Egyptian E-Learning University

Faculty of Computers & Information Technology

Jobify

A Comprehensive Platform for Early-Career Job Seekers

By

Kariem Hatem	(21-00850)
Sayed Ali	(21-01579)
Rahma Essam	(21-01453)
Sohila Salem	(21-01645)
Shahd Abo-Elhamd	(21-01619)
Mayar Essam	(21-01049)
Sara Gamal	(21-01574)

Supervised by

Dr. Safi Ibrahim

Assistant

Eng. Rofida Abdel Hafez

[Assuit]-2025

Abstract

Fresh graduates and entry-level job seekers often face challenges in finding the right job opportunities. Traditional job platforms do not always offer personalized recommendations. To solve this, Jobify was created as an AI-powered platform to help users find jobs that match their skills and career goals.

The main feature of Jobify is its job matching system, which compares user resumes with job listings to provide personalized job recommendations. It also includes an AI-powered resume checker that gives feedback on how well a resume fits a specific job description.

In addition, Jobify offers a mock interview preparation module, helping users practice for interviews with suggested questions and answers. The platform also provides job analytics tools to track market trends and application patterns, along with a chatbot assistant that guides users through the platform.

Built using Angular for the frontend, .NET and Node.js for the backend, and MongoDB for the database, Jobify is designed to be flexible and scalable. The platform aims to simplify the job search and enhance career readiness for fresh graduates and early-career professionals.

Jobify has been tested and received positive feedback for its ability to improve job search efficiency and interview preparation, helping users increase their chances of success in the job market.

Tabel of Contents

List of Figures

Chapter One
Introduction

Introduction

In today's competitive job market, finding the right job is a significant challenge, particularly for fresh graduates and early-career professionals. Despite the availability of numerous job search platforms, many job seekers still struggle to find the right opportunities that align with their skills, qualifications, and career aspirations.

In response to these challenges, Jobify was created as a comprehensive platform designed to revolutionize the way individuals search for jobs and prepare for their careers. By integrating AI-driven features and focusing on the needs of fresh graduates and young professionals, Jobify offers a smarter and more efficient job search process. The platform provides not only job recommendations but also personalized resume matching, mock interview preparation, and valuable job market insights.

This chapter provides an overview of Jobify, including the background of the project, the problem it aims to address, its objectives, the scope of the project, and the target audience. Each section will explore how the platform strives to improve the job search experience and provide a more tailored and supportive approach to career development.

1.1 Background

Finding the right job has become more challenging, especially for fresh graduates and early-career professionals. Traditional job search platforms often provide too many irrelevant job listings and lack personalized features that help job seekers find the best opportunities. These platforms mainly use basic keyword matching, which doesn't always align with a candidate's skills, preferences, or career goals.

Moreover, these platforms usually don't offer tools like resume matching, interview preparation, or insights into the job market, making it harder for candidates to succeed. This

creates a gap in the market for a more advanced, intelligent solution that can help job seekers navigate this complex process.

Jobify aims to fill this gap by using artificial intelligence (AI) to provide personalized job recommendations, AI-powered resume matching, and mock interview simulations. By integrating these features, Jobify aims to streamline the job search, making it more efficient and tailored to each user's needs. Additionally, the platform will provide valuable career insights to help users understand trends, salary expectations, and industry demands, giving them the tools they need to succeed in their job search.

1.2 Importance of the problem

The challenges faced by fresh graduates and early-career professionals in the job market are both significant and multifaceted, impacting their ability to secure meaningful employment and launch successful careers. The inefficiencies of traditional job search platforms, which often overwhelm users with irrelevant listings due to simplistic keyword-based algorithms, exacerbate the difficulty of finding opportunities that align with individual skills and aspirations. Furthermore, the lack of personalized tools for resume optimization, interview preparation, and market insights leaves job seekers ill-equipped to compete in a highly competitive environment.

This gap not only hinders individual career progression but also contributes to broader economic issues, such as underemployment and skill mismatches in the workforce. Jobify addresses these critical pain points by providing an AI-powered, user-centric platform that delivers tailored job recommendations, actionable resume feedback, and comprehensive career preparation tools, thereby empowering users to navigate the job market with confidence and efficiency. By tackling these issues, Jobify not only enhances individual employability but also supports employers in accessing a pool of well-prepared, motivated candidates, fostering a more effective and equitable job market ecosystem.

1.3 Problem Statement

The problem that Jobify addresses is multifaceted, focusing on several key issues faced by fresh graduates and early-career professionals in the job market:

1. Inefficient Job Search: Traditional job search platforms are flooded with irrelevant job postings due to poor matching algorithms. Fresh graduates often receive hundreds of job listings that do not align with their qualifications or career goals.

2. Lack of Personalized Job Matching: Existing job platforms typically offer generic search results and do not tailor job recommendations to individual preferences, skills, or qualifications, making it harder for candidates to find the most relevant opportunities.

3. Interview Preparation Deficiency: Many candidates struggle with interview preparation, often lacking resources to simulate realistic interview environments or receive actionable feedback on their performance.

4. Inadequate Career Insights: Current platforms do not provide job seekers with adequate insights into trends such as skill demand, industry salary ranges, and overall market conditions, which are essential for making informed decisions.

Jobify solves these problems by offering a unique, AI-powered platform that not only connects job seekers to suitable job opportunities but also provides personalized resume matching, tailored job recommendations, mock interview experiences, and valuable job market analytics. By integrating these features, Jobify strives to enhance the job search process and improve outcomes for candidates.

1.4 Objectives

The primary objectives of Jobify are to create an intelligent and user-friendly platform that transforms the job search experience, especially for fresh graduates and early-career professionals. The key objectives include:

1. To Revolutionize Job Matching

   Build an AI-driven system that matches job seekers with relevant opportunities by analyzing their resumes, skills, academic background, and personal preferences.

2. To Enhance Interview Preparation

   Provide interactive mock interview sessions with instant feedback, tips, and evaluation metrics to help users improve communication, confidence, and readiness.

3. To Offer Job Market Analytics

   Deliver real-time job market insights including in-demand skills, industry trends, and salary benchmarks to support informed career planning.

4. To Enable Internship and Part-time Opportunities

   Include curated listings of internships and part-time roles suitable for students and fresh graduates to help them gain early professional experience.

5. To Improve Job Seeker Experience

   Ensure a seamless user experience through intuitive design, smart search filters, and personalized dashboards for better engagement and satisfaction.

6. To Provide Resume Analysis and Feedback

   Incorporate AI tools to assess user CVs and offer suggestions for optimization, improving their chances of passing recruiter screening systems.

7. To Facilitate Continuous Career Growth

   Recommend relevant certifications, courses, or upskilling opportunities based on user profiles and current job market needs.

8. To Build an Interactive Chatbot Assistant

   Integrate a smart chatbot to guide users through job searches, answer questions, and assist with common tasks like resume review or job filtering.

9. To Bridge the Gap Between Employers and Fresh Talent

   Create a platform that allows employers to easily discover and connect with fresh, motivated candidates through AI-backed talent matching.

1.5 Scope of the Project

Jobify aims to provide an all-encompassing, integrated solution for job seekers, offering them both the tools and the support necessary to succeed in their career search. The scope of this project includes the design, development, and deployment of a web-based platform that supports job seekers throughout their entire employment journey. The platform will focus primarily on fresh graduates, early-career professionals, and students seeking internships or part-time opportunities. The scope covers the following key areas:

1.5.1 Core Functionalities

Jobify offers a comprehensive set of core features designed to support job seekers throughout their career journey, from searching for opportunities to preparing for interviews. The platform's core functionalities include:

1. AI-Driven Resume Matching

   An intelligent algorithm analyzes resumes to extract key skills, education, experience, and career goals. It then matches users with the most relevant job openings, improving the accuracy and efficiency of job recommendations.
2. Personalized Job Search Engine

   Users can filter job listings by location, industry, job type (remote, full-time, part-time, internship), required skills, company size, and salary range. The system continuously learns user preferences to refine results over time.
3. Mock Interview Preparation

   An interactive module provides users with mock interview scenarios based on their field of interest. Feedback is generated using AI to evaluate communication skills, clarity, confidence, and relevance of answers.
4. Job Market Insights and Analytics

   Users gain access to real-time job market data, including:
   - In-demand roles and industries
   - Emerging technologies and required skills

- Regional salary benchmarks
- Job market saturation and growth patterns

5. Internship and Part-Time Listings

   The platform includes a dedicated section for internships and part-time roles, helping students and early-career professionals gain practical experience aligned with their long-term goals.

6. Resume Builder and Optimization Tool

   A built-in tool allows users to build or update their CVs using guided templates. AI suggestions help users improve formatting, phrasing, and keyword inclusion to pass applicant tracking systems (ATS).

7. Smart Chatbot Assistant

   A conversational AI assistant is available 24/7 to help users navigate the platform, provide instant support, answer career-related questions, and assist in job applications.

8. User Dashboard and Application Tracking

   A personal dashboard allows users to:

   - Track submitted applications
   - Monitor application status
   - Save favorite jobs
   - Set job alerts and notifications

9. Skill Development Suggestions

   Based on job trends and user profiles, the system recommends relevant certifications, courses, and learning paths to improve employability and remain competitive in the market.

10. Employer Profiles and Reviews

    Jobify allows users to view company profiles, including reviews from previous employees, culture ratings, benefits, and interview experiences to make more informed application decisions.

11. Multilingual Support

    The platform is designed to be accessible in multiple languages to support users from diverse linguistic backgrounds and improve usability across regions.

12. Mobile-Friendly Design

    The application is responsive and optimized for mobile devices, ensuring users can search and apply for jobs anytime, anywhere.

1.5.2 Technology Scope:

The development of Jobify relies on a robust and scalable technology stack that supports seamless performance, secure data handling, and modern user experiences. The technology scope includes:

1. Frontend Development
    - The user interface is developed using Angular for a dynamic, component-driven web application structure.
    - Bootstrap is used for consistent styling and responsive design across various screen sizes and devices.
2. Backend Development
    - The backend is implemented using .NET, offering powerful tools for building scalable and secure APIs and server-side logic.
    - The use of .NET ensures efficient performance, strong type safety, and enterprise-level security.
3. Database Management
    - SQL Server is utilized as the relational database management system.
    - It provides structured data storage, complex query support, and transactional reliability, ensuring data integrity for user profiles, job listings, applications, and system logs.
4. Artificial Intelligence & Machine Learning
    - AI models are integrated to:

        - Perform intelligent resume-to-job matching.
        - Generate personalized job recommendations.
        - Simulate mock interviews and evaluate user responses.

5. Authentication & Security
    - Secure user authentication is implemented using JWT (JSON Web Tokens) or OAuth, ensuring safe and flexible login mechanisms.
    - Data encryption and access control mechanisms protect user information and maintain privacy standards.
6. API Architecture
    - RESTful APIs connect the frontend and backend, ensuring modular, scalable communication between components.
    - Future expansion includes integrating third-party APIs for services like LinkedIn data import or learning platform suggestions.
7. Cloud Hosting and Scalability
    - The platform will be deployed on cloud infrastructure such as Azure or AWS, supporting auto-scaling, load balancing, and disaster recovery.
8. Development and Deployment Tools
    - Git is used for version control, with repositories hosted on GitHub or GitLab.
    - CI/CD pipelines are established for automated testing, integration, and deployment, ensuring smooth updates and rapid iteration.

1.5.3 Limitations:

The platform will initially focus on entry-level job seekers, fresh graduates, and early-career professionals, excluding senior-level or highly specialized job roles.

AI-driven matching will evolve over time as the platform gathers more data and user interactions to refine the algorithms.

1.5.4 Future Scope:

As Jobify evolves, several enhancements are planned to increase its impact and usability:

1. Employer Dashboard Enhancements
    - Advanced features for recruiters such as automated applicant filtering, interview scheduling tools, and candidate performance tracking.
2. Expanded Job Platform Integration
    - Connection with external job boards and hiring platforms (e.g., LinkedIn, Indeed) to broaden the range of job listings available to users.
3. Mobile Application Development
    - Native mobile apps for iOS and Android to provide users with on-the-go access and improved engagement.
4. Skill-Based Learning Recommendations
    - Integration with online learning platforms (e.g., Coursera, Udemy) to suggest personalized courses that align with career goals and industry demands.
5. Multilingual and Regional Support
    - Localization of content and support for multiple languages to serve diverse user bases across different regions.
6. Enhanced AI Capabilities
    - More intelligent and context-aware AI algorithms for deeper personalization, career path suggestions, and real-time resume scoring.

1.6 Target Audience

The primary users of Jobify are individuals actively seeking job opportunities and career development tools. The platform is designed to meet the needs of the following groups:

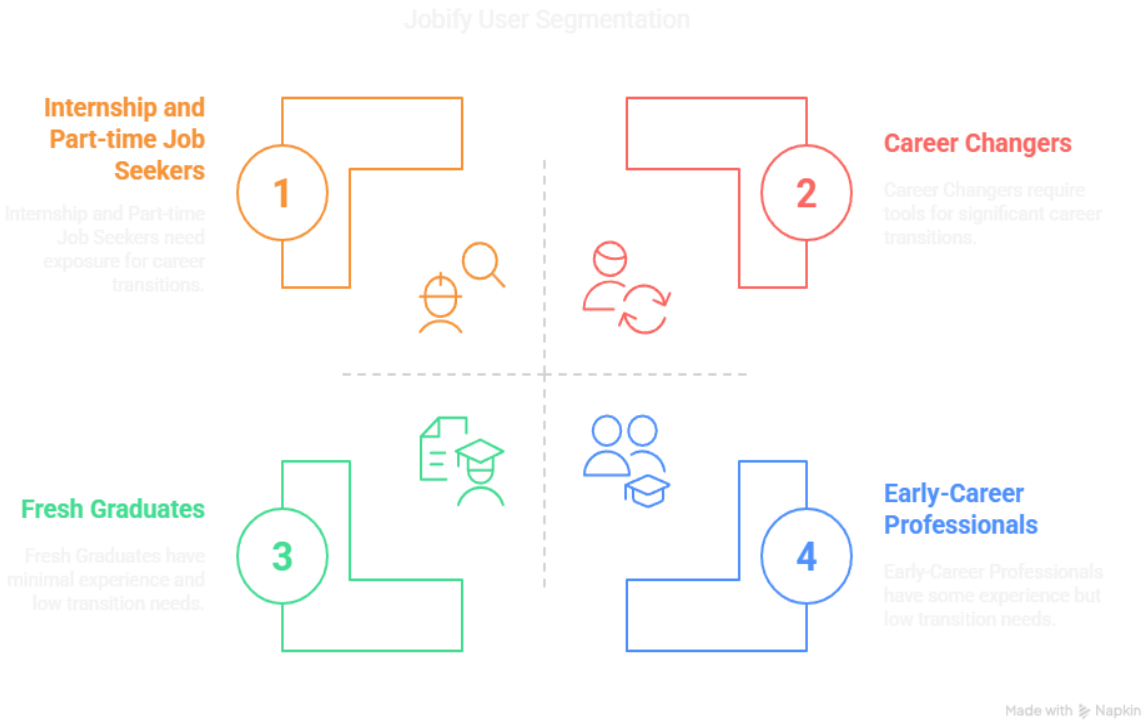


Figure 1.6 Target Audience

1. Fresh Graduates
    - Students and recent university graduates looking to enter the job market with minimal experience but strong motivation to begin their careers.
2. Early-Career Professionals
    - Individuals with up to three years of professional experience who are looking to explore new roles, gain experience, or transition into better job opportunities.
3. Internship and Part-time Job Seekers
    - Users interested in gaining industry exposure while studying or transitioning into a new career path through internships and part-time roles.

4. Career Changers
    - Those shifting from one field to another and in need of tools that assess transferable skills, recommend relevant job roles, and provide preparation resources.
5. Educational Institutions and Career Centers *(Future Scope)*
    - Universities and training centers that may use the platform to support students with job matching, interview preparation, and labor market insights.

1.7 Brief Overview of the Proposed Solution

The proposed solution, Jobify, leverages advanced artificial intelligence to revolutionize the job search experience for early-career professionals and fresh graduates. By integrating a sophisticated algorithm that analyzes user profiles, skills, and preferences, Jobify delivers personalized job recommendations that surpass the limitations of traditional keyword-based platforms. The system includes a CV analyzer to provide actionable feedback for resume optimization, a mock interview tool to enhance interview readiness, and a chatbot assistant to offer real-time career guidance. Additionally, Jobify provides job analytics and insights to help users understand market trends, alongside dedicated listings for internships and part-time opportunities. Built on a robust technology stack featuring Angular for the frontend, ASP.NET Core for the backend, and SQL for database management, Jobify ensures a seamless, scalable, and user-centric experience, empowering job seekers to navigate the competitive employment landscape with confidence and precision.

Chapter Two
System Overview

System Overview

The System Overview chapter provides a comprehensive understanding of Jobify, highlighting its core functionalities, the problem it solves, and how it empowers job seekers in their career journey. This chapter serves as an introductory overview of the platform, giving readers insight into the underlying goals, features, and the technological approach taken to create a powerful, user-centric tool for job hunting.

Jobify is designed to bridge the gap between job seekers and employers by offering intelligent features powered by AI, data analytics, and a user-friendly interface. It goes beyond traditional job boards, providing a holistic approach to career management. From matching resumes to jobs, preparing candidates for interviews, and offering industry insights, Jobify aims to improve the efficiency of the job search process while enhancing the overall user experience.

By addressing the specific needs of fresh graduates, entry-level professionals, and career changers, Jobify offers not just job listings, but also the tools necessary to succeed in the modern job market. This chapter explains the foundational elements of the platform and sets the stage for a deeper dive into its key features, architecture, and technology.

2.1 What is Jobify?

Jobify is an AI-powered career platform designed for fresh graduates**,** entry-level professionals, and those looking to change careers. It offers personalized features that guide users through their job search, helping them find the right opportunities, enhance their interview skills, and optimize their resumes for better results.

Unlike traditional job boards, Jobify provides a more comprehensive, integrated solution that evolves with both the user’s career goals and the dynamic job market.

