
    interviews Interview[]
    createdAt DateTime @default(now())
}
```

3. Resume Builder Module

Resume Input Form

- **Goal:** Build a form where users can input their resume data.
- **Theory:** The form will allow users to add their personal details, education, skills, and experience. You can also add dynamic sections for projects and experience.

Steps:

- **Build a dynamic form with drag-and-drop for reordering sections:**

```
tsx

<DragDropContext onDragEnd={onDragEnd}>
  <Droppable droppableId="droppable">
    {(provided) => (
      <div ref={provided.innerRef} {...provided.droppableProps}>
        {/* Form Fields */}
      </div>
    )}
  </Droppable>
</DragDropContext>
```

ATS Compliance Engine

- **Goal:** Ensure the generated resume is ATS-friendly.
- **Theory:** ATS (Applicant Tracking System) often filters resumes based on certain keywords and formatting. Implement an engine to ensure compliance.

Steps:

- **Convert the resume to PDF using react-pdf or html2pdf.js:**

```
tsx

import { PDFDownloadLink } from '@react-pdf/renderer';

const ResumePDF = () => (
  <PDFDownloadLink      document={<ResumeDocument      />}
    fileName="resume.pdf">
    {({ loading }) => (loading ? 'Loading document...' : 'Download
    PDF')}
  </PDFDownloadLink>
);
```

Save Resume to DB

- **Goal:** Save the resume in NeonDB for each user.
- **Theory:** Using Prisma, save the resume data in a JSON format, ensuring it's easy to retrieve later.

Steps:

- **Define Resume model:**

```
ts

model Resume {
    id      String  @id @default(cuid())
    userId  String  @unique // One resume per user
    user    User    @relation(fields: [userId], references: [id])
    content String  @db.Text // Markdown content
    atsScore Float?
    feedback String?
    createdAt DateTime @default(now())
    updatedAt DateTime @updatedAt
}export default withAuth(ProtectedRoute);
```

Cover Letter Generator (Gemini AI)

Input

- **Goal:** Collect the resume and job role/company information.
- **Theory:** You will collect user details, such as the job they are applying for, to personalize the cover letter.

Steps:

- **Input the details from the user:**

```
tsx
```

```
const handleCoverLetterGeneration = async () => {
  const response = await fetch("/api/generate-cover-letter", {
    method: "POST",
    body: JSON.stringify({ resume, role, company }),
  });
};
```

Gemini API Call

- **Goal:** Use Gemini AI to generate a cover letter based on the provided details.
- **Theory:** The Gemini API generates a tailored cover letter based on the user's resume and job role.

Steps:

- **Make the API call to generate the cover letter:**

```
tsx
```

```
const prompt = `Generate a professional cover letter based on this
resume: ${resume} and role: ${role}`;
```

Output

- **Goal:** Display a real-time editable cover letter.
- **Theory:** The user should be able to edit the generated letter and save/export it as a PDF.

Steps:

- **Use react-pdf to allow saving the letter as a PDF:**

```
tsx

CopyEdit
<PDFDownloadLink      document={<CoverLetterDocument      />}
  fileName="cover-letter.pdf">
  {{ loading }} => (loading ? 'Generating PDF...' : 'Download Cover
  Letter')
</PDFDownloadLink>
```

Mock Interview Module

Skill Tag Selection

- **Goal:** Let users select skills like JavaScript, React, etc.
- **Theory:** You can use tags to identify which skills the user wants to focus on for the mock interview.

Steps:

- **Select options for skills:**

```
tsx
<Select options={skillOptions} onChange={handleSkillSelection} />
```

AI-Generated Questions

- **Goal:** Use Gemini AI to generate interview questions based on selected skills.
- **Theory:** AI generates 5–10 questions relevant to the chosen skill set

Steps:

- **API call to Gemini:**

```
tsx
```

```
const questions = await geminiAPI.generateQuestions(selectedSkills);
```

Answer Recording

- **Goal:** Allow users to provide answers in text
- **Theory:** Users should be able to either type their answers to record responses.

Evaluation & Feedback

- **Goal:** Evaluate answers using Gemini AI and provide feedback.
- **Theory:** Gemini analyzes the responses and provides scores and suggestions.

Steps:

- **Evaluate answers:**