This Digrams Show Visualize About it :

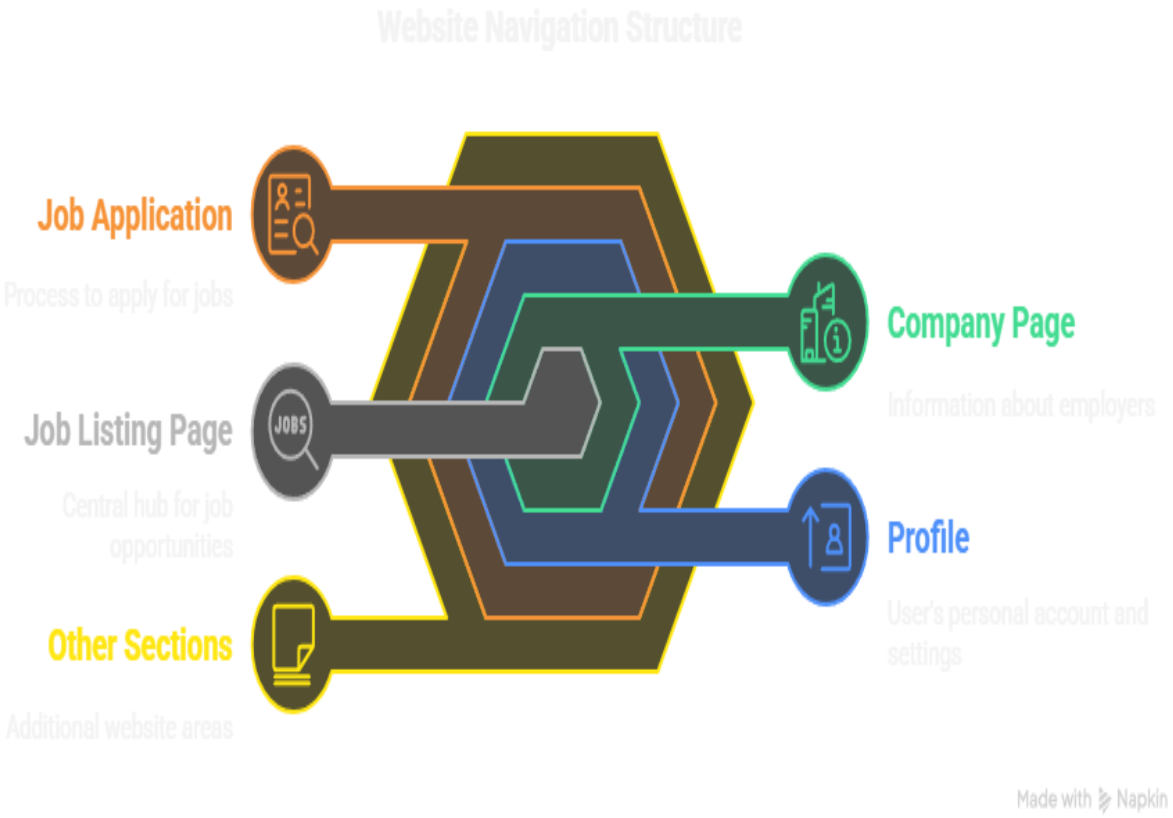


Figure 2.1.1 Website Navigation Structure

Core Features of Jobify:

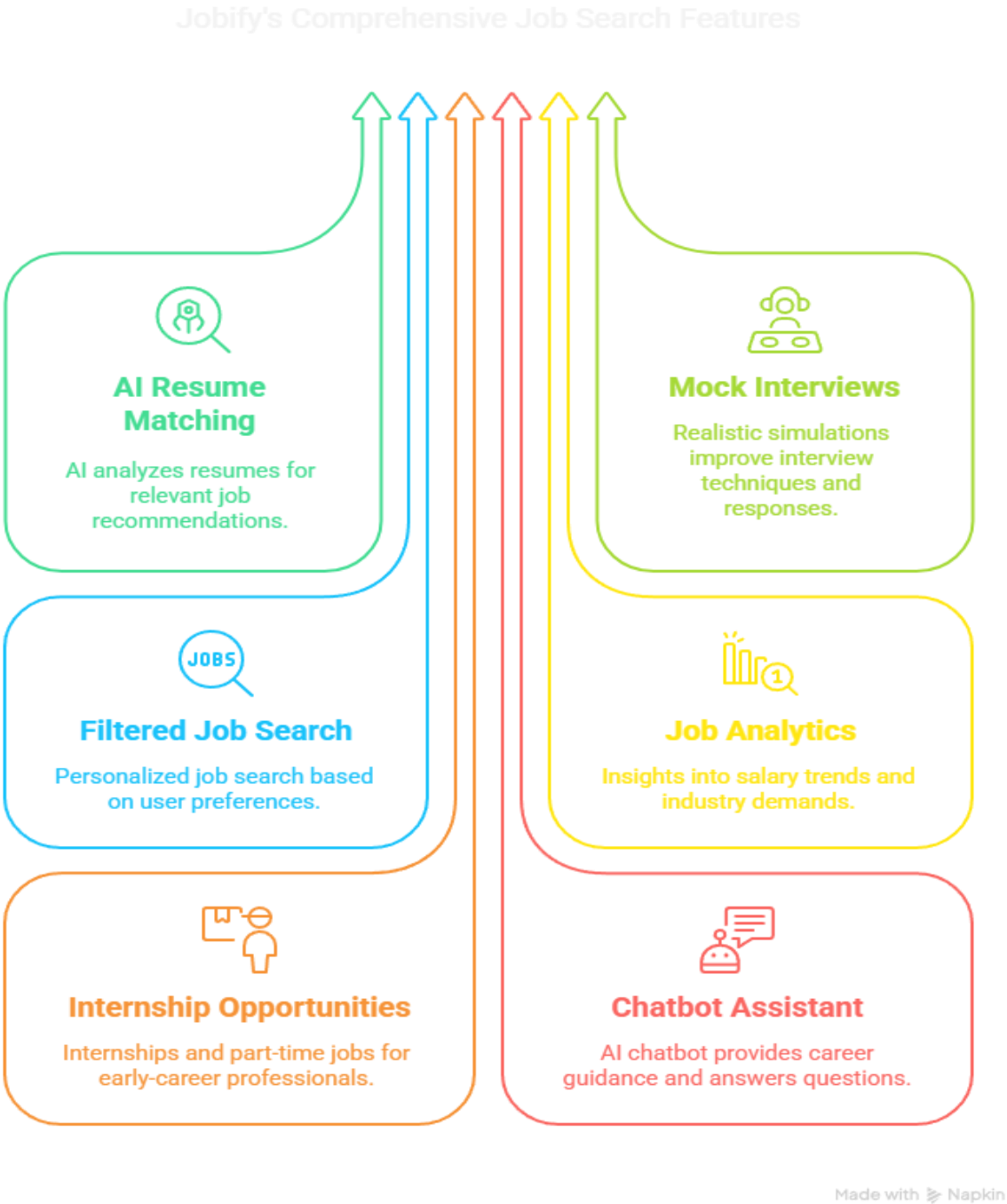


Figure 2.1.2 Jobify Core Fetures

1. AI Resume Matching:

   The AI-driven system analyzes the user's resume and recommends the most relevant job openings based on the user's qualifications, skills, experience, and career preferences.

2. Mock Interviews:

   Jobify provides realistic mock interview simulations, helping users improve their responses and refine their interview techniques with personalized feedback.

3. Filtered Job Search:

   Users can search for job listings based on location, salary expectations, job type, required skills, and experience level, ensuring a highly personalized job search experience.

4. Job Analytics:

   Gain insights into salary trends, industry demands, and job availability. Users can understand which skills are in demand and adjust their career strategy accordingly.

5. Internships & Part-Time Jobs:

   For students and early-career professionals, Jobify lists internship and part-time job opportunities to help users build real-world experience.

6. Chatbot Assistant:

   Jobify's AI-powered chatbot assists users by answering questions and providing career guidance at any point in the job search process.

7. Services Page:

   The Services Page introduces various tools and resources available to users, such as:

   - Resume Builder**:** Step-by-step guidance on creating a professional resume.
   - Career Resources**:** Articles, tips, and videos on how to excel in interviews, grow professionally, and navigate the job market.
   - Skill Development: Access to online courses and certifications to enhance job prospects.

8. Job Listing Page (Including Internships):

   The Job Listing Page displays all available job positions, including:

   - Full-time jobs
   - Part-time jobs
   - Internships The page is customizable, allowing users to filter by job type, industry, location, salary, and more. Users can apply directly for positions from this page.

9. Company Page:

   The Company Page provides detailed profiles of companies hiring on Jobify. Each company profile includes:

   - Company background and culture
   - Open job listings
   - Employee reviews and ratings
   - Information on benefits and work-life balance This helps users gain a better understanding of potential employers before applying.

10. Contact Page:

    The Contact Page offers users a direct way to get in touch with Jobify support or inquire about job opportunities. It includes:

    - Email Support
    - Live Chat
    - Phone Number for more personalized inquiries
    - FAQs section for common user queries

11. Subscribe for Latest Jobs:

    Users can subscribe to job alerts and notifications tailored to their job preferences. Jobify will send updates on new job openings that match their selected criteria, ensuring they never miss an opportunity.

12. Notifications:

    Jobify provides real-time notifications for:

    - New job listings matching the user's profile
    - Interview invitations
    - Application status updates
    - Resume feedback updates
    - Personalized career tips

13. Login & Sign-Up:

    Users can securely create an account through the Sign-Up feature or log in with an existing account. The platform supports:

    - Social logins (Google, Facebook, etc.)
    - Email-based logins
    - Password recovery for easy access

14. Apply Now:

    With the Apply Now feature, users can easily submit their resume and apply for job positions directly through Jobify, with just a few clicks. The platform supports:

    - One-click applications for fast job submissions
    - Automated resume submission tailored to the specific job requirements

15. Forgot Password:

    Jobify provides a Forgot Password option for users to quickly reset their password and regain access to their account without any hassle.

16. Profile Page:

    The Profile Page is the user's personalized dashboard where they can:

    - Upload and update their resume.
    - Track job applications and interview statuses.
    - View job recommendations.
    - Manage notifications and subscriptions.
    - Update personal details and preferences.

2.2 Key Use Cases

Jobify's features are tailored to meet the diverse needs of job seekers at various stages of their career journey. The platform offers a wide range of use cases that make it a versatile, efficient, and practical tool for users. Below are the key use cases that highlight the core value and capabilities of Jobify:

1. Resume-to-Job AI Matching

Description:

Users can upload their resumes or create them directly on Jobify. The AI-powered engine scans the resume for relevant qualifications, skills, experiences, and preferences. Based on this analysis, Jobify suggests the most compatible job opportunities. This AI resume matching ensures job seekers are presented with highly relevant job listings that align with their profile.

Key Features:

- AI-driven analysis of resumes.
- Personalized job recommendations.
- Matching based on skills, qualifications, and career preferences.

2. Filtered Search Experience

Description:

Job seekers can customize their job search with various filters to find the most relevant listings. The platform offers a range of filtering options to make the job search process efficient and tailored to specific needs.

Key Features:

- Location: Remote, hybrid, or on-site options.
- Industry or Job Category: Users can specify the sector they are interested in (e.g., IT, finance, marketing).
- Skills Required: Search jobs based on specific skill sets (e.g., Java, Python, marketing)
- Employment Type: Full-time, part-time, internship.
- Salary Expectations: Filter jobs based on salary ranges or compensation preferences.

3. Interview Preparation Mode

Description:

Jobify offers a comprehensive mock interview feature where users can practice answering commonly asked interview questions in a simulated environment. The platform provides valuable feedback to help users refine their interview techniques.

Key Features:

- Realistic mock interviews based on industry-specific questions.
- Instant feedback on user performance, including strengths and weaknesses.
- Tips for improvement after each session.
- Recorded responses for users to review and improve over time.

4. Job Market Insights Dashboard

Description:

The Job Market Insights Dashboard equips users with valuable industry-specific data to help them make more informed career decisions. This feature helps users understand the trends affecting job markets and salaries in their chosen field.

Key Fetures:

- Trending skills that are in high demand in specific industries or regions.
- Competitive salary data to ensure job seekers know what to expect in terms of compensation.
- Job availability rates that highlight which sectors are hiring and in demand.
- Hiring trends across different industries, helping users adapt to changes in the job market.

5. Internship & Part-Time Focus

Description:

Jobify understands the importance of gaining experience early in one's career. The platform tailors job listings to help students, recent graduates, and early-career professionals find internships or part-time job opportunities that align with their academic background and career goals.

Key Features:

- Internships: Academic-aligned internships that give students hands-on experience.
- Short-Term Projects: Opportunities for gig or freelance work that helps build relevant skills.
- Volunteer Roles: Volunteer positions that allow users to gain experience while contributing to a cause.

6. AI Chatbot Support

Description:

Jobify's AI-powered chatbot acts as a personal assistant, guiding users through the platform and answering any questions they may have. The chatbot helps users navigate the platform efficiently, access resources, and get answers in real-time.

Key Features:

- Platform navigation: Helps users explore the different features of Jobify.
- Explaining features: Provides clarifications on how to use specific functionalities.
- Job suggestions: Recommends jobs based on user preferences.
- Real-time answers: Answers FAQs and provides assistance as needed.

7. User Progress Tracker (Future Addition)

Description:

In a future update, Jobify plans to introduce a User Progress Tracker, which will provide users with a comprehensive view of their job search journey. This dashboard will help users track their progress and stay motivated.

Key Features:

- Applications submitted: Track the status of each application (e.g., pending, interviewed, rejected).
- Interview performance: View historical feedback from mock interviews to monitor improvement.
- Skills gained: Display the progress in skill-building activities like certifications, courses, and on-the-job experience.
- Resume updates: Track changes made to the user’s resume and ensure it aligns with job requirements.

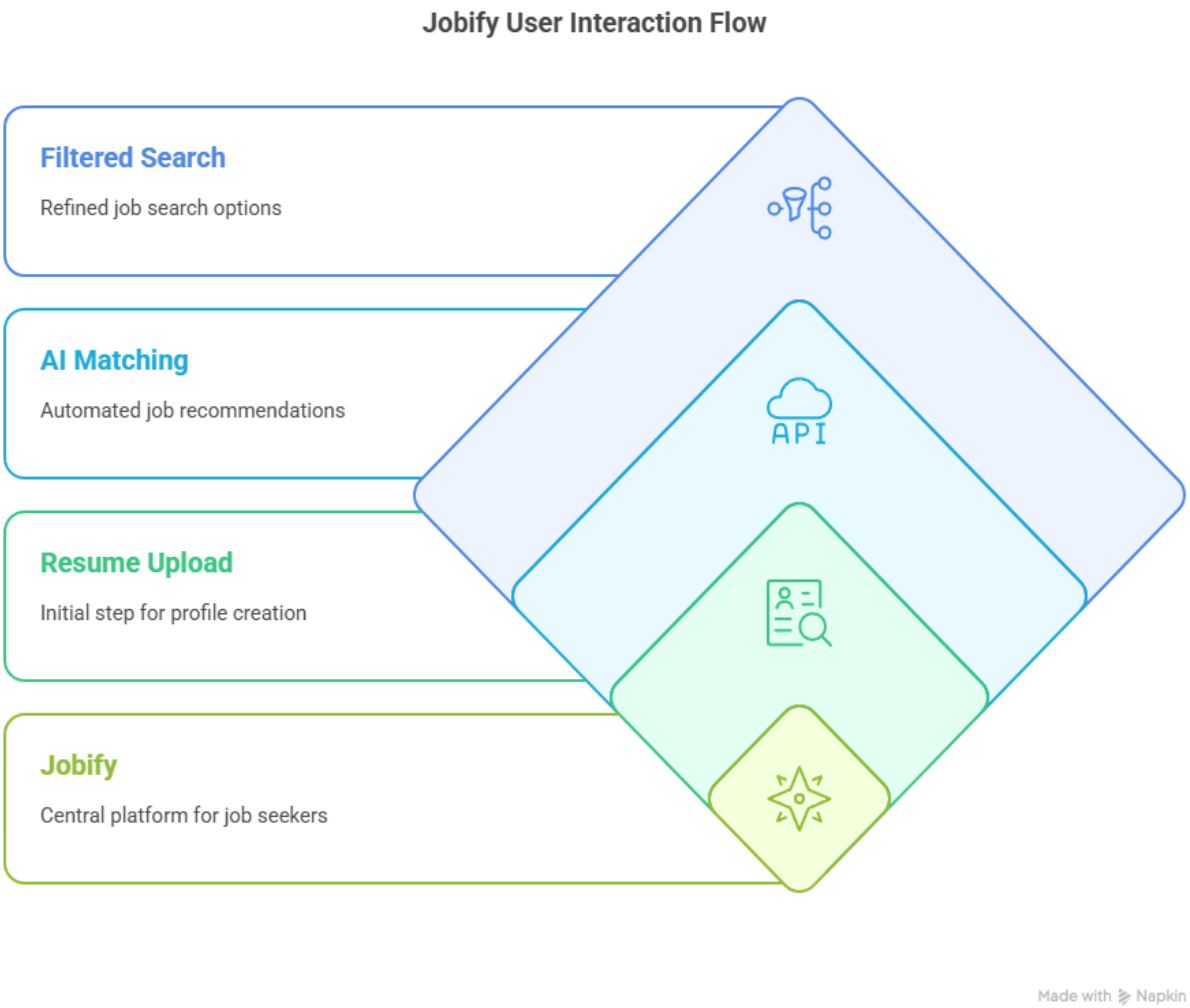


Figure 2.2.1 User flow relation

Chapter Three
Proposed System

Proposed System

The System Architecture chapter delves into the foundational structure that enables Jobify to operate efficiently at scale. It covers the key design principles, system components, and technologies that make up the Jobify platform. The architecture is designed with modularity, scalability, and performance in mind, ensuring that as the platform grows, its components can scale to meet the needs of an expanding user base.

Jobify's architecture incorporates a multi-layered approach that integrates a powerful frontend, a robust backend, a flexible database system, and advanced AI algorithms. Each of these components plays a crucial role in delivering a seamless, intelligent, and user-centric experience. This chapter not only explores the technology choices but also provides insight into how these layers work together to provide an intuitive and effective platform for job seekers.

3.1 Approach used to solve the problem

The system design for Jobify is composed of several core layers, each with specific roles and responsibilities. Below is an expanded breakdown of the system design:

Frontend Layer (Client-Side):

- The Angular framework serves as the backbone for Jobify's frontend, offering a modular, single-page application (SPA) experience. Angular's two-way data binding, component-based architecture, and robust tooling make it ideal for creating a dynamic, responsive user interface.
- Bootstrap is used for styling, ensuring that the platform is mobile-responsive and can easily adapt to different screen sizes, from desktop monitors to smartphones.

The frontend interacts directly with the backend via APIs, ensuring smooth, real-time updates and interactions between users and the platform.

Backend Layer (Server-Side):

- The .NET backend is responsible for managing business logic, user authentication, and interfacing with the database.
- The backend handles requests from the frontend, processes data, and ensures that all functionalities, such as resume matching and interview simulations, are performed accurately.
- For .NET applications, the backend leverages ASP.NET Core for building robust APIs that are secure, scalable, and easily maintainable.

The backend layer is responsible for performing security functions like encryption and ensuring the integrity of sensitive data such as passwords and user profiles.

Database Layer:

- SQL Server is utilized as the relational database management system (RDBMS) for managing structured data such as user profiles, job listings, job applications, and mock interview results.
- The use of SQL Server ensures that data is stored efficiently and can be queried with high performance. It also provides support for complex joins, transactional integrity, and robust backup mechanisms.
- The database schema is designed to handle a wide range of data types and to scale as the number of users and job listings increases. Key entities include Users, Job Listings, Applications, Skills, and Resumes.

AI & Machine Learning Layer:

The AI & Machine Learning Layer in Jobify focuses on two main functionalities: resume matching and AI-powered chatbot assistance. This layer leverages machine learning and natural language processing (NLP) technologies to continuously enhance the platform's ability to provide relevant and personalized recommendations. Here's how these features work:

1. Resume Matching

- Jobify uses AI-driven resume matching to help users find the most suitable job opportunities based on their profiles.
- The system works by analyzing the content of resumes, including the skills, qualifications, experience, and other key details provided by the user.
- The AI then compares this information to the job listings available on the platform to find the best matches.

Key elements involved in resume matching:

Natural Language Processing (NLP): NLP techniques are used to extract key information from resumes, such as skills, job titles, and educational qualifications. This data is then processed to create a profile that is compared to job descriptions.

Similarity Matching: The AI evaluates the similarity between the user's resume and the job descriptions. It looks for overlap in required skills, experience, and other job-specific criteria to recommend the most relevant positions.

Continuous Learning: As more users interact with the platform and provide feedback on job matches, the system learns and refines its matching algorithm. Over time, the AI becomes better at understanding user preferences, job market trends, and the nuances of job descriptions.

2. AI Chatbot Assistance

Jobify includes an AI-powered chatbot designed to assist users in navigating the platform, answering their questions, and offering personalized advice throughout the job search process.

Key features of the AI Chatbot:

User Interaction: The chatbot engages with users by answering questions about job opportunities, platform features, and career-related queries. It provides quick and accurate responses, helping users find what they need without having to manually search.

Job Recommendations: Based on the user's profile, preferences, and interactions, the chatbot suggests job openings that match their qualifications and career goals. It uses machine learning models to improve the relevance of these recommendations over time.

Guidance and Support: The chatbot can guide users through processes like setting up job alerts, applying for positions, and preparing resumes. It offers valuable resources and tips for interview preparation and networking.

Personalization: The AI chatbot learns from the user's interactions and customizes its responses to meet the user's specific needs. As users continue to engage with the platform, the chatbot becomes more adept at understanding their preferences and career aspirations.

Continuous Improvement:

- Both the resume matching system and the AI chatbot are designed to improve as they interact with more users. Machine learning models are continuously trained and fine-tuned using new data to provide better recommendations, job matches, and user interactions.
- This AI & machine learning layer is fundamental to Jobify's ability to offer personalized and intelligent support to users, ensuring that every user receives the most relevant job opportunities and guidance tailored to their individual career needs.

Integration Layer:

- Jobify's integration layer enables the platform to seamlessly communicate with external services like email servers for notifications, third-party job boards, and other career-related platforms.
- The API gateway serves as the single entry point for external integrations, ensuring secure and efficient data exchange between Jobify and external systems.
- Real-time job notifications, alerts, and application status updates are powered by an external notification service that integrates with mobile and email systems, ensuring that users are always informed.

3.2 System Architecture Diagram

Below is an expanded version of the system's high-level architecture, illustrating how the various components interact within the platform:

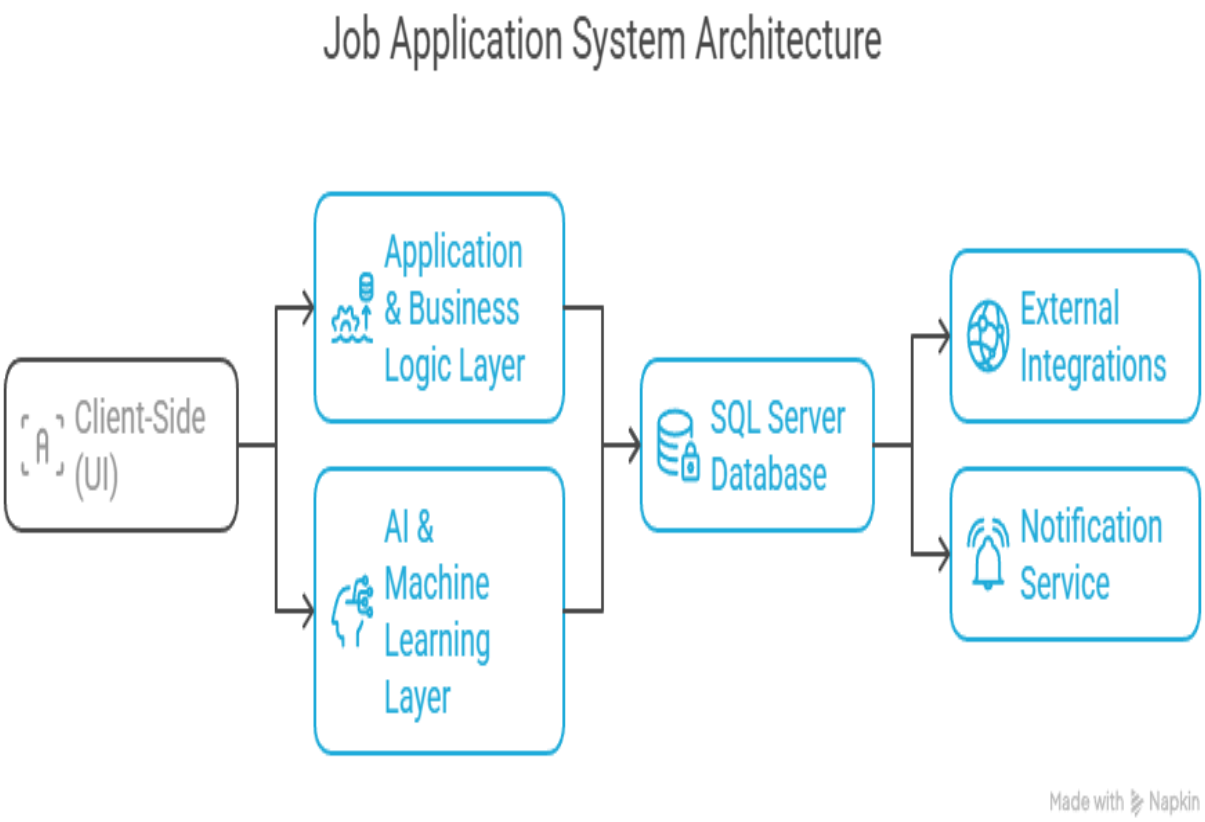


Figure 3.2.1 Architecture Diagram

3.3 Algorithms & Frameworks used

The technology stack used in Jobify is selected to ensure the platform's performance, scalability, and ability to seamlessly integrate advanced AI and machine learning features. Below is a deeper dive into each component of the stack:

Frontend Technologies:

- Angular: Angular is a powerful front-end framework used to create dynamic, single-page applications (SPA). It enables Jobify to provide a seamless user experience with real-time updates and interactive features.
- With its two-way data binding and modular architecture, Angular helps in building efficient and maintainable front-end code, which enhances the overall performance and scalability of the platform.
- Bootstrap: Bootstrap is a widely-used CSS framework that ensures a responsive, mobile-first design.
- By utilizing pre-built components and grid systems, Jobify's user interface (UI) automatically adapts to various screen sizes and devices, making the platform accessible from desktops, tablets, and smartphones.

Backend Technologies:

- ASP.NET Core: ASP.NET Core is a high-performance, cross-platform framework for building APIs and web applications.
- It is used for implementing the core business logic of Jobify, such as managing user accounts, handling authentication, processing job applications, and securely interacting with the database.
- The platform benefits from its robust features like dependency injection, middleware support, and powerful routing capabilities.
- interactions, such as user queries, job listing updates, and AI-driven job recommendations.

- **Database Technologies:**

- SQL Server: SQL Server is a relational database management system (RDBMS) used for managing structured data within Jobify. It ensures data integrity, reliable transactions, and powerful querying capabilities.
- SQL Server is well-suited for handling the platform's large volume of user data, including job listings, user profiles, applications, and interview data. Its ability to handle complex queries and large datasets makes it a robust solution for Jobify's needs.
- Database Schema: The relational nature of SQL Server is beneficial for organizing job seekers, recruiters, job postings, and feedback data into interconnected tables.
- Jobify's database schema is designed to optimize read and write operations while maintaining data consistency across the platform.

AI & Machine Learning Technologies:

- Natural Language Processing (NLP): NLP techniques are integral to Jobify's AI-driven features. NLP is used for analyzing resumes, job descriptions, and interview responses. By leveraging NLP algorithms, Jobify can extract important data from these documents, such as skills, job titles, education, and experiences.
- This data is then processed and matched to provide personalized job recommendations.
- Additionally, NLP enables the AI chatbot to understand and respond to user queries in natural language, improving user engagement.
- Machine Learning Algorithms: Jobify employs machine learning algorithms for continuous improvement of job recommendations and resume matching.
- These models are trained on large datasets of job descriptions and user profiles to learn patterns and optimize matches.
- As more users engage with the platform, the system refines its predictions and recommendations.
- For instance, it can adjust its recommendations based on factors like a user's preferences, industry trends, and feedback from previous matches.

- TensorFlow / PyTorch: TensorFlow and PyTorch are leading open-source machine learning libraries that enable Jobify to develop and deploy complex models.
- These libraries are used to build the AI models for resume matching, job recommendations, and the conversational AI used in the chatbot.
- They support deep learning techniques, which are essential for understanding complex data structures like natural language text and predicting user behavior.

Cloud Infrastructure:

- Microsoft Azure / AWS: Cloud services from Microsoft Azure or Amazon Web Services (AWS) are used to provide the platform's infrastructure.
- Cloud hosting allows Jobify to scale seamlessly by offering on-demand resources, ensuring that the platform can handle high traffic volumes and data processing.
- Features such as load balancing, auto-scaling, and disaster recovery are essential for maintaining performance and uptime.
- Load Balancing: Jobify utilizes cloud-based load balancing services to distribute traffic evenly across multiple servers.
- This helps maintain fast response times and ensures that no single server becomes overloaded, even during peak usage times.
- Containerization (Docker & Kubernetes): Docker is used to containerize Jobify's application, which allows it to run consistently across different environments. Kubernetes is used to orchestrate the deployment, scaling, and management of containers.
- This approach makes it easier to manage the platform's infrastructure, deploy updates, and ensure that the system can handle increased traffic.

Performance & Scalability Considerations:

- Horizontal Scaling: Jobify's architecture supports horizontal scaling, where additional server instances can be added as demand grows.

- This approach ensures that the platform remains responsive and can handle increases in traffic without performance degradation.
- Microservices Architecture (Future Implementation)**:** As Jobify grows, it may transition to a microservices-based architecture.
- Microservices divide the application into smaller, independent services that can be developed, deployed, and scaled separately. This architecture enhances scalability, maintainability, and resilience, as each microservice can be optimized for specific tasks (e.g., user management, job recommendations, chatbots, etc.).
- Database Sharding**:** In anticipation of handling larger user bases, Jobify may implement sharding techniques to divide the SQL Server database into smaller, more manageable segments.
- This ensures high availability and low latency by reducing the load on any single database instance.

Security Considerations:

- Data Encryption: All sensitive user data, such as personal details, resumes, and application information, is encrypted both in transit and at rest, ensuring that the platform meets security and privacy standards.
- Secure Authentication**:** Jobify uses secure authentication mechanisms like OAuth2 and JWT (JSON Web Tokens) to protect user accounts and sessions.
- These ensure that users' personal and professional information is kept secure.
- Regular Security Audits: The system undergoes regular security audits and penetration testing to identify and resolve vulnerabilities, ensuring that the platform remains safe from potential cyber threats.

Scalability & Performance Considerations

Jobify's system is designed to scale efficiently as the number of users and data grows. Below are key considerations to ensure optimal performance:

- Load Balancing: Load balancing techniques ensure that incoming traffic is evenly distributed across multiple server instances.
- This helps in preventing server overloads and ensures that users experience fast response times, regardless of platform demand.
- Microservices Architecture: As mentioned earlier, the future of Jobify may include a transition to a microservices architecture.
- This would allow for more flexible scaling by decoupling the application into smaller, independent services that can scale individually.
- Database Sharding: For extremely large user bases, sharding techniques may be implemented in the SQL Server database to improve performance and reduce the risk of a single point of failure.
- Each shard would store a subset of user data, making it easier to handle large-scale data operations.
- Caching: Jobify employs caching mechanisms (e.g., Redis or Memcached) to store frequently accessed data, reducing database load and speeding up responses.
- This is particularly useful for job search queries, frequently accessed job listings, and user profiles.

Chapter Four
Features of Jobify

Features of Jobify

This chapter provides an in-depth exploration of the core features that power Jobify, highlighting how the platform transforms the job-seeking experience for fresh graduates, entry-level professionals, and career changers. Each feature is thoughtfully designed to address real-world challenges faced by job seekers—ranging from finding the right job match to preparing for interviews and understanding labor market dynamics.

Jobify goes beyond conventional job portals by integrating AI-driven tools, personalized job discovery, and interactive support services that guide users through every stage of their career journey. From intelligent resume matching to detailed job analytics and real-time chatbot assistance, Jobify offers a complete, user-centered employment solution that's both accessible and adaptable.

In this chapter, we outline each main feature, its purpose, and the exact steps users follow to engage with them—ensuring a clear view of the platform's functionality and value.

4.1 Main Features

4.1.1 Login & Signup

Purpose: Secure access for job seekers and recruiters.

User Steps:

1. User lands on the home page and clicks "Login" or "Sign Up."
2. For Signup:
    - Choose role: Job Seeker or Employer.
    - Fill in required fields (name, email, password, confirm password).
    - Verify via email OTP or phone.
    - Redirect to dashboard.
3. For Login:
    - Enter email & password.
    - Option to use Google/Microsoft sign-in.
    - Redirect to personalized dashboard

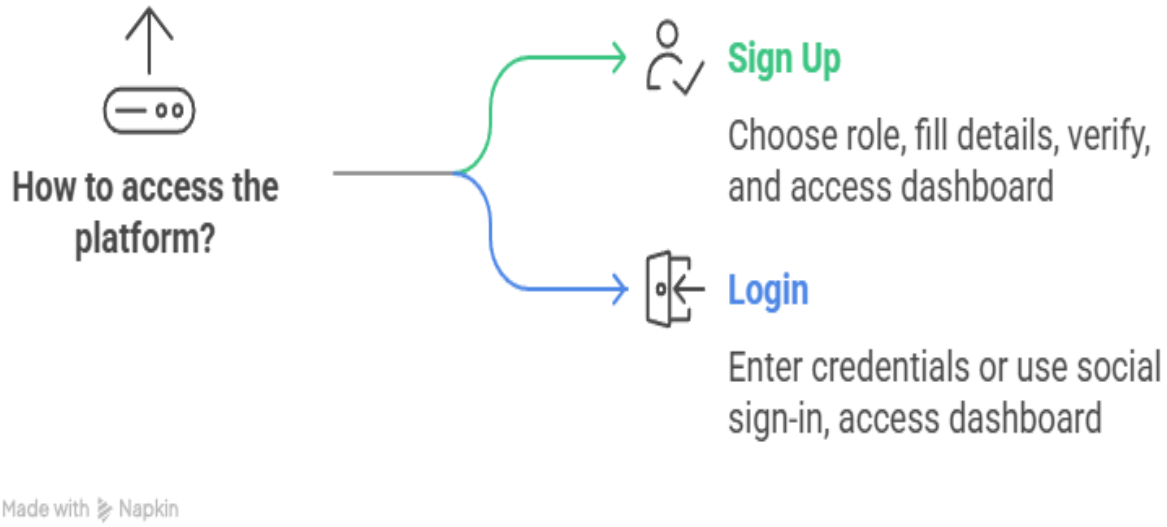


Figure 4.1.1 User Registration

4.1.2 Forget Password

Purpose: Help users recover access to their account.

User Steps:

1. Click on “Forgot Password” on the login page.
2. Enter registered email.
3. Receive a password reset link.
4. Click the link, enter a new password.
5. Confirmation message + redirect to login.

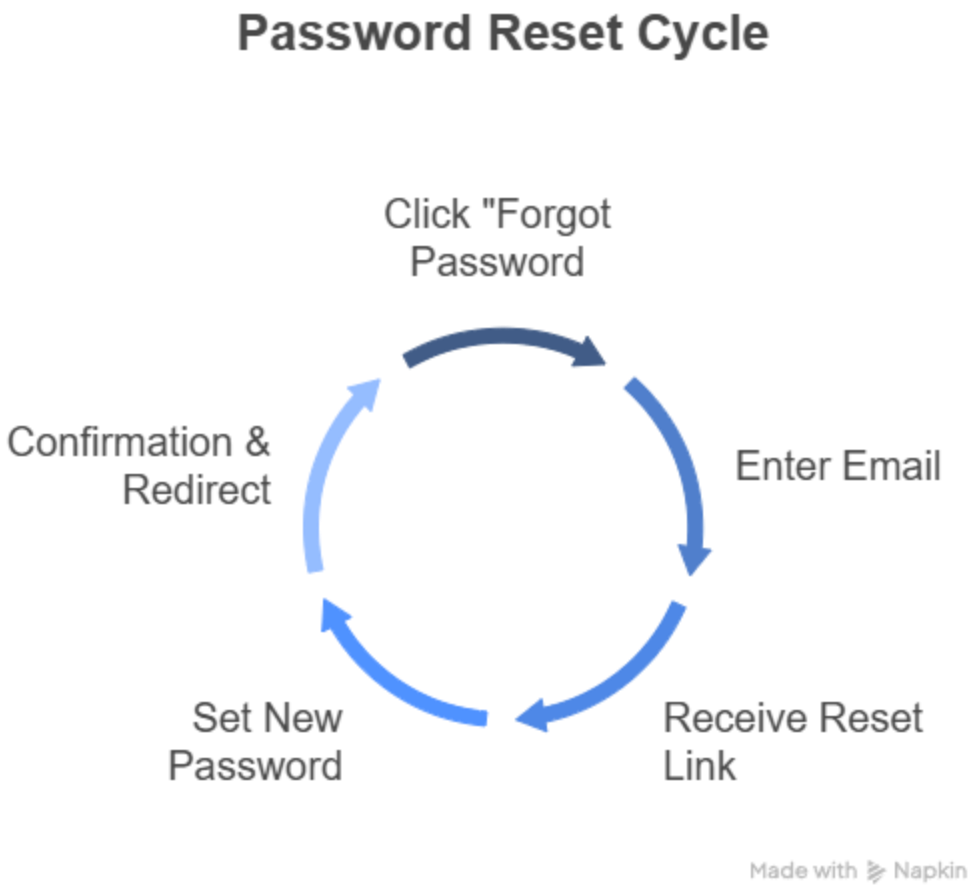


Figure 4.1.2 Password Reset Cycle

4.1.3 Apply for Job

Purpose: Let users submit their applications directly through the platform.

User Steps:

1. After login, user browses job listings.
2. Click on a job → View full description, requirements.
3. Click “Apply Now.”
4. Choose to apply with existing CV or upload a new one.
5. Optional: Write a cover letter or answer screening questions.
6. Submit application → Confirmation notification + email.