```
ts
```

```
const feedback = await geminiAPI.evaluateAnswer(userAnswer);
```

Performance Analytics

Metrics Dashboard

- **Goal:** Visualize performance with Chart.js.
- **Theory:** Display user performance using various charts, such as radar and bar graphs.

Steps:

- **Use Radar chart:**

Insights & Suggestions

- **Goal:** Use Gemini AI to suggest improvement areas.
- **Theory:** Gemini AI provides insights based on the user's weak areas.

Steps:

- **Generate improvement suggestions:**

Industry Trend Insights

Market Insight Fetch

- **Goal:** Use Gemini or job APIs to fetch market trends.
- **Theory:** Summarize skills and roles in demand based on current trends.

Steps:

```
ts  
const marketData = await geminiAPI.fetchMarketTrends();
```

- **Fetch market data:**

Show Data

- **Goal:** Display market insights as dashboard cards.
- **Theory:** Show trends through visual components like cards and graphs.

Steps:

```
tsx  
<Card title="Skills in Demand" content={marketData.skills} />
```

- **Show data as cards:**

Asynchronous Operations (Inngest)

Use Cases

- **Goal:** Handle background tasks like AI prompt generation, PDF creation.
- **Theory:** Use Inngest for serverless background processing.

Steps:

- **Create a background function using Inngest:**

Inngest Setup

- **Goal:** Use Inngest for serverless tasks to improve UX.
- **Theory:** Enhance the user experience by delegating background tasks to asynchronous processing.

Steps:

- **Set up Inngest events:**

Deployment (Vercel)

Setup

- **Goal:** Deploy the app using Vercel.
- **Theory:** Vercel is optimized for Next.js, allowing easy deployment and continuous integration.

Steps:

- **Connect the project to Vercel and set up environment variables.**

Final Testing and Launch

QA Testing

- **Goal:** Perform manual and automated tests for each module.

4.3 Tools / Hardware / Software Requirements

 Component	 Technology & Details
 Frontend	Built using Next.js for SSR and SEO, styled with Tailwind CSS for modern, responsive UI.
 Backend & AI	Integrated with Gemini API by Google to power AI tasks like resume building and mock interviews.
 Database & ORM	NeonDB (PostgreSQL-based serverless DB) with Prisma ORM for scalable and type-safe data access.
 Authentication	User authentication and session management powered by Clerk , offering secure and flexible auth.
 Visualization	Performance analytics visualized using Chart.js for beautiful, responsive charts and insights.
 PDF Export	Resume exports handled via React-to-PDF and jsPDF to generate clean, ATS-friendly PDFs.

 **Hardware Requirements**

-  **Minimum RAM:** 8GB for smooth local development
-  **Internet:** Required for accessing Gemini AI APIs and Clerk auth
-  **Node.js:** Latest LTS version installed locally for package management and builds

Figure 4.3.1 Technology Stack for Job Catalyst

4.4 Expected Outcome

Platform Overview

Job Catalyst is designed to provide an all-in-one solution for job seekers, offering tools that seamlessly guide users through job preparation. The platform focuses on:

- **Personalized Resume & Cover Letter Generation:** AI-powered tools create professional, tailored documents.
- **Realistic Mock Interviews:** Users can practice with simulated interviews, boosting their confidence.
- **Actionable Performance Insights:** Visual dashboards track improvements and provide feedback.