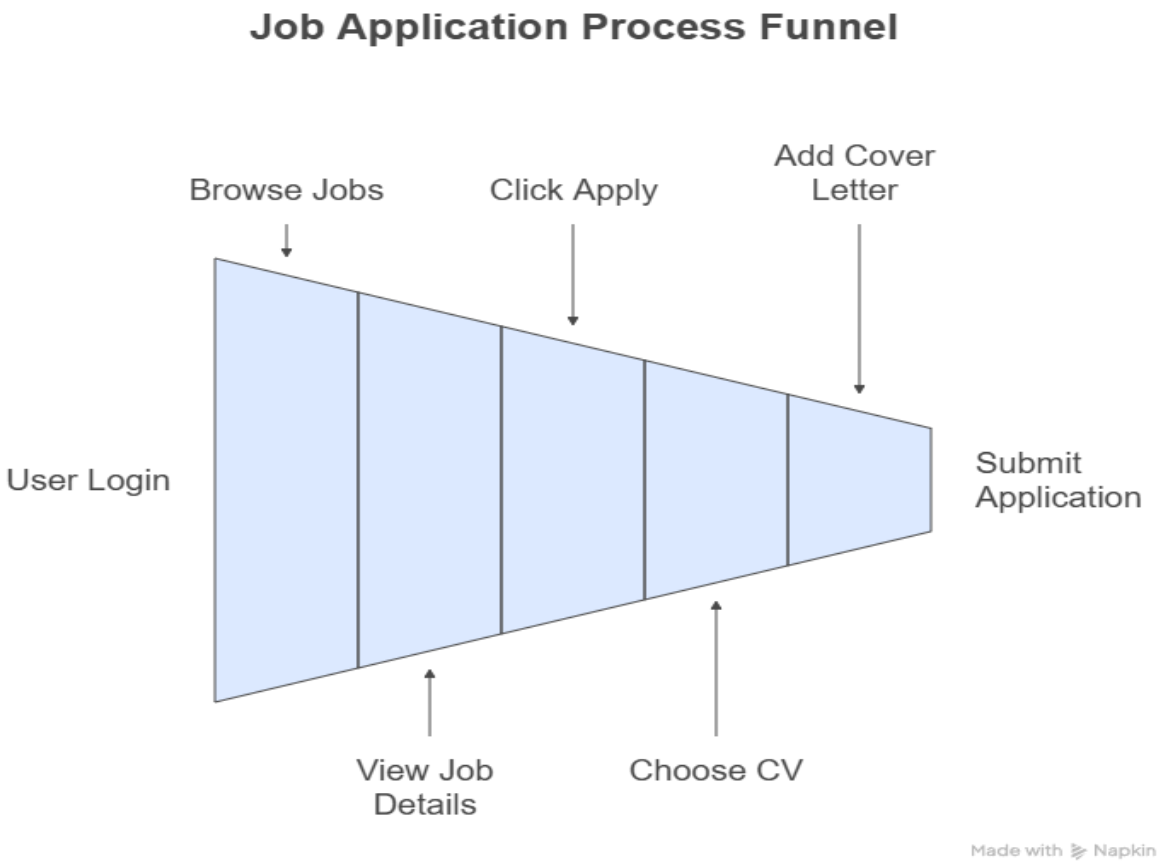


Figure4.1.3 Job Apply Process

4.1.4 Notifications for Job

Purpose: Keep users updated about job matches, deadlines, and interview updates.

User Steps:

1. User receives a real-time notification bell icon in the navbar.
2. Types of notifications:
    - New job matches.
    - Application status updates.
    - Interview invitations.
    - Weekly job market insights.
3. Option to enable/disable specific types of notifications in settings.

Figure 4.1.4 Notifications Cycle

4.1.5 Job Search

Purpose: Allow job seekers to find jobs efficiently.

User Steps:

1. Navigate to “Job Search.”
2. Use filters: location, job type, salary, experience level, skills.
3. Results load dynamically with pagination or infinite scroll.
4. Save favorite jobs or apply directly.

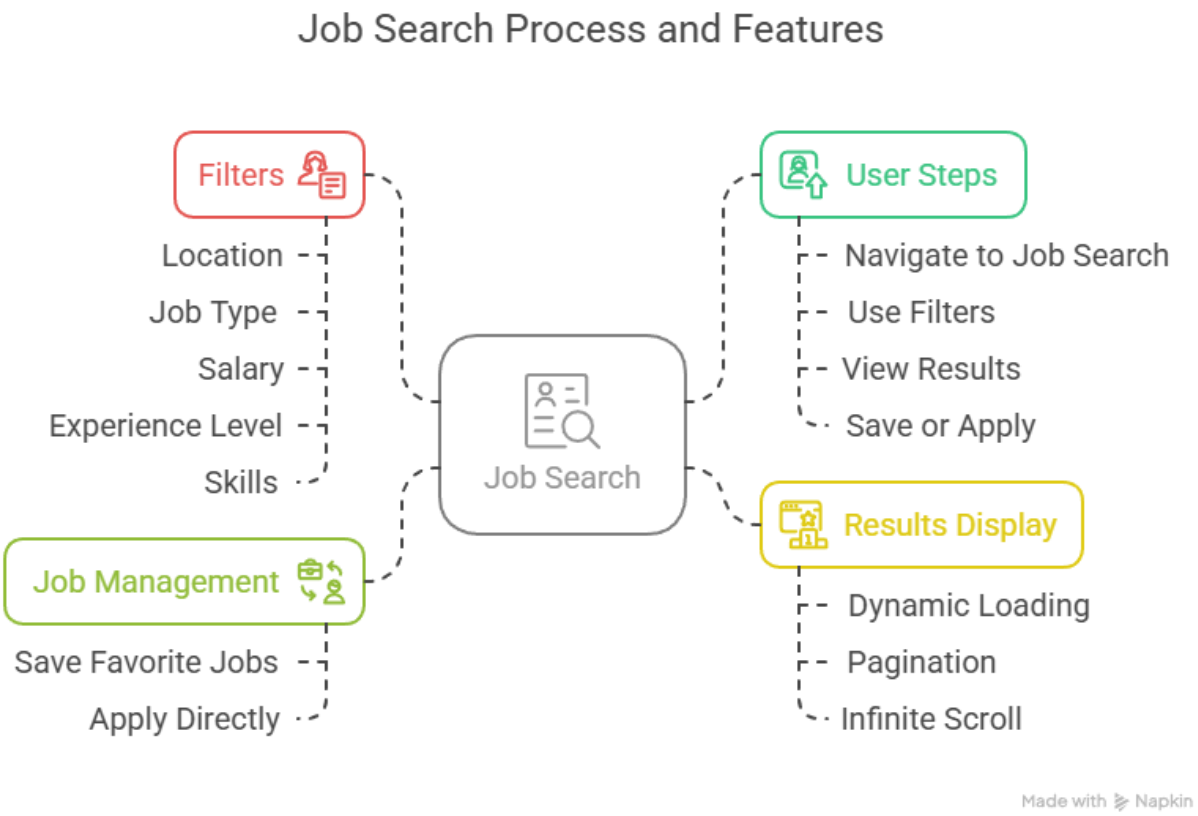


Figure 4.1.5 Job Search Process

4.1.6 AI Resume Matching

Purpose: Recommend jobs that best align with the user's profile.

User Steps:

1. Upload resume or create profile.
2. AI scans skills, experience, and preferences.
3. Matches are displayed under "Recommended Jobs."
4. User can apply or fine-tune preferences.

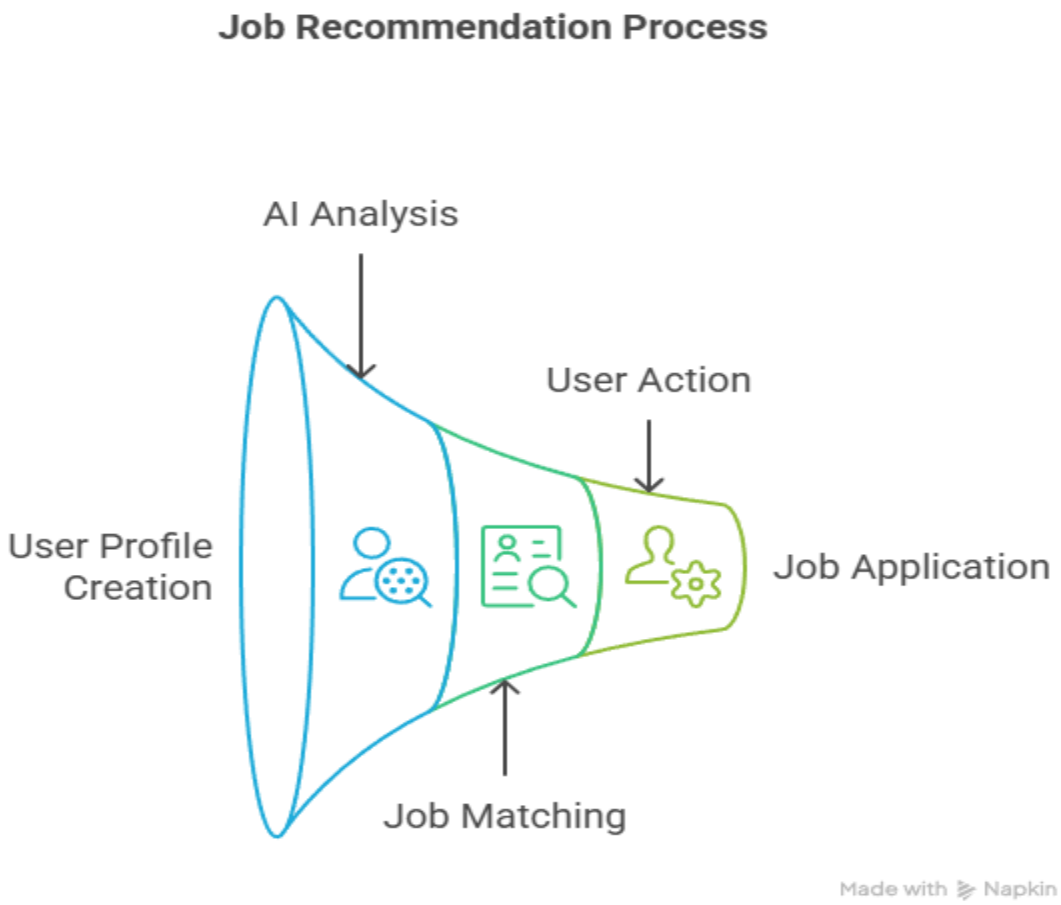


Figure 4.1.6 AI Analysis Process

4.1.7 Mock Interview Preparation

Purpose: Prepare users for real interviews.

User Steps:

1. Navigate to “Interview Practice.”
2. Select job type or industry.
3. User records answers or types responses.
4. Instant feedback is provided (tone, clarity, relevance).
5. History of mock sessions is stored for review.

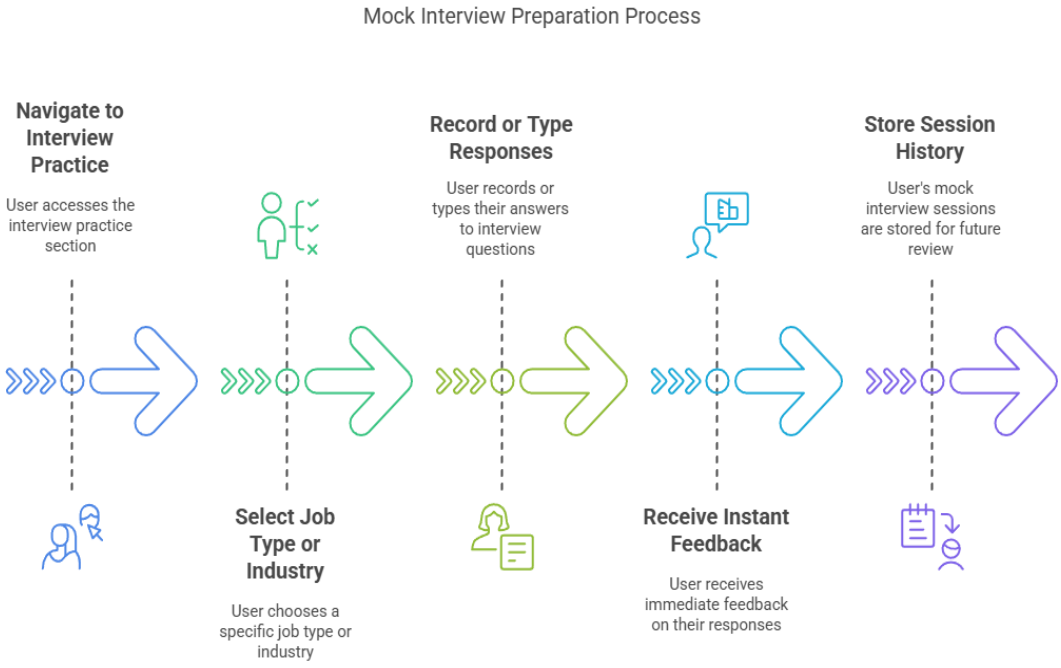


Figure 4.1.7 Interview process

4.1.8 Job Analytics and Insights

Purpose: Help users understand market trends and tailor their job search.

User Steps:

1. Go to “Job Insights” dashboard.
2. View salary trends, in-demand skills, popular roles in selected industries.
3. Use graphs, heatmaps, or charts.
4. Adjust filters by region, job type, or experience level.

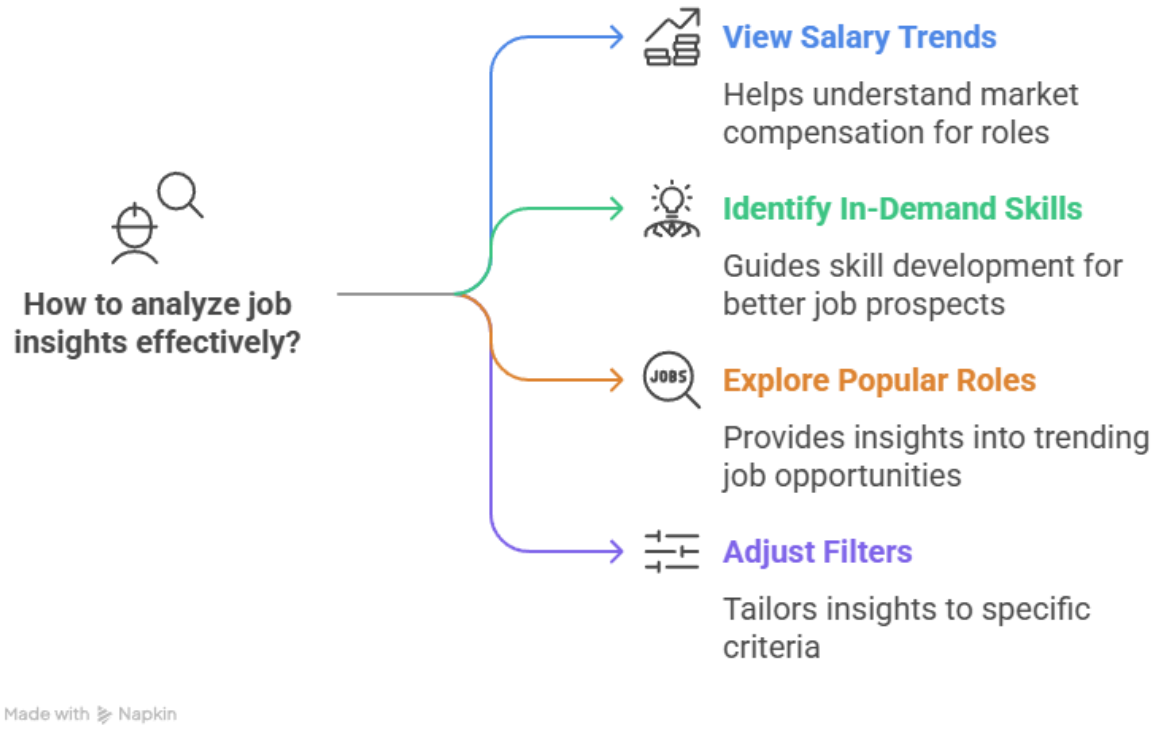


Figure 4.1.8 Job Analytics

4.1.9 Internship & Part-time Listings

Purpose: Support early-career individuals.

User Steps:

1. Choose filter: “Internship” or “Part-time.”
2. Browse curated listings ideal for students or fresh grads.
3. View eligibility (e.g., school year, field of study).
4. Apply like any regular job with simplified CV.

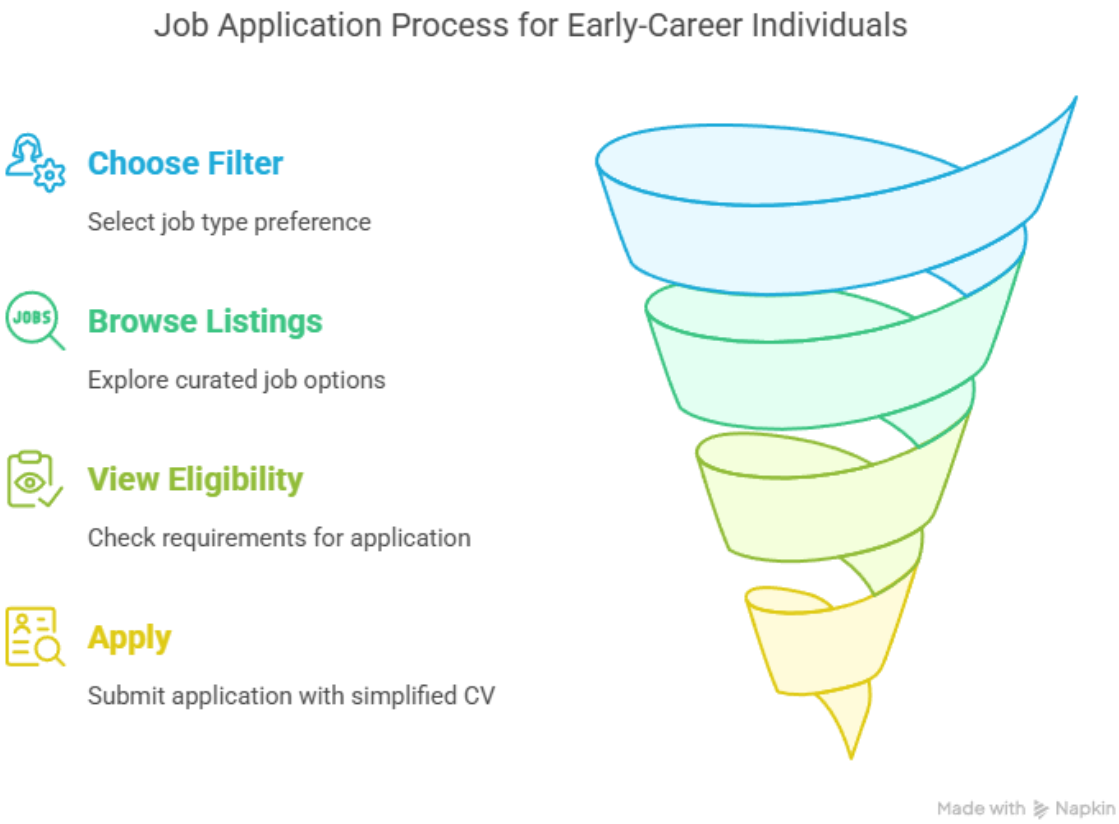


Figure 4.1.9 Job List Process

4.2 AI & Machine Learning Features

4.2.1 CV Analyzer

Purpose: Help users improve resumes before applying.

User Steps:

1. Navigate to “CV Analyzer.”
2. Upload resume (PDF, DOCX).
3. AI provides:
    - Skill gap analysis.
    - Formatting suggestions.
    - Keyword relevance (based on selected job roles).
4. Option to download improved version or use built-in resume builder.