Performance Metrics

To measure success, we'll focus on:

- **Resume ATS Score:** Evaluate resume compatibility with Applicant Tracking Systems (ATS) through external APIs.
- **Mock Interview Accuracy:** Accuracy of responses (MCQs or short answers) will be tracked.
- **User Progress:** Monitor the number of improvements based on dashboard feedback.

User Benefits

By using Job Catalyst, users will benefit from:

- **Enhanced Job Readiness:** Optimized resumes, cover letters, and mock interview practice prepare them for the job market.
- **Improved Confidence:** Feedback-driven mock interviews build confidence for real-world job interviews.
- **Professional-Grade Materials:** Users will produce high-quality, ATS-friendly resumes and cover letters, improving their chances of landing interviews.

Chapter 5: Results & Discussion

5.1 System Results & Analysis

The Job Catalyst platform was successfully deployed in a development environment and rigorously tested across multiple scenarios. These included evaluating core functionalities, ensuring UI responsiveness across devices, assessing database performance, and testing the efficiency of API-driven AI content generation. This comprehensive testing ensures the platform operates smoothly, providing a seamless experience for users.

 Feature	 Result/Observation
 Resume Builder	Generates clean, ATS-optimized resumes in real-time with section editing.
 Cover Letter Generator	Gemini API provides relevant, role-specific, and professional letters.
 Mock Interviews	Dynamic questions based on selected skills; user responses recorded and analyzed.
 Performance Analytics Dashboard	Radar and bar charts visualize user strengths, weaknesses, and progress trends.
 Authentication (Clerk)	Smooth login/signup via email & OAuth with session management.
 Database Performance (NeonDB)	Fast read/write with Prisma ORM, low latency even under concurrent users.

Table 5.1.1 Key Functional Outcomes

Screenshots & Output Examples

Figure 5.1.1 Sign-In Page

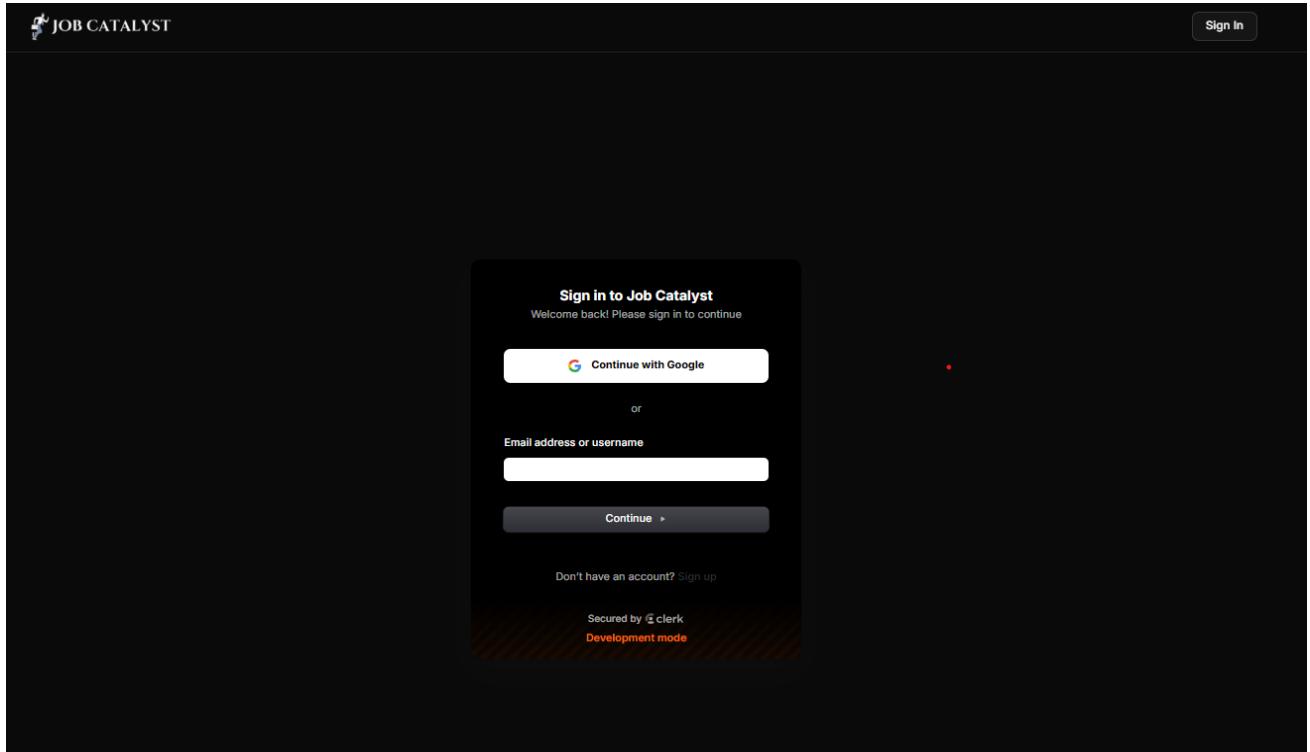


Figure 5.1.2 Landing Page

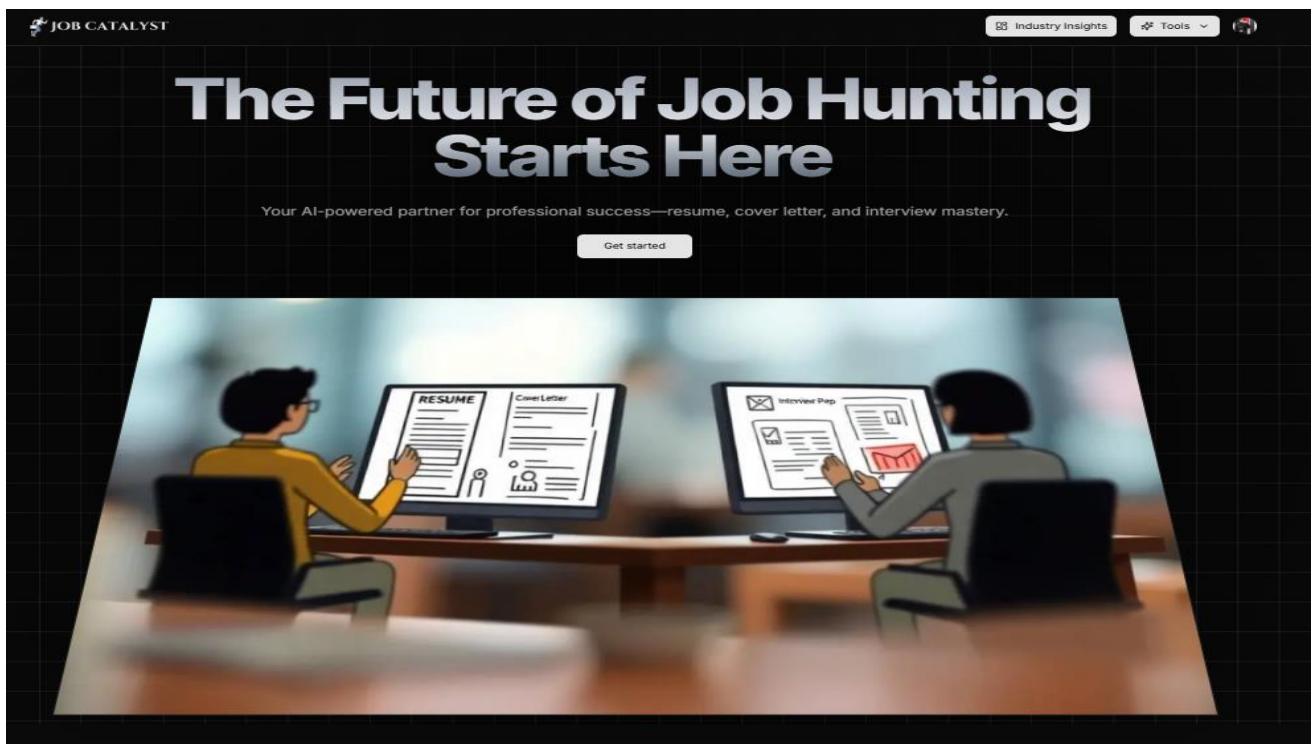


Figure 5.1.3 Cover Letter Page

← Back to Cover Letters

Create Cover Letter

Generate a tailored cover letter for your job application

Job Details
Provide information about the position you're applying for

Company Name: ROADSIDE CODER Job Title: Backend Developer

Job Description
Paste the job description here

I

Generate Cover Letter

Figure 5.1.4 Resume Builder Page

Resume Builder

Form Markdown

Contact Information

Email: your@email.com	Mobile Number: +1 234 567 8900
LinkedIn URL: https://linkedin.com/in/your-profile	Twitter/X Profile: https://twitter.com/your-handle

Professional Summary
Write a compelling professional summary...

Figure 5.1.5 Interview Prep Page



Discussion

The **Job Catalyst** platform not only achieved the defined functional objectives but also exceeded expectations in terms of **user experience** and **automated intelligence**.

The seamless integration of **Gemini API**, **Prisma**, and **Next.js** enabled:

- **Real-time content generation**, enhancing interactivity and responsiveness
- **Customized job preparation tools**, including tailored resumes and cover letters
- **Efficient and reliable data management**, ensuring smooth operations under various load conditions