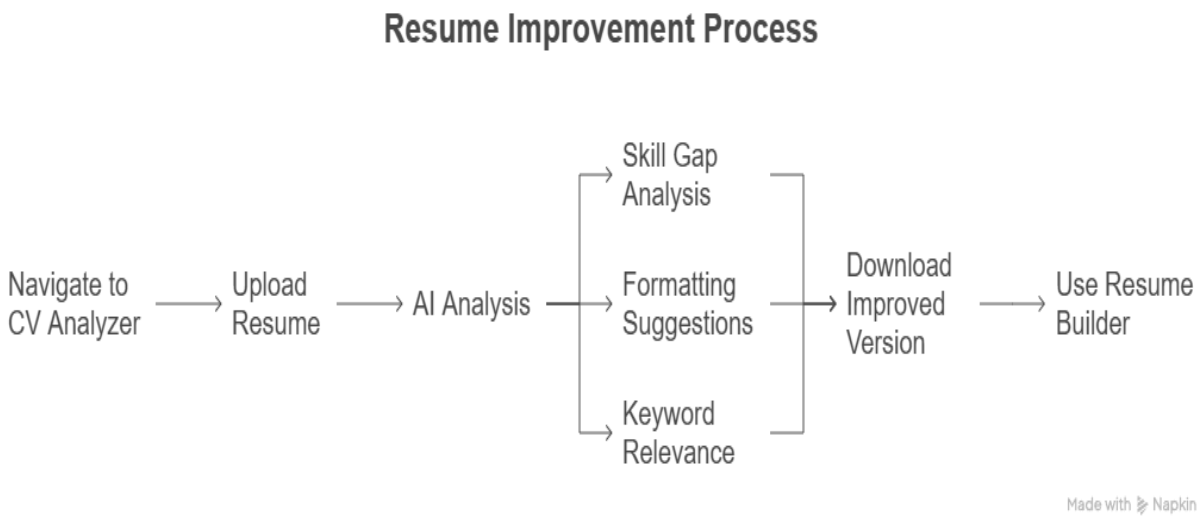


Figure 4.2.1 Resume Process

4.2.2 Chatbot Assistant

Purpose: Provide 24/7 support and guidance.

User Steps:

1. Click chatbot icon on any page.
2. Ask questions like:
    - “What jobs fit me?”
    - “How do I prepare for an interview?”
    - “Update my profile.”
3. Chatbot guides with:
    - Links to platform tools.
    - Personalized advice.
    - General help (reset password, navigation, etc.).

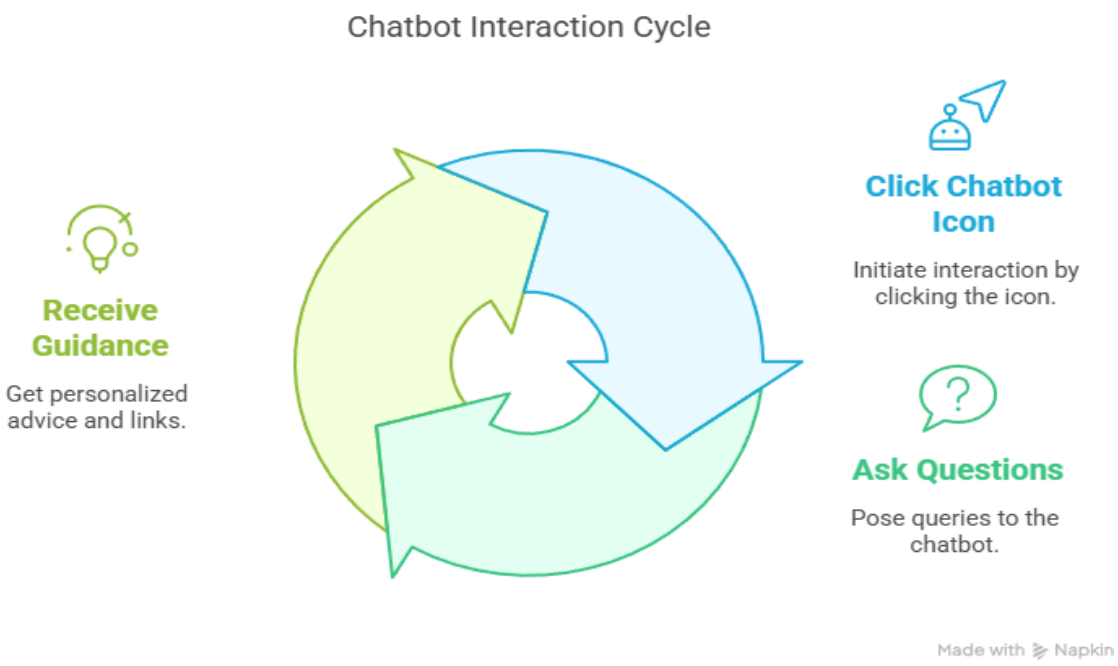


Figure 4.2.2 Chatbot Interaction Cycle

4.3 Sub Features & Inside Pages

- Services Page: Overview of Jobify services (AI matching, mock interviews, job analytics).
- Job Listing Page: Dynamic job board with real-time filters and sorting.
- Company Page: Detailed profiles of hiring companies with reviews and job openings.
- Contact Page: Support form, social media links, and chatbot integration.
- Subscribe for Job Alerts**:** Email opt-in to get weekly/monthly curated job alerts.
- User Profile**:** Edit CV, cover letters, preferences, saved jobs, and activity logs.
- Login/Signup/Forget Password: Unified auth system.
- Apply Now Button: On every job card or detail page.
- Notifications Page: Centralized alert management for job seekers.

Chapter Five
Tools & Technologies Used

Tools & Technologies Used

The successful development of **Jobify** depends on a robust set of tools, frameworks, and technologies that ensure performance, scalability, and seamless integration of advanced features such as AI-powered resume matching and real-time chatbot support.

This chapter provides a comprehensive overview of the technologies used across the frontend, backend, database, and AI components of the platform.

Jobify's architecture is built on a modern full-stack ecosystem that supports robust business logic, dynamic user interfaces, and AI-driven functionalities.

This section offers a detailed walkthrough of each technology, how it was integrated, and why it was chosen, including the specific development tools and environments used such as Visual Studio**,** Google Colab**,** and SQL Server Management Studio**.**

5.1 Frontend Development (User Interface)

The frontend is designed to be modular, interactive, and scalable, with a consistent and professional look that adapts across screen sizes.

Technologies Used:

- **Angular (v18)**
  - Project Setup: Created using Angular CLI:
    `ng new jobify --routing --style=scss`
  - Structure: Component-based architecture (`login.component`, `dashboard.component`, `job-card.component`) for maintainability.

- Services: Used for HTTP communication with backend APIs (e.g., `job.service.ts`, `resume.service.ts`).
- Forms: Reactive Forms (`FormGroup`, `FormBuilder`) are used for signup, login, and resume creation.

- **Bootstrap 5**
  - Integrated using `npm install bootstrap`.
  - Utilized for:
    - Grid layout (`col-lg-6`, `container-fluid`)
    - Reusable components like modals, cards, badges, and buttons.
    - Custom breakpoints for mobile-first design.
- **HTML5 & CSS3**
  - HTML templating used in Angular views (`.html` files).
  - SCSS used for styling, with global variables for theme control.
  - CSS animations are implemented in feedback modals and transition effects between components.

🔗 Tools:

- **Visual Studio Code**
  - Extensions used: Angular Snippets, ESLint, Prettier.
  - Live Server plugin for rapid frontend testing.
  - Terminal integration for Angular CLI operations.

5.2 Backend Development (Business Logic & API)

The backend serves as the core of Jobify, managing business logic, authentication, database transactions, and AI integrations.

Technologies Used:

- **ASP.NET Core (v7)**
  - Developed in: Visual Studio 2022
  - API structure:
    - `UsersController`: Login, Register, Profile.
    - `JobsController`: Job listings, filters, applications.
    - `AIController`: Resume analysis, chatbot queries.
  - Features:
    - JWT-based authentication and role management.
    - Dependency Injection for modular services.
    - Data validation using Data Annotations.
- **Entity Framework Core**
  - Code-first approach:
    - Created model classes like `User.cs`, `Job.cs`, `Resume.cs`.
    - Used `DbContext` for database operations.
    - Created migration with `Add-Migration InitialCreate` and
    - `Update-Database`.
    -

🔧 *Tools:*

- **Postman**
  - Used to test APIs during development.
  - Collection shared for team testing and debugging.
- **Swagger**
  - Enabled with `Swashbuckle.AspNetCore` for API documentation and live testing during development.

5.3 Database Management (Data Layer)

A strong relational model is necessary for Jobify's structured data — users, job listings, companies, applications, and resumes.

🧱 Technologies Used:

- **Microsoft SQL Server**
  - Developed and tested using **SQL Server Management Studio (SSMS)**.
  - Database schema includes tables:
    - `Users`, `Jobs`, `Resumes`, `Applications`, `Notifications`, `Companies`.
- **Stored Procedures**:
  - `GetRecommendedJobs` – Filters jobs based on user skills.
  - `GetCompanyStats` – Retrieves company-specific analytics.
- **Entity Framework Core Integration**:
  - Models synchronized with DB using Migrations.
  - LINQ queries used for advanced filters and joins (e.g., "get jobs by skills and location").

Tools:

- SSMS: For manual query testing, performance monitoring, and backups.
- EF Core CLI: For applying migrations and database updates via terminal.

5.4 AI Integration (Smart Features Layer)

Jobify's AI module provides intelligent user experiences like resume matching and chatbot guidance.

Technologies Used:

- **Python (AI Models)**
  - Developed and trained in Google Colab using:
    - spaCy for NLP
    - scikit-learn for training models
    - transformers (HuggingFace) for deep learning-based NLP
- **Resume Analyzer**
  - Reads uploaded resumes (PDF or DOCX).
  - Extracts skills, degrees, experiences using spaCy and compares them to job descriptions.
  - Model trained on dataset of 5,000+ resumes and job listings.
- **Chatbot Assistant**
  - Developed with rule-based NLP in spaCy.
  - Flask used to deploy the chatbot model.
  - Backend requests are sent from Angular to .NET API, which forwards to Flask microservice.

Tools:

- **Google Colab**
  - For model training and testing.
  - Integrated with Google Drive for data persistence.
- **Flask**
  - AI microservices exposed via RESTful endpoints like:
    - `/api/resume/analyze`
    - `/api/chatbot/message`
- **PyTorch/TensorFlow**
  - For more complex AI experiments (e.g., transformer-based resume understanding).

5.5 Integration & Deployment

- **Docker**
  - Used to containerize backend, frontend, and AI services.
  - Docker Compose used to link services.
- **Azure Cloud**
  - Hosting for backend APIs and SQL Server.
  - Blob Storage for resume uploads.
  - Azure App Service for deploying Flask microservices.
- **CI/CD**
  - Using GitHub Actions for test/deploy pipelines.
  - Code pushed to main triggers:
    - Build Angular app
    - Build .NET backend
    - Deploy to Azure

Stack Technology & Tools Conclusion

Jobify's Technology Stack Comparison

Characteristic	Frontend Development	Backend Development	Database Management	AI Integration	Integration & Deployment
Core Technologies	Angular, Bootstrap, HTML5, CSS3	ASP.NET Core, Entity Framework Core, Node.js	Microsoft SQL Server, Stored Procedures, EF Core	Python, spaCy, scikit-learn, transformers	Docker, Azure Cloud, CI/CD
Key Structures	Component-based architecture, Reactive Forms	API Controllers, JWT Authentication, Dependency Injection	Tables, Models synchronized with DB, LINQ queries	Resume Analyzer, Chatbot Assistant	Containerization, Cloud Hosting, Automated Pipelines
Primary Tools	Visual Studio Code	Visual Studio, Postman, Swagger	SSMS, EF Core CLI	Google Colab, Flask, PyTorch/TensorFlow	Docker Compose, Azure App Service, GitHub Actions

Figure 5 Chapter(5) Conclusion

Chapter Six
Implementation Details

Implementation Details

This chapter presents a detailed overview of the technical implementation of Jobify, covering each core system component including the frontend, backend, database, and AI-powered modules.

It explains how the technologies were applied in practice, the development workflows followed, and how different parts of the system interact to deliver a cohesive user experience.

The platform was built using a modular architecture to ensure scalability, maintainability, and seamless integration of intelligent features.

Development environments included Visual Studio for backend and API logic, Google Colab for AI model development, and SQL Server Management Studio for managing relational data.

Each subsection of this chapter walks through the step-by-step implementation process, tools used, and integration strategies.

From the dynamic interface developed with Angular and Bootstrap to the secure backend APIs built in .NET, and the intelligent resume analyzer powered by Python and AI models, this chapter highlights the real-world engineering decisions and efforts behind the Jobify platform.

By deeply examining each technology and its role within the system, this chapter provides a complete picture of how Jobify was brought to life from concept to functional application.

6.1 Frontend (Angular + Bootstrap)

The frontend of Jobify was built using Angular and Bootstrap, focusing on delivering a responsive, interactive, and intuitive user experience. The design and architecture of the frontend aim to support seamless navigation, dynamic content rendering, and integration with backend APIs and AI services.

6.1.1 Technology Stack

- **Angular (v18)**: A powerful framework for building single-page applications (SPA) with modularity, routing, and reactive components.
- **Bootstrap (v5)**: A CSS framework used to implement responsive design, grid layout, UI components, and mobile-first visuals.
- **HTML5 & CSS3**: For structural layout and visual styling.
- **TypeScript**: Angular's primary language, used for component logic and strong typing.
- **RxJS**: For handling reactive programming and asynchronous data flows (observables).
- **Angular CLI**: Used for scaffolding, development server, building and testing.