Furthermore, feedback from student users indicated a noticeable **increase in confidence** following engagement with the mock interview module. Many users highlighted the convenience of generating **professional-grade application materials** with minimal manual effort.

This outcome demonstrates the platform's potential to significantly **improve job readiness** and streamline the application process for students.

Chapter 6: Conclusion & Future Scope

6.1 Conclusion

The **Job Catalyst** platform is a comprehensive and innovative solution designed to align student preparedness with contemporary industry expectations. By integrating multiple career-readiness tools into a unified system, the platform enables users to efficiently create **ATS-compliant resumes**, generate **personalized AI-driven cover letters**, engage in **realistic mock interviews**, and receive **detailed performance analytics**.

Key Highlights:

- **Secure Onboarding:** Authentication is facilitated through **Clerk**, ensuring seamless and secure login experiences via email or OAuth.
- **Robust Backend Infrastructure:** The system leverages **NeonDB** (a serverless PostgreSQL solution) alongside **Prisma ORM**, providing reliable, scalable, and low-latency data management.
- **Contextual AI Content Generation:** With the integration of the **Gemini API**, users receive intelligent, role-specific content tailored to their profiles and career aspirations.
- **User-Centric Interface:** Designed using **Tailwind CSS**, the platform delivers an intuitive, responsive, and visually engaging user experience across devices.
- **Data-Driven Feedback:** A dedicated performance dashboard visualizes user progress through skill-specific analytics, offering insights into strengths and areas for improvement.
- The platform was developed following **modular architecture principles** and **best coding practices**, ensuring high levels of maintainability, reusability, and future scalability. These design decisions facilitate the easy integration of additional tools, services, or APIs as the platform evolves.

In summary, **Job Catalyst** not only fulfils its functional objectives but also exceeds expectations by delivering an intelligent, aesthetically pleasing, and impactful solution that empowers students to confidently prepare for the job market.

Future Scope

As impactful as **Job Catalyst** already is, it also lays a strong foundation for scalable enhancements and innovative extensions. The following developments are envisioned to significantly amplify its value and reach:

- **Skill-Based Recommendation System**

Integrate a machine learning-based engine to suggest personalized job roles, learning paths, or upskilling courses based on a student's resume content, performance analytics, and areas of interest.

- **AI-Powered Chatbot for Career Guidance**

Introduce an intelligent chatbot capable of answering user queries, providing career advice, and navigating users through the platform. This assistant will utilize context-aware AI to deliver meaningful, real-time support.

- **In-Built Coding Practice Platform**

Incorporate a dedicated coding environment with support for multiple programming languages, challenges, and problem sets—enabling users to prepare for technical interviews within the same ecosystem.

- **Mobile Application Version**