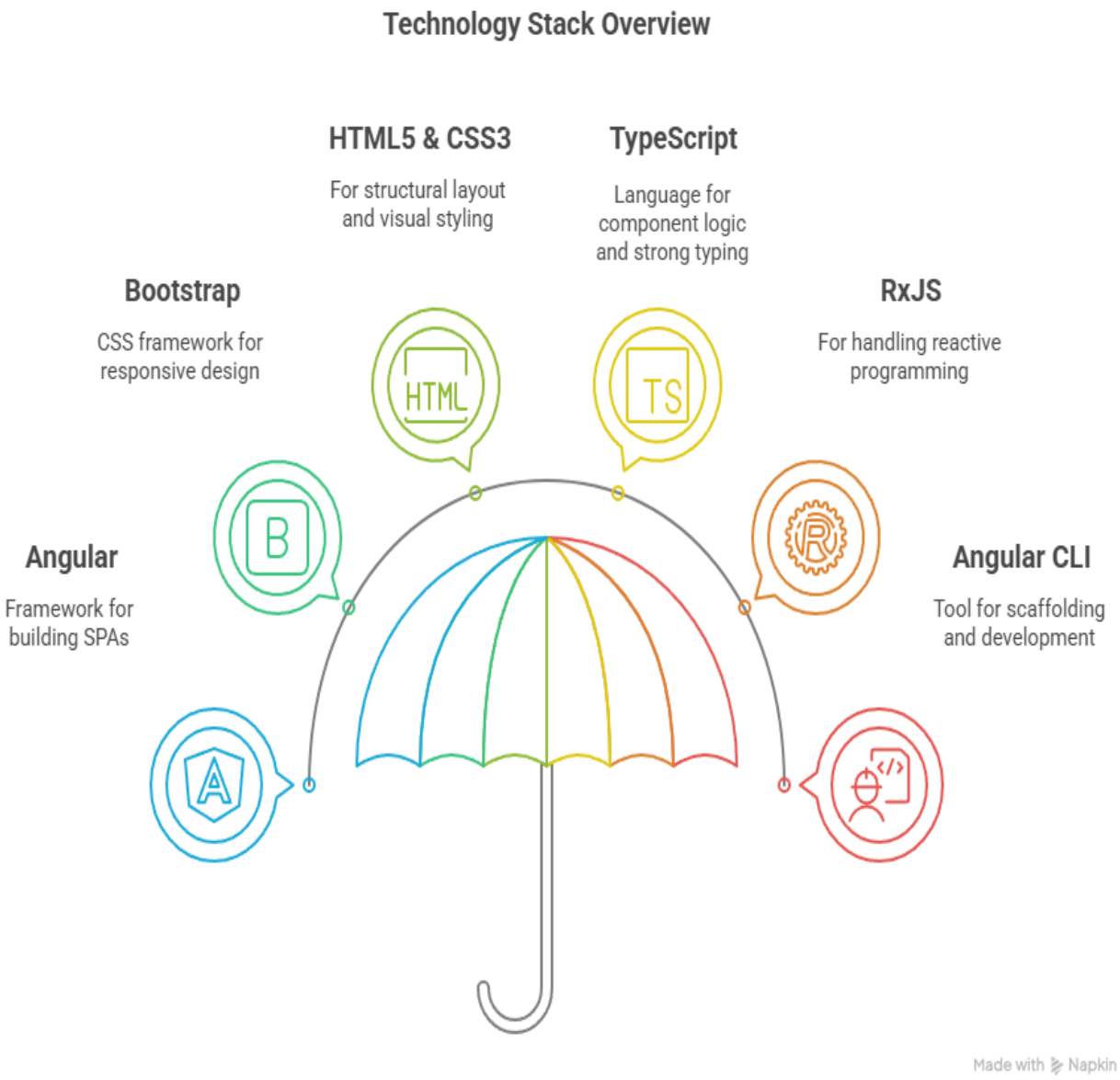


Figure 6.1.1 FrontEnd Stack

6.1.2 Folder Structure

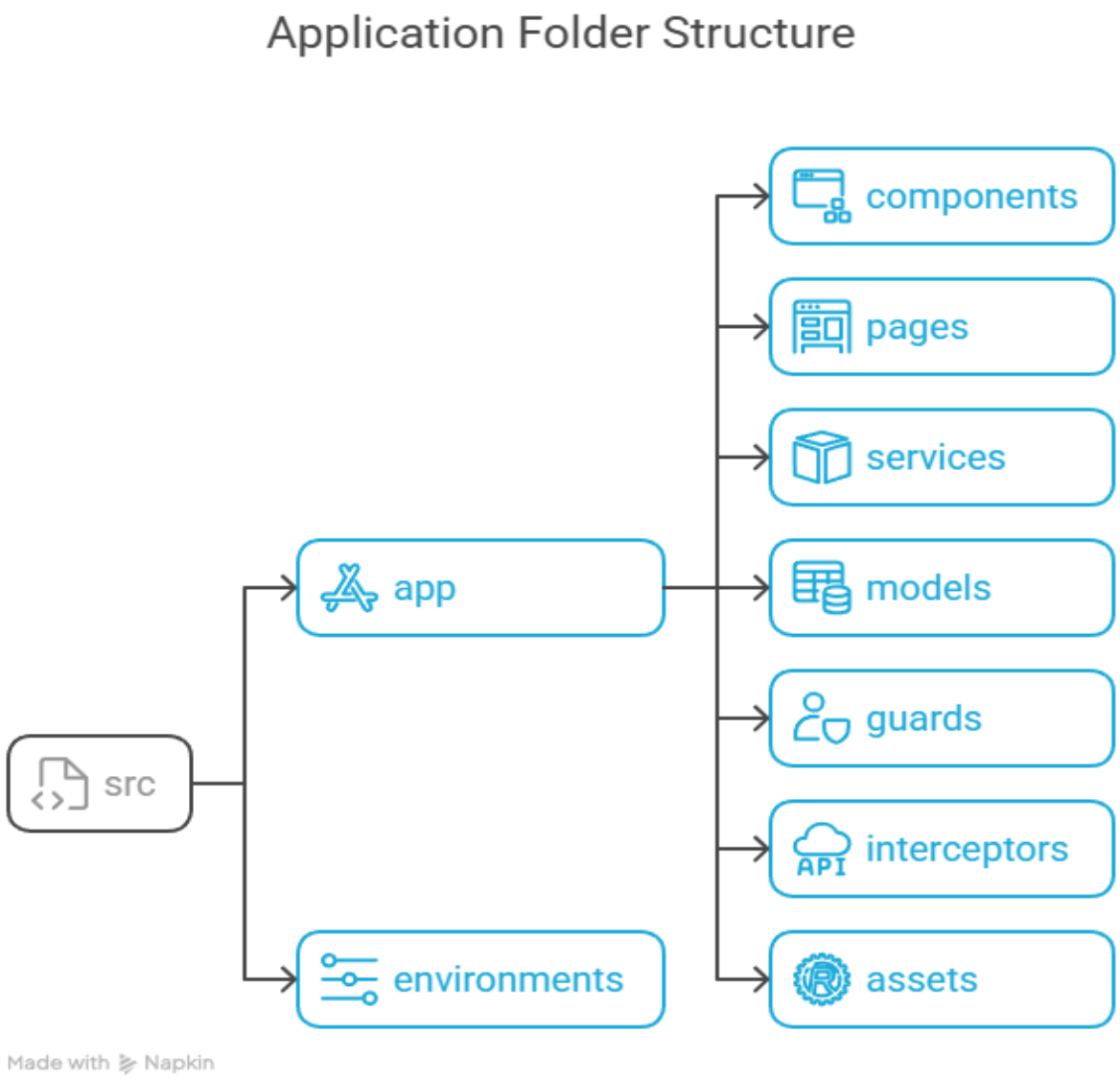


Figure 6.1.2 Application Structurre

6.1.3 Key Pages & Components

Figure 6.1.3 Page Component

6.1.4 Key Functionalities

- **Home Page :**

```html
<div class="container mt-5">
  <h1 class="text-center fw-bold main-title">Featured Jobs</h1>
  <p class="text-center title">
    The #1 Job Board for Hiring Creative Professionals
  </p>
  <div class="row section-main" [routerLink]="['/job-list']">
    <div class="col-lg-6 col-md-6 col-sm-12" *ngFor="let job of jobs">
      <div class="job-card fade-in-up" appScrollAnimate>
        <div class="job-header d-flex">
          <img [src]="job.imgSrc" [alt]="job.title" class="job-image" />
          <div class="info" appScrollAnimate>
            <a class="fw-bold m-auto d-flex">{{ job.title }}</a>
            <a class="fw-bold a-custom" [routerLink]="['/company']">{{
              job.company
            }}</a>
            <span class="location-custom">
              <i class="fa-solid fa-location-dot"></i>
              {{ job.location }}
            </span>
            <span class="job-custom">
              <i class="fa-solid fa-briefcase"></i>
              {{ job.jobType }}
            </span>
            <span class="time-custom">
              <i class="fa-regular fa-clock"></i>
              {{ job.postedTime }}
            </span>
            <span class="price">{{ job.salary }}</span>
          </div>
        </div>
        <p class="main-text">{{ job.description }}</p>
        <span *ngIf="job.urgent" class="btn btn-1-color">Urgent</span>
        <span *ngIf="job.senior" class="btn btn-2-color">Senior</span>
        <span *ngIf="job.fullTime" class="btn btn-3-color">
          <i class="fa-solid fa-circle"></i> Full time
        </span>
        <div class="icons">
          <a href="#"><i class="fa-regular fa-circle-check"></i></a>
          <a href="#"><i class="fa-regular fa-bookmark"></i></a>
        </div>
      </div>
    </div>
  </div>
  <button class="browse-more-btn" [routerLink]="['/job-list']" appScrollAnimate>
    Browse More
    <i class="fa-solid fa-angles-right"></i>
  </button>
</div>
<app-how-work></app-how-work>
<app-trusted-companies></app-trusted-companies>
<app-subscription></app-subscription>
<app-about-us></app-about-us>
```

Addition info and code in the link of Repo on GotHub Link In Refrence

- **Routing & Navigation**:
    - Implemented with Angular Router to manage page transitions without reloading.
    - Route Guards for protected routes (e.g., dashboard, apply pages).

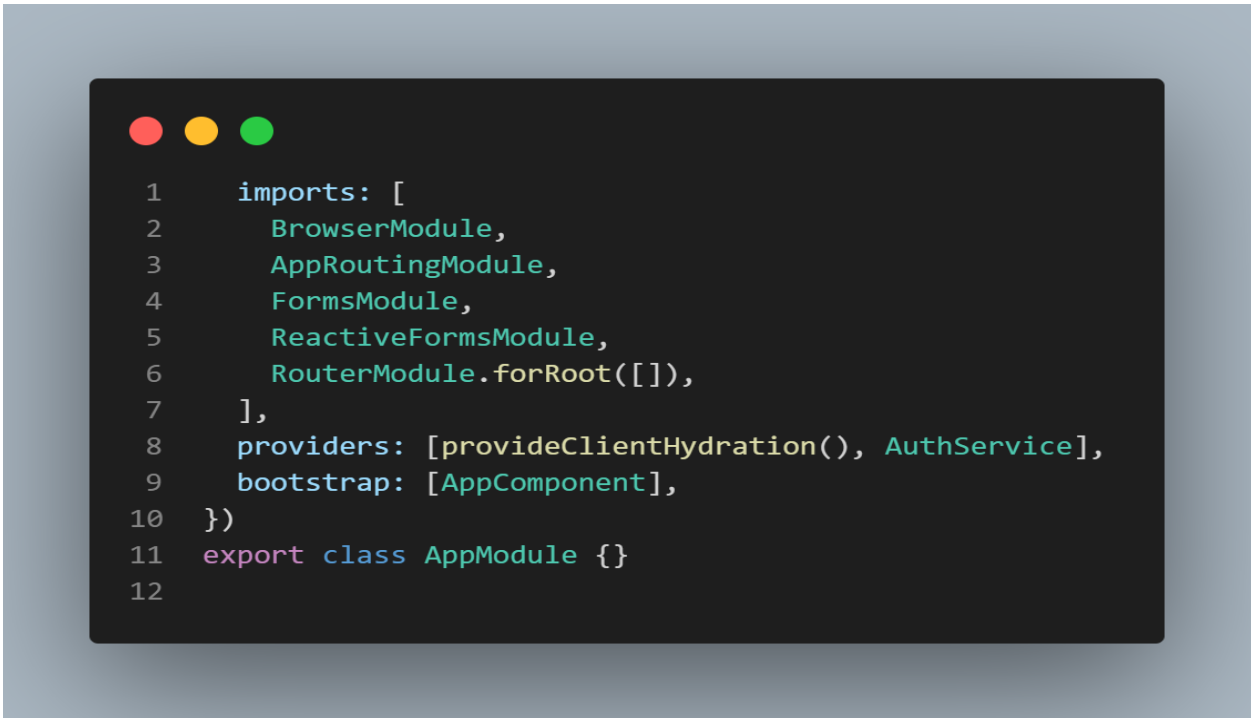
```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { HomeComponent } from './home/home.component';
import { LoginComponent } from './login/login.component';
import { SignupComponent } from './signup/signup.component';
import { ForgetPassComponent } from './forget-pass/forget-pass.component';
import { ProfileComponent } from './profile/profile.component';
import { JobListComponent } from './job-list/job-list.component';
import { JobDetailsComponent } from './job-details/job-details.component';
import { ApplyNowComponent } from './job-details/apply-now/apply-now.component';
import { NotFoundComponent } from './not-found/not-found.component';
import { NotificationComponent } from './notification/notification.component';
import { CompanyComponent } from './company/company.component';
import { SavedJobComponent } from './saved-job/saved-job.component';
import { ServicesPageComponent } from './services-page/services-page.component';
import { AboutComponent } from './about/about.component';
import { ContactComponent } from './contact/contact.component';
import { MockInterviewComponent } from './mock-interview/mock-interview.component';

import { AuthGuard } from './guards/auth.guard';
```

```
const routes: Routes = [
  { path: 'signup', component: SignupComponent },
  { path: 'login', component: LoginComponent },
  { path: 'forget-pass', component: ForgetPassComponent },
  { path: 'profile', component: ProfileComponent, canActivate: [AuthGuard] },
  { path: 'home', component: HomeComponent, canActivate: [AuthGuard] },
  { path: 'job-list', component: JobListComponent, canActivate: [AuthGuard] },
  { path: 'company', component: CompanyComponent, canActivate: [AuthGuard] },
  { path: 'saved-job', component: SavedJobComponent, canActivate: [AuthGuard] },
  {
    path: 'services-page',
    component: ServicesPageComponent,
    canActivate: [AuthGuard],
  },
  { path: 'about', component: AboutComponent, canActivate: [AuthGuard] },
  {
    path: 'contact',
    component: ContactComponent,
    canActivate: [AuthGuard],
  },
  {
    path: 'mock-interview',
    component: MockInterviewComponent,
    canActivate: [AuthGuard],
  },

  {
    path: 'notification',
    component: NotificationComponent,
    canActivate: [AuthGuard],
  },
  { path: 'apply-now', component: ApplyNowComponent, canActivate: [AuthGuard] },
  {
    path: 'job-details',
    component: JobDetailsComponent,
    canActivate: [AuthGuard],
  },
  { path: 'apply-now', component: ApplyNowComponent, canActivate: [AuthGuard] },
  { path: '', redirectTo: '/home', pathMatch: 'full' },
  { path: '**', component: NotFoundComponent }, // Wildcard route for 404
];

```

Reactive Forms:

- Used for login, signup, CV upload, and application forms.
- Includes field validation, error handling, and API submission.

```
import { NgModule } from '@angular/core';
import {
  BrowserModule,
  provideClientHydration,
} from '@angular/platform-browser';
```



```
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule,
    ReactiveFormsModule,
    RouterModule.forRoot([]),
  ],
  providers: [provideClientHydration(), AuthService],
  bootstrap: [AppComponent],
})
export class AppModule {}

```

Service Integration:

- Angular services are used to call .NET backend APIs (e.g., job listings, auth).
- HTTP interceptors attach JWT tokens to requests for secure access.

```typescript
import { Injectable } from '@angular/core';
import { Observable, of } from 'rxjs';

@Injectable({
  providedIn: 'root',
})
export class AuthService {
  private loggedIn = false;

  signup(userDetails: any): Observable<boolean> {
    // Store user details in localStorage (For demo purposes, storing entire user info)
    localStorage.setItem('user', JSON.stringify(userDetails));
    return of(true);
  }

  login(credentials: any): Observable<boolean> {
    // Check if user exists in localStorage
    const storedUser = localStorage.getItem('user');
    if (storedUser) {
      const user = JSON.parse(storedUser);

      // Check if the entered email exists
      if (credentials.email === user.email) {
        // Validate the password as well
        if (credentials.password === user.password) {
          this.loggedIn = true;
          localStorage.setItem('loggedIn', 'true');
          return of(true); // Valid login
        } else {
          return of(false); // Invalid password
        }
      } else {
        return of(false); // Invalid email
      }
    }
    return of(false); // No user found in localStorage
  }

  isLoggedIn(): boolean {
    return localStorage.getItem('loggedIn') === 'true';
  }

  logout(): void {
    this.loggedIn = false;
    localStorage.removeItem('loggedIn');
  }
}

```

Responsive Design:

- Bootstrap grid and components ensure full responsiveness across devices.
- Mobile-first UI with adaptive menus, forms, and cards.

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="description" content="JOBIFY" />
    <meta name="theme-color" content="#9777fa" />
    <link
      rel="icon"
      type="image/x-icon"
      href="assets\images\Logo\Job-Logo.png"
    />
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
      crossorigin="anonymous"
    />
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9
```

6.1.5 Development Tools & Workflow

- **IDE**: Visual Studio Code
- **Version Control**: Git with GitHub
- **Dev Server**: `ng serve` for live preview
- **Linting & Formatting**: ESLint and Prettier
- **Testing**: Jasmine and Karma for unit testing of components
- **Deployment**: Built and deployed as static files on hosting platforms (e.g., Azure Static Web Apps or Firebase Hosting , Vercel)

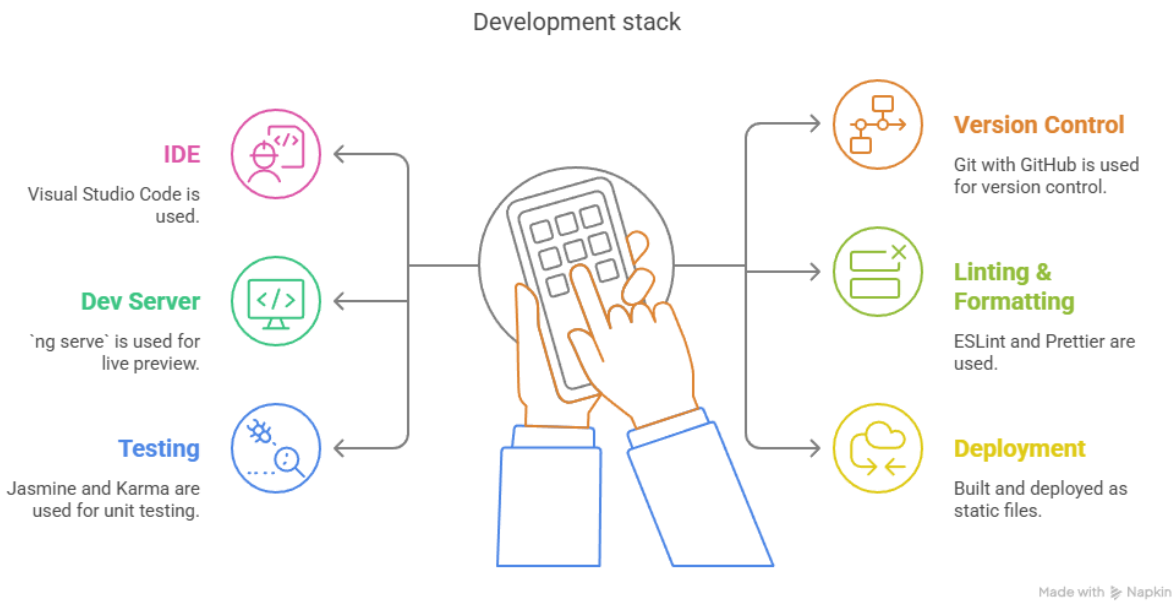


Figure 6.1.5 Tools Stack

6.2 Backend (.NET)

The backend of Jobify is built using ASP.NET Core**,** a powerful and high-performance framework developed by Microsoft for creating web APIs and dynamic server-side logic. This layer is responsible for managing data transactions, enforcing business rules, securing the platform, and integrating with AI modules and the SQL Server database.

6.2.1 Why ASP.NET Core?

- **Cross-platform**: Runs on Windows, macOS, and Linux.
- **Performance**: Extremely fast response times under heavy load.
- **Security**: Built-in support for secure authentication (JWT, OAuth).
- **Scalability**: Designed for microservices and enterprise-grade applications.
- **Dependency Injection**: Built-in support for modular, testable code.

6.2.2 Core Functional Modules

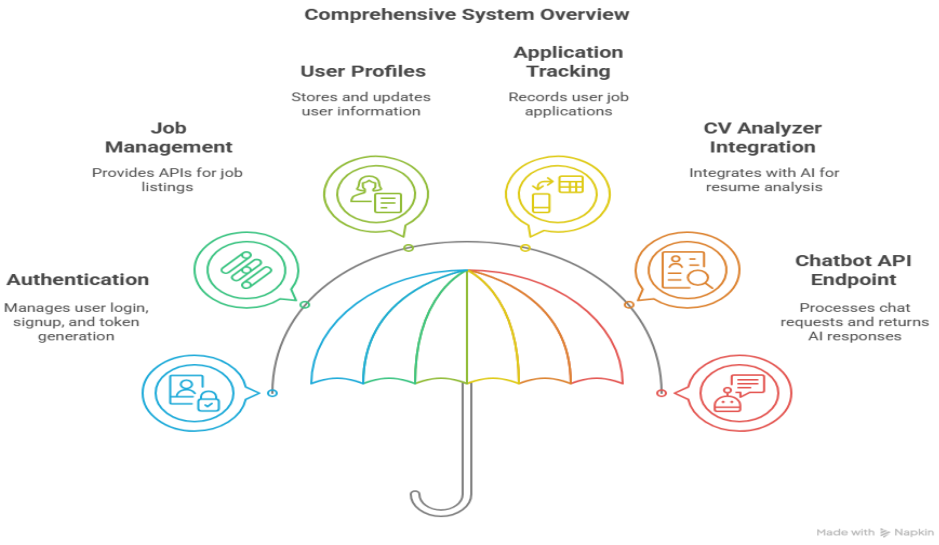
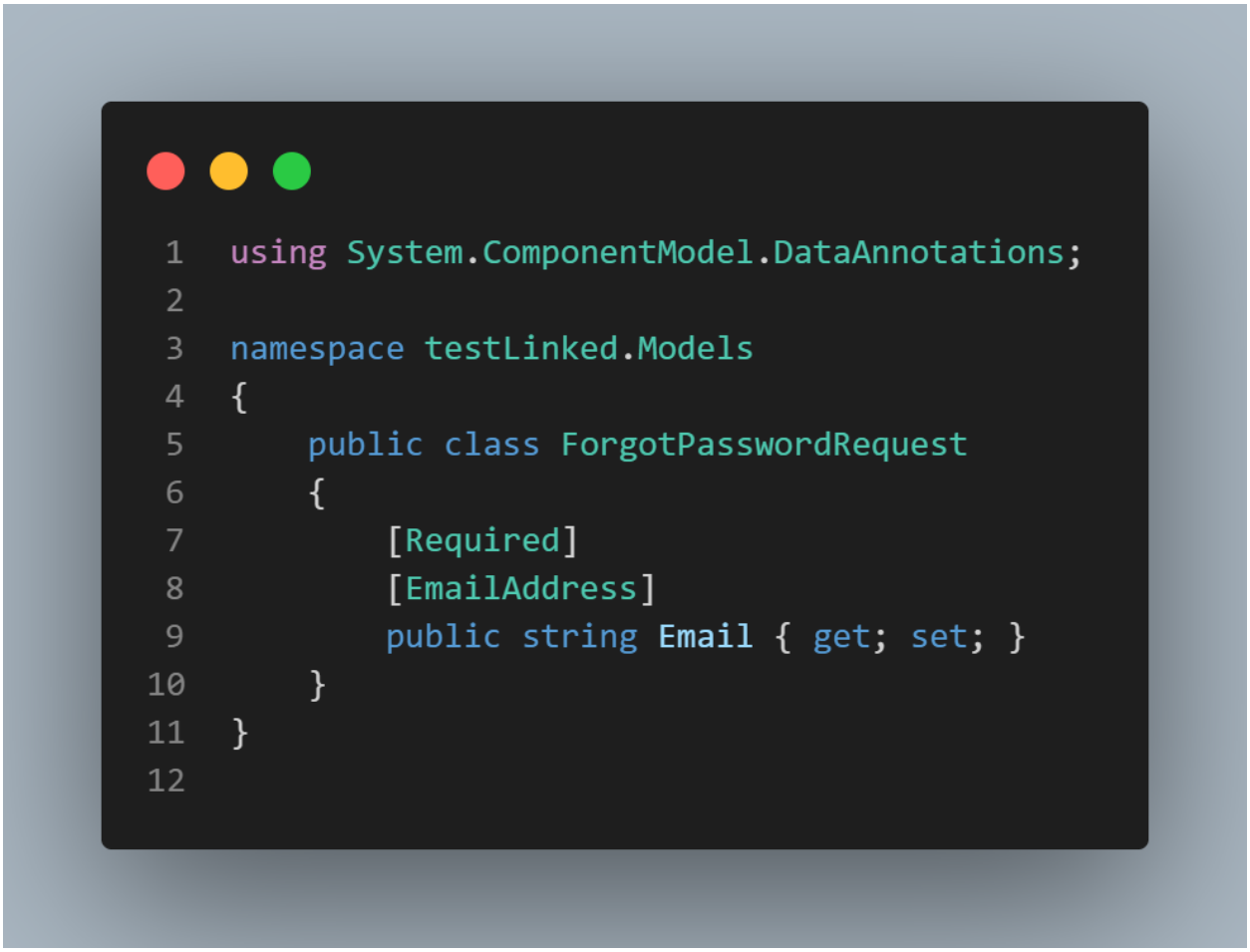


Figure 6.2.2 Core Modulees

Company Models Implementation

```
using System.ComponentModel.DataAnnotations;

namespace testLinked.Models
{
    public class company
    {
    }
}


namespace testLinked.Models
{
    public class Company
    {
        [Key]
        public int Id { get; set; }

        [Required, MaxLength(100)]
        public string Name { get; set; } // Company name (e.g., "VOIS-Code")

        [Required, MaxLength(100)]
        public string Location { get; set; } // City, Country

        [Required, MaxLength(50)]
        public string Category { get; set; } // Industry category (e.g., Software)

        public string Description { get; set; } // About the company

        public string CompanyLogo { get; set; } // Path to company logo image

        [MaxLength(20)]
        public string Phone { get; set; }

        [EmailAddress]
        public string Email { get; set; }

        public string Website { get; set; } // Company's website

        public DateTime Founded { get; set; } // Year company was founded

        public int EmployeeCount { get; set; } // Number of employees

        public bool IsHiring { get; set; } // Whether the company is hiring

        // Relationship: One company can have many jobs
        public List<Job> Jobs { get; set; } = new();
    }
}

```

Forget-Password Models Implementation



```
using System.ComponentModel.DataAnnotations;

namespace testLinked.Models
{
    public class ForgotPasswordRequest
    {
        [Required]
        [EmailAddress]
        public string Email { get; set; }
    }
}

```

Interview Models Implementation

```
using System.ComponentModel.DataAnnotations;

namespace testLinked.Models
{
    public class InterviewQuestion
    {
        [Key]
        public int Id { get; set; }

        [Required, MaxLength(500)]
        public string Question { get; set; } // The interview question

        [Required, MaxLength(500)]
        public string Answer { get; set; } // Suggested answer

        [MaxLength(500)]
        public string Tip { get; set; } // Extra explanation for answering


    }
}

```

Jobs Models Implementation

```
using System.ComponentModel.DataAnnotations;

namespace testLinked.Models
        {
            public class Job
            {
                [Key]
                public int Id { get; set; }

                [Required, MaxLength(100)]
                public string Title { get; set; }

                [Required, MaxLength(100)]
                public string Company { get; set; }

                [Required, MaxLength(100)]
                public string Location { get; set; }

                [Required, MaxLength(50)]
                public string Type { get; set; } // e.g., Full time, Part time

                public DateTime Posted { get; set; } = DateTime.UtcNow;

                [Required]
                public string Description { get; set; }

                public string JobDescriptionInfo { get; set; }

                [MaxLength(50)]
                public string Salary { get; set; }

                public List<string> About { get; set; } = new(); // Stores multiple about descriptions

                public List<string> Requirements { get; set; } = new(); // List of requirements

                public List<string> PreferredSkills { get; set; } = new(); // List of preferred skills

                public string ProductDesigner { get; set; } // Additional job description

                public DateTime ExpirationDate { get; set; }

                [MaxLength(20)]
                public string Phone { get; set; }

                [EmailAddress]
                public string Email { get; set; }

                public string CompanyLogo { get; set; } // Path to company logo image

                public bool Urgent { get; set; }
                public bool Senior { get; set; }
                public bool FullTime { get; set; }
            }
        }

```

Profile Models Implementation

```
using System;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace testLinked.Models
{
    public class Profile
    {
        [Key]
        public int Id { get; set; }

        [Required]
        public int UserId { get; set; } // FK to User

        public string FirstName { get; set; }
        public string LastName { get; set; }
        public string Location { get; set; }
        public string Phone { get; set; }
        public string ProfessionalHeadline { get; set; }
        public string ShortBio { get; set; }

        public string CV { get; set; } // Path or URL to uploaded CV
        public string ProfilePicture { get; set; } // Path or URL to uploaded profile image

        public bool OpenToRemote { get; set; }
        public bool OpenToRelocation { get; set; }

        public string LinkedIn { get; set; }
        public string Twitter { get; set; }
        public string Website { get; set; }
        public string Video { get; set; }

        public string AdditionalSkills { get; set; }

        public DateTime DateCreated { get; set; } = DateTime.UtcNow;

        [ForeignKey("UserId")]
        public User User { get; set; }
    }
}
```

JobApplication Models Implementation

```
using System;
using System.ComponentModel.DataAnnotations;

namespace testLinked.Models
{
    public class JobApplication
    {
        [Key]
        public int Id { get; set; }

        [Required, MaxLength(100)]
        public string FullName { get; set; }

        [Required, EmailAddress]
        public string Email { get; set; }

        [Required, MaxLength(20)]
        public string ContactNumber { get; set; }

        public string Description { get; set; }

        public string ResumePath { get; set; } // Path to uploaded resume file

        public bool AgreedToTerms { get; set; }

        public DateTime AppliedAt { get; set; } = DateTime.UtcNow;

        // Foreign key reference to the job
        public int JobId { get; set; }
        public Job Job { get; set; }
    }
}

```

User Models Implementation

```
using System.ComponentModel.DataAnnotations;

namespace testLinked.Models
{
    public class User
    {
        [Key]
        public int Id { get; set; }

        [Required]
        public string Username { get; set; }

        [Required]
        [EmailAddress]
        public string Email { get; set; }

        [Required]
        public string Password { get; set; }

        public string Role { get; set; } = "User"; // Default role
    }
}

```

6.2.3 Folder Structure Example

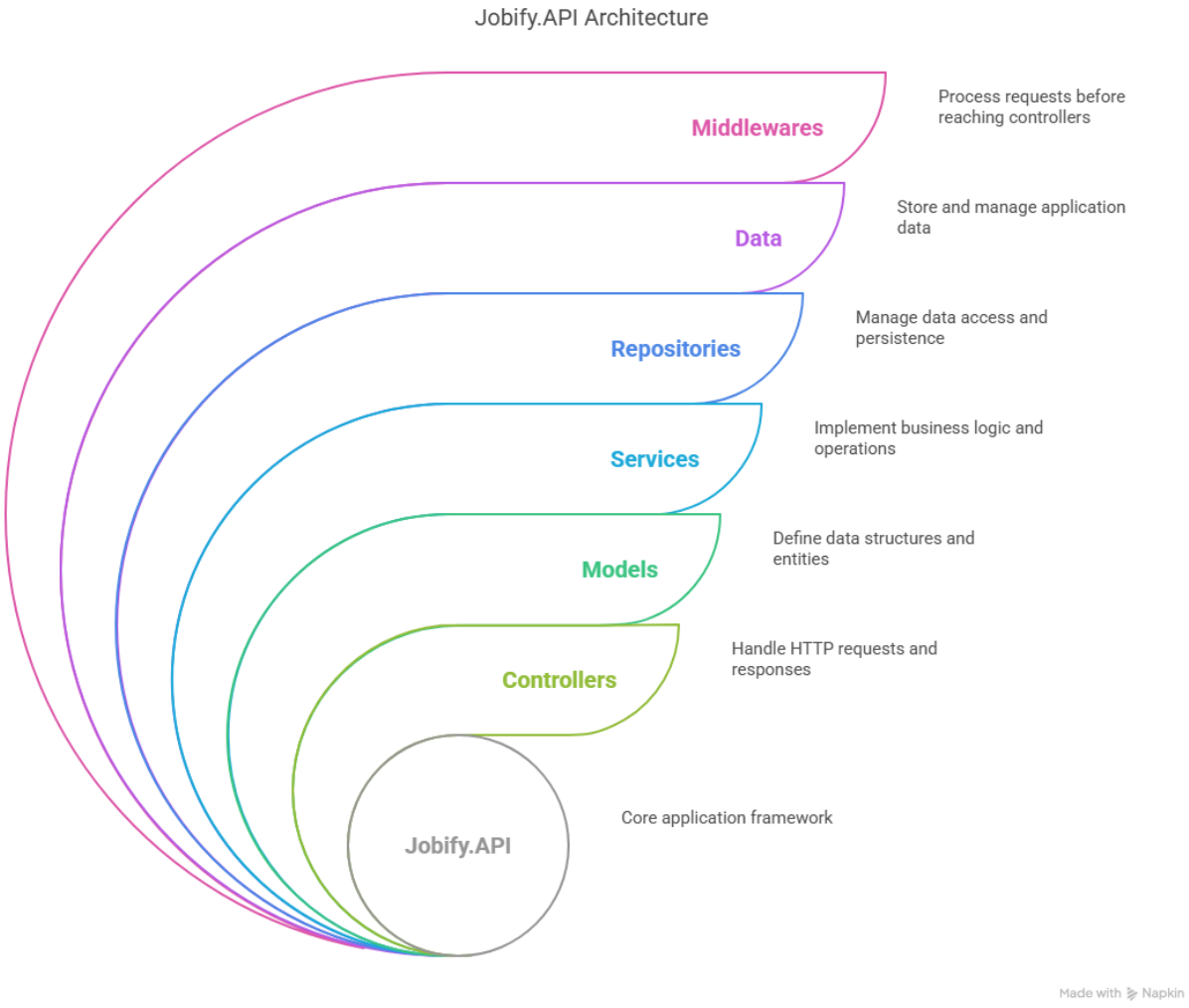


Figure 6.2.3 Folder Structure

6.2.4 Authentication & Security

- **JWT Authentication**: Every user receives a token upon login, attached to each API request.
- **Role-based Authorization**: Admin, Employer, and Job Seeker roles control access to different endpoints.
- **Password Hashing**: Uses ASP.NET Identity to hash and securely store user credentials.
- **CORS Policy**: Configured to allow secure communication between Angular and .NET servers.

```
using Microsoft.AspNetCore.Authentication.JwtBearer;
using Microsoft.EntityFrameworkCore;
using Microsoft.IdentityModel.Tokens;
using System.Text;
using testLinked.Data; // Ensure this namespace is correctly imported

var builder = WebApplication.CreateBuilder(args);

// 🔹 Add Database Context
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

// 🔹 Authentication & Authorization
builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
    .AddJwtBearer(options =>
    {
        options.TokenValidationParameters = new TokenValidationParameters
        {
            ValidateIssuer = true,
            ValidateAudience = true,
            ValidateLifetime = true,
            ValidateIssuerSigningKey = true,
            ValidIssuer = builder.Configuration["Jwt:Issuer"],
            ValidAudience = builder.Configuration["Jwt:Audience"],
            IssuerSigningKey = new SymmetricSecurityKey(
                Encoding.UTF8.GetBytes(builder.Configuration["Jwt:Key"]))
        };
    });

// 🔹 Add services to the container
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

// 🔹 Authorization
builder.Services.AddAuthorization(options =>
{
    options.AddPolicy("AdminPolicy", policy => policy.RequireRole("Admin"));
});

var app = builder.Build();

// 🔹 Middleware configuration
app.UseSwagger();
app.UseSwaggerUI();

app.UseHttpsRedirection(); // Ensures HTTPS
app.UseAuthentication();
app.UseAuthorization();

app.MapControllers();

app.Run();
```

6.2.5 API Development

- **RESTful APIs**: Follows REST architecture with HTTP verbs (GET, POST, PUT, DELETE).
- **Swagger**: Used for API documentation and testing.
- **AutoMapper**: Transforms entity models to DTOs for cleaner API responses.
- **Global Exception Handling**: Middleware catches errors and returns proper status codes/messages.

```
<!DOCTYPE html>
<!-- saved from url=(0042)https://localhost:44327/swagger/index.html -->
<html lang="en"><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    <title>Swagger UI</title>
    <link rel="stylesheet" type="text/css" href="./Swagger UI_files/swagger-ui.css">
    <link rel="icon" type="image/png" href="https://localhost:44327/swagger/favicon-32x32.png" sizes="32x32">
    <link rel="icon" type="image/png" href="https://localhost:44327/swagger/favicon-16x16.png" sizes="16x16">
    <style>

        html {
            box-sizing: border-box;
            overflow: -moz-scrollbars-vertical;
            overflow-y: scroll;
        }

        *,
        *:before,
        *:after {
            box-sizing: inherit;
        }

        body {
            margin: 0;
            background: #fafafa;
        }
    </style>

</head>
```

```
<body>

    <!-- Workaround for https://github.com/swagger-api/swagger-editor/issues/1371 -->
    <script>
        if (window.navigator.userAgent.indexOf("Edge") > -1) {
            console.log("Removing native Edge fetch in favor of swagger-ui's polyfill")
            window.fetch = undefined;
        }
    </script>

    <script src="./Swagger UI_files/swagger-ui-bundle.js.download"></script>
    <script src="./Swagger UI_files/swagger-ui-standalone-preset.js.download"></script>
    <script>
        /* Source: https://gist.github.com/lamberta/3768814
         * Parse a string function definition and return a function object. Does not use eval.
         * @param {string} str
         * @return {function}
         *
         * Example:
         *  var f = function (x, y) { return x * y; };
         *  var g = parseFunction(f.toString());
         *  g(33, 3); //=> 99
         */
        function parseFunction(str) {
            if (!str) return void (0);

            var fn_body_idx = str.indexOf('{'),
                fn_body = str.substring(fn_body_idx + 1, str.lastIndexOf('}')),
                fn_declare = str.substring(0, fn_body_idx),
                fn_params = fn_declare.substring(fn_declare.indexOf('(') + 1, fn_declare.lastIndexOf(')')),
                args = fn_params.split(',');

            args.push(fn_body);

            function Fn() {
                return Function.apply(this, args);
            }
            Fn.prototype = Function.prototype;

            return new Fn();
        }

        window.onload = function () {

            // Workaround for https://github.com/swagger-api/swagger-ui/issues/5945
            configObject.urls.forEach(function (item) {
                if (item.url.startsWith("http") || item.url.startsWith("/")) return;
                item.url = window.location.href.replace("index.html", item.url).split('#')[0];
            });

            // If validatorUrl is not explicitly provided, disable the feature by setting to null
            if (!configObject.hasOwnProperty("validatorUrl"))
                configObject.validatorUrl = null

            // If oauth2RedirectUrl isn't specified, use the built-in default
            if (!configObject.hasOwnProperty("oauth2RedirectUrl"))
                configObject.oauth2RedirectUrl = (new URL("oauth2-redirect.html", window.location.href)).href;

            // Apply mandatory parameters
            configObject.dom_id = "#swagger-ui";
            configObject.presets = [SwaggerUIBundle.presets.apis, SwaggerUIStandalonePreset];
            configObject.layout = "StandaloneLayout";

            // Parse and add interceptor functions
            var interceptors = JSON.parse('{"RequestInterceptorFunction":null,"ResponseInterceptorFunction":null}');
            if (interceptors.RequestInterceptorFunction)
                configObject.requestInterceptor = parseFunction(interceptors.RequestInterceptorFunction);
            if (interceptors.ResponseInterceptorFunction)
                configObject.responseInterceptor = parseFunction(interceptors.ResponseInterceptorFunction);

            // Begin Swagger UI call region

            const ui = SwaggerUIBundle(configObject);

            ui.initOAuth(oauthConfigObject);

            // End Swagger UI call region

            window.ui = ui
        }
    </script>

<!-- Visual Studio Browser Link -->
<!-- End Browser Link -->
<script src="./Swagger UI_files/aspnetcore-browser-refresh.js.download"></script>

</body></html>
```

6.2.6 Development Tools

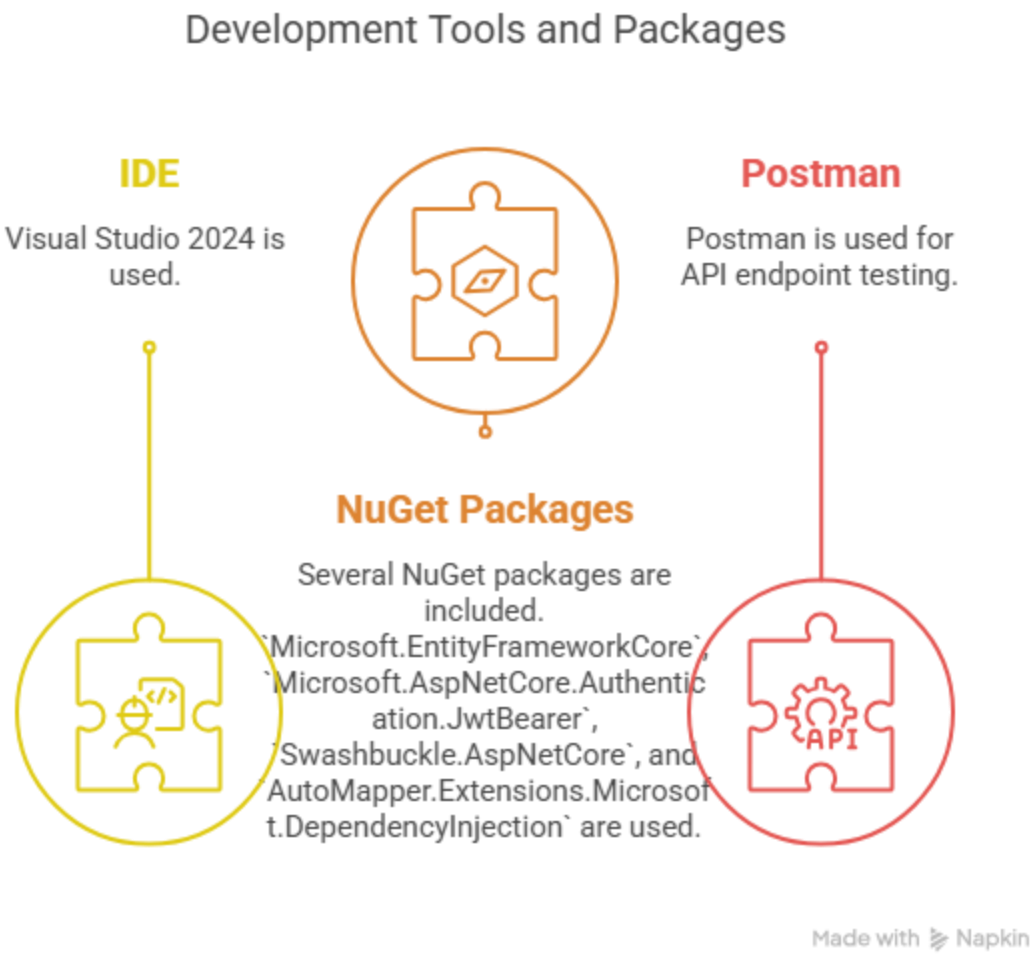


Figure 6.2.6 Development Tools

6.2.7 Key APIs Overview

Jobify API Endpoint Table:

Method	Endpoint	Description
POST	`/api/Auth/register`	Register a new user
POST	`/api/Auth/login`	User login and JWT generation
POST	`/api/Password/forgot`	Send password reset link to user's email

Jobs :

Method	Endpoint	Description
GET	`/api/Jobs`	Get all job listings
POST	`/api/Jobs`	Add a new job (employer only)
GET	`/api/Jobs/{id}`	Get job details by ID
PUT	`/api/Jobs/{id}`	Update a job posting
DELETE	`/api/Jobs/{id}`	Delete a job listing
POST	`/api/apply/{jobId}`	Submit job application

Companies :

Method	Endpoint	Description
GET	`/api/Companies`	Retrieve all companies
POST	`/api/Companies`	Add a new company
GET	`/api/Companies/{id}`	Get a company by ID

Method	Endpoint	Description
PUT	`/api/Companies/{id}`	Update company details
DELETE	`/api/Companies/{id}`	Delete a company

Interview Questions :

Method	Endpoint	Description
GET	`/api/InterviewQuestions`	Get all interview questions
POST	`/api/InterviewQuestions`	Add a new interview question
GET	`/api/InterviewQuestions/{id}`	Get an interview question by ID
PUT	`/api/InterviewQuestions/{id}`	Update interview question
DELETE	`/api/InterviewQuestions/{id}`	Delete interview question

Job Applications :

Method	Endpoint	Description
GET	`/api/JobApplications`	Get all job applications
POST	`/api/JobApplications`	Submit a job application
GET	`/api/JobApplications/{id}`	Get job application details
DELETE	`/api/JobApplications/{id}`	Delete a job application

Profile :

Method	Endpoint	Description
GET	`/api/Profile`	Get all user profiles
POST	`/api/Profile`	Create a new user profile
GET	`/api/Profile/{id}`	Get profile by ID
PUT	`/api/Profile/{id}`	Update a user profile
DELETE	`/api/Profile/{id}`	Delete a user profile

AI-Powered Services

Method	Endpoint	Description
POST	`/api/cv/analyze`	Send CV to AI model and return feedback
POST	`/api/chat`	Send message to chatbot

6.3 Database (SQL Server)

Overview

Jobify uses Microsoft SQL Server as the primary relational database management system. SQL Server was chosen for its robustness, support for stored procedures, advanced querying capabilities, and strong integration with .NET. This allows for efficient, secure, and scalable management of structured data such as user information, job listings, applications, interview questions, and company details.

Key Implementation Components :

Component	Purpose
User Table	Stores information about users (name, email, password hash, role)
Jobs Table	Holds job listings including title, description, company, location, salary, etc.
Companies Table	Contains data about companies posting jobs
Applications Table	Tracks user applications for specific jobs
Profiles Table	Stores extended user profile data (skills, education, experience)
Interview Questions Table	Contains common interview questions for various roles
Notifications Table	Logs job alerts and platform notifications
CV Feedback Table	Stores AI feedback results from resume analysis

Connection String to DataBase (SQL) :