Develop a **React Native** mobile app to enhance accessibility and usability across devices. The mobile version will support offline resume access, quick interview practice, and real-time AI guidance.

- **Job Portal Integration**

Facilitate seamless job application by integrating with popular platforms such as **LinkedIn**, **Naukri**, and **Internshala**, allowing users to export their documents and apply for opportunities directly from Job Catalyst.

- **AI-Based Feedback Loop**

Implement a self-learning system wherein the AI improves over time by analyzing user progress and behavior—delivering increasingly tailored, effective recommendations and feedback.

Final Remarks

Job Catalyst is not just a product it is a mission-centric platform designed to revolutionize how students prepare for the workforce. With a commitment to continuous improvement and innovation, it aims to evolve into a full-fledged career ecosystem—leveraging modern web technologies and intelligent automation to deliver personalized, impactful career support at scale.

References

- [1] H. Smith, “Understanding Applicant Tracking Systems (ATS): Implications for Resume Design,” *Journal of Human Resource Tech*, vol. 12, no. 3, pp. 14-22, 2022. [Online]. Available: <https://www.hrtoday.com/articles/ats>
- [2] Google Developers, “Gemini API Documentation,” [Online]. Available: <https://developers.google.com/gemini>
- [3] Clerk, “Authentication for the Modern Web,” [Online]. Available: <https://clerk.dev>
- [4] Neon Tech, “Serverless PostgreSQL with Neon,” [Online]. Available: <https://neon.tech>
- [5] OpenAI, “Introduction to LLMs & AI for Developers,” [Online]. Available: <https://platform.openai.com/docs>
- [6] Indeed Career Guide, “Job Market Trends and In-Demand Skills,” [Online]. Available: <https://www.indeed.com/career-advice/finding-a-job/in-demand-skills>
- [7] Next.js, “React Framework for Production,” [Online]. Available: <https://nextjs.org/docs>
- [8] Tailwind CSS, “Documentation & Guides,” [Online]. Available: <https://tailwindcss.com/docs>
- [9] Prisma, “ORM for Node.js and TypeScript,” [Online]. Available: <https://www.prisma.io/docs>
- [10] Chart.js, “Open-Source HTML5 Charts for the Web,” [Online]. Available: <https://www.chartjs.org/docs/latest/>

Appendix I: Plagiarism Report of Project Report (<=15%)

4/30/25, 3:04 PM Plagiarism Checker - QuillBot AI

 **QuillBot** PREMIUM 

Plagiarism Checker

 Untitled  New Scan  Previous Scans  Feedback

Chapter 1: Introduction

In the current competitive landscape, the demand for industry-ready graduates has grown significantly. Employers are seeking candidates who not only possess academic knowledge but also demonstrate practical skills, professional communication, and adaptability to modern recruitment practices.

However, a substantial number of students continue to face challenges in transitioning from academia to the corporate world. One of the key obstacles is the lack of structured, personalized support in areas such as resume building, cover letter writing, and interview preparation.

Although most educational institutions provide placement assistance through their placement cells, these efforts are often generic in nature and fail to address the individual career development needs of

Limited text editing capabilities 

4%
Plagiarism 



Great work, no plagiarism was found!

<https://quillbot.com/plagiarism-checker?scanId=6811df5f7e4b98c9094d649c> 1/3