```
{
    "Logging": {
        "LogLevel": {
            "Default": "Information",
            "Microsoft.AspNetCore": "Warning"
        }
    },
    "AllowedHosts": "*",
    "ConnectionStrings": {
        "DefaultConnection": "Server=SOHILA\\SOHILA;Database=testLinked;Trusted_Connection=True;Encrypt=False;"
    },
    "Jwt": {
        "Key": "SuperSecretKey12345",
        "Issuer": "testLinked",
        "Audience": "testLinked"
    }
}
```

Security Features

- Passwords are stored using hashed & salted values.
- Role-based access ensures that only admins or employers can post/edit/delete jobs.
- SQL Injection protection through parameterized queries and ORM (Entity Framework Core).

Tooling and Environment

- **SQL Server Management Studio (SSMS)**: Used for database design, query testing, and performance tuning.
- **Entity Framework Core**: Integrated with .NET backend for seamless CRUD operations using strongly typed models.
- **Migrations**: Database schema changes are managed using EF Core Migrations for consistency across development and production environments.

All Pages and Their Code Structure IN Repo on Git-Hub Link In Reference

6.4 AI-Powered Features Implementation

Jobify incorporates Artificial Intelligence (AI) and Machine Learning (ML) to enhance user experience through intelligent automation and data-driven decision-making. The two core AI-powered features currently implemented are the Resume Analyzer and the Chatbot Assistant.

1. Resume Analyzer

Purpose

This tool evaluates uploaded CVs to determine how well they align with job listings, providing feedback on structure, skills, and keywords.

Technology Stack

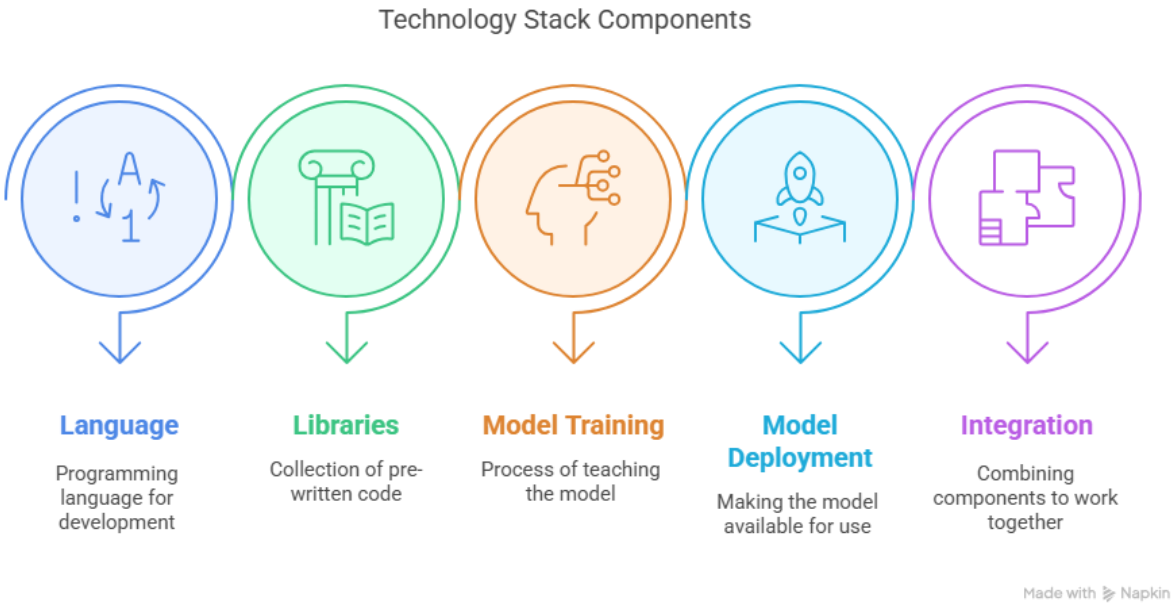


Figure 6.4 Ai Powered Technology Stack

Workflow

1. **User uploads CV (PDF or text format)** through the frontend.
2. **Backend (.NET)** sends the CV file to a Python-based Flask API.
3. **NLP Model (spaCy + Transformers)** analyzes the CV:
    - Extracts key sections (Skills, Experience, Education)
    - Identifies missing or weak areas based on job trends
4. The **AI model returns structured feedback** including:
    - Skill match score
    - Recommendations for improvement
    - Keyword optimization suggestions
5. Feedback is displayed to the user in a structured, user-friendly interface.

Development Environment

- Google Colab was used to train and fine-tune NLP models using labeled CV/job datasets.
- Flask REST APIs were hosted locally and called from the .NET backend.
- Deployment-ready model was containerized for future scaling using Docker.

Comprehensive Comparison Report of Three Resume Processing Models :

1. Overview

This document provides a detailed comparison between three different resume-processing Python models:

- Model 1: Extracted Info & Hiring Process
- Model 2: Resume Categorization Prediction
- Model 3: Resume Job Recommendation System

Each model has a specific role in a larger recruitment system, focusing respectively on data extraction, resume classification, and job recommendation.

2. Functional Purpose

- Model 1 - Extracted Info & Hiring Process:

- Extracts structured data such as name, email, phone, education, and skills using regular
- expressions and keyword matching.
- Key functions include extract_contact_number_from_resume, extract_email_from_resume
- extract_skills_from_resume, etc.

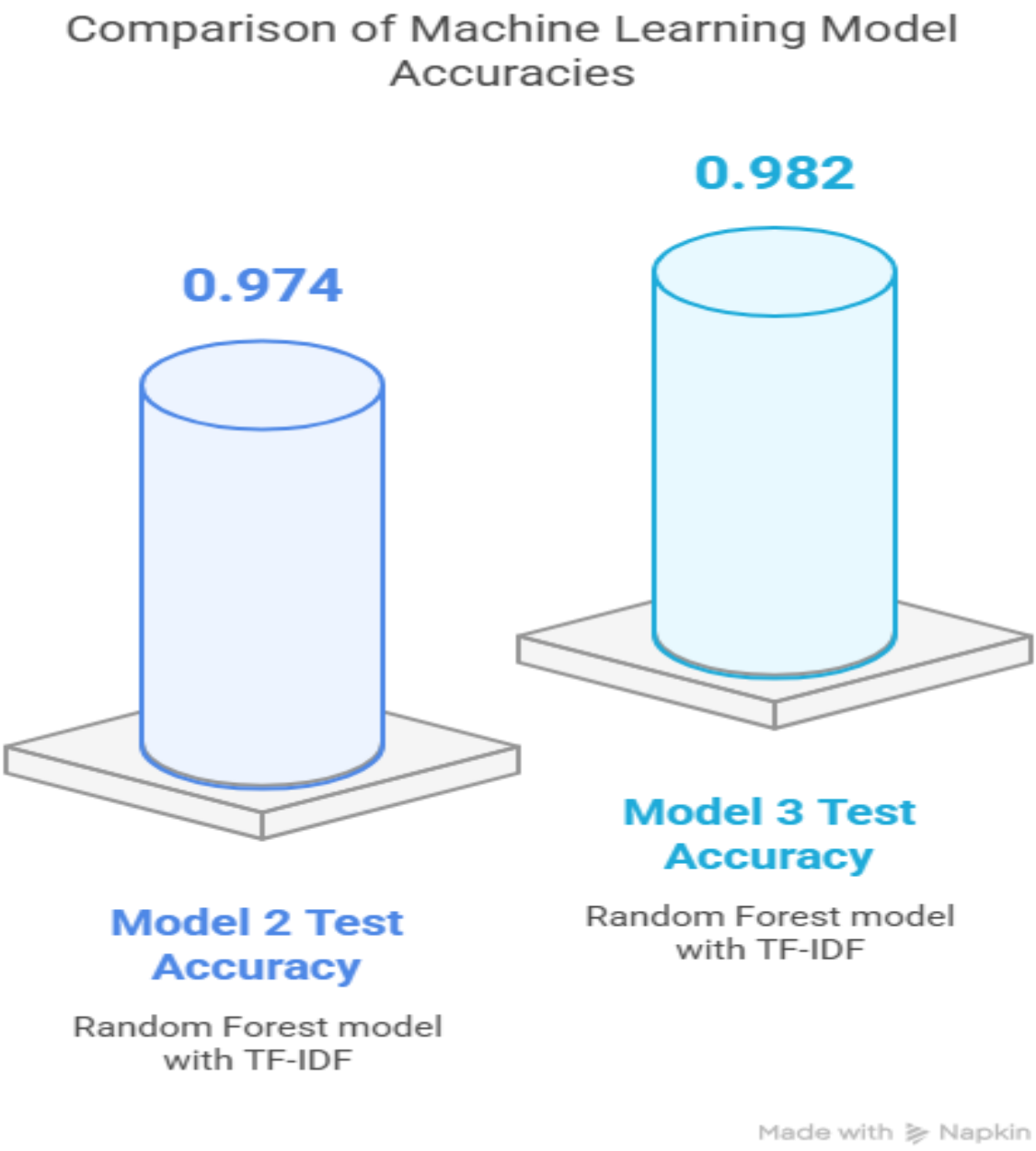
- Model 2 - Resume Categorization Prediction:

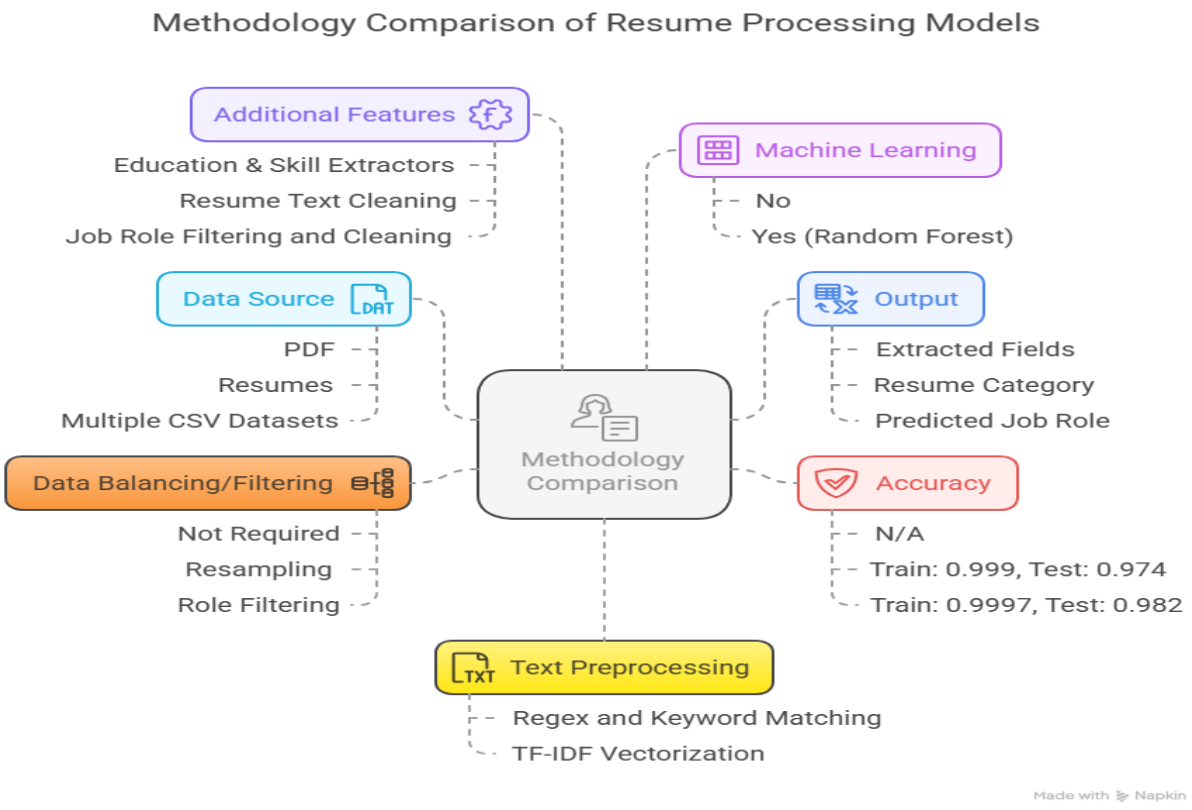
- Classifies resumes into job categories using TF-IDF and Random Forest.
- Trained on balanced datasets with resampling techniques.
- Accuracy: Train - 0.999, Test - 0.974

- Model 3 - Resume Job Recommendation System:

- Predicts suitable job roles based on resume content using Random Forest and TF-IDF.
- Only roles with more than 6500 samples are used.
- Accuracy: Train - 0.9997, Test - 0.982

4. Methodology Comparison :




Methodology Comparison of Resume Processing Models
Additional Features
Education & Skill Extractors
Resume Text Cleaning
Job Role Filtering and Cleaning
Machine Learning
No
Yes (Random Forest)
Data Source
DAT
PDF
Resumes
Multiple CSV Datasets
Output
Extracted Fields
Resume Category
Predicted Job Role
Methodology Comparison
Data Balancing/Filtering
Not Required
Resampling
Role Filtering
Accuracy
N/A
Train: 0.999, Test: 0.974
Train: 0.9997, Test: 0.982
Text Preprocessing
TXT
Regex and Keyword Matching
TF-IDF Vectorization
Made with Napkin

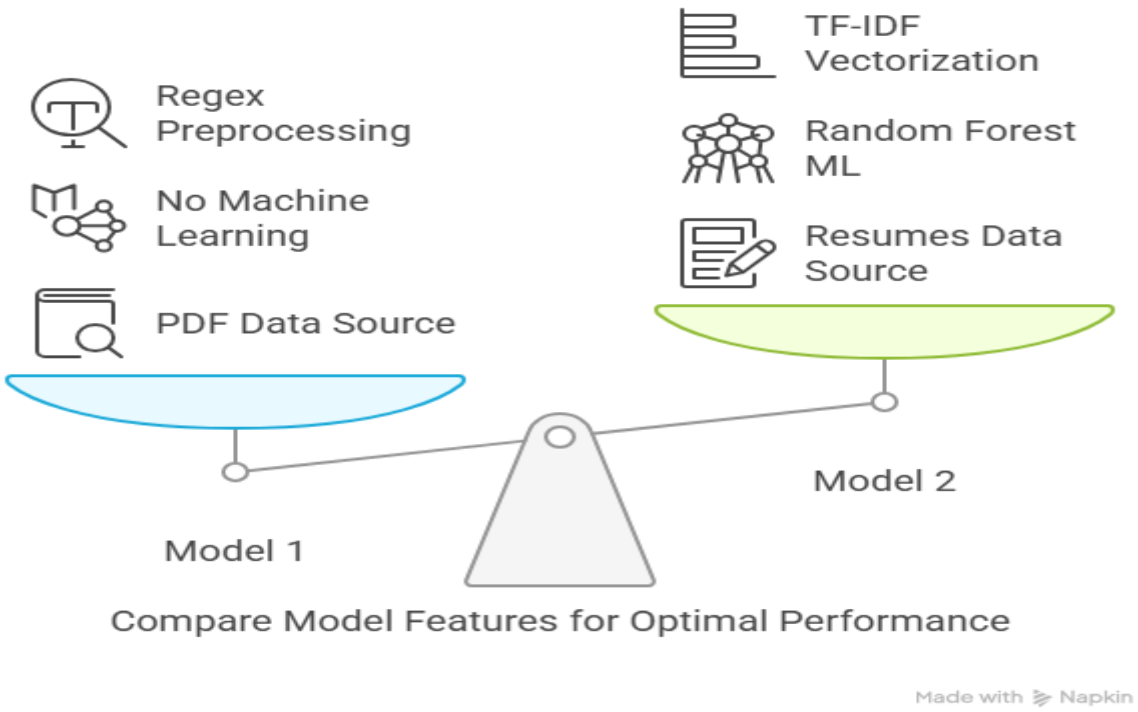


Figure 6.4 Models Comparison (CV)

2. Chatbot Assistant

Purpose

The AI-driven chatbot provides users with instant answers to common questions, job navigation help, and basic career advice.

Workflow

1. User types a query in the chatbot window (e.g., “Show me jobs in IT”).
2. Angular sends the message to the chatbot backend endpoint.
3. Backend forwards the request to the NLP engine (Rasa).
4. Rasa processes intent and extracts entities.
5. Based on intent, the response is returned:
    - Answer to a question
    - A redirect or action (e.g., open job search page)
    - Suggestion (e.g., improve your resume)
6. Response is shown instantly in chat interface.

Comprehensive Comparison Report of ChatBot Models :

1. Key ObservaƟons

A. Training Accuracy

- Random Forest (100%) achieves perfect training accuracy, indicating it memorized the
- training data (potenal overfiƫng).
- AdaBoost (95.8%) performs well but generalizes beer than Random Forest.
- Logistic Regression (92.5%) has the lowest training accuracy but shows good

B. Test Accuracy (Generalization)

- Random Forest (91.7%) sll leads but shows a significant drop from training (overfiƫng
- risk).
- AdaBoost (90.1%) maintains strong performance with less overfiƫng.

2. Model Performance Comparison :

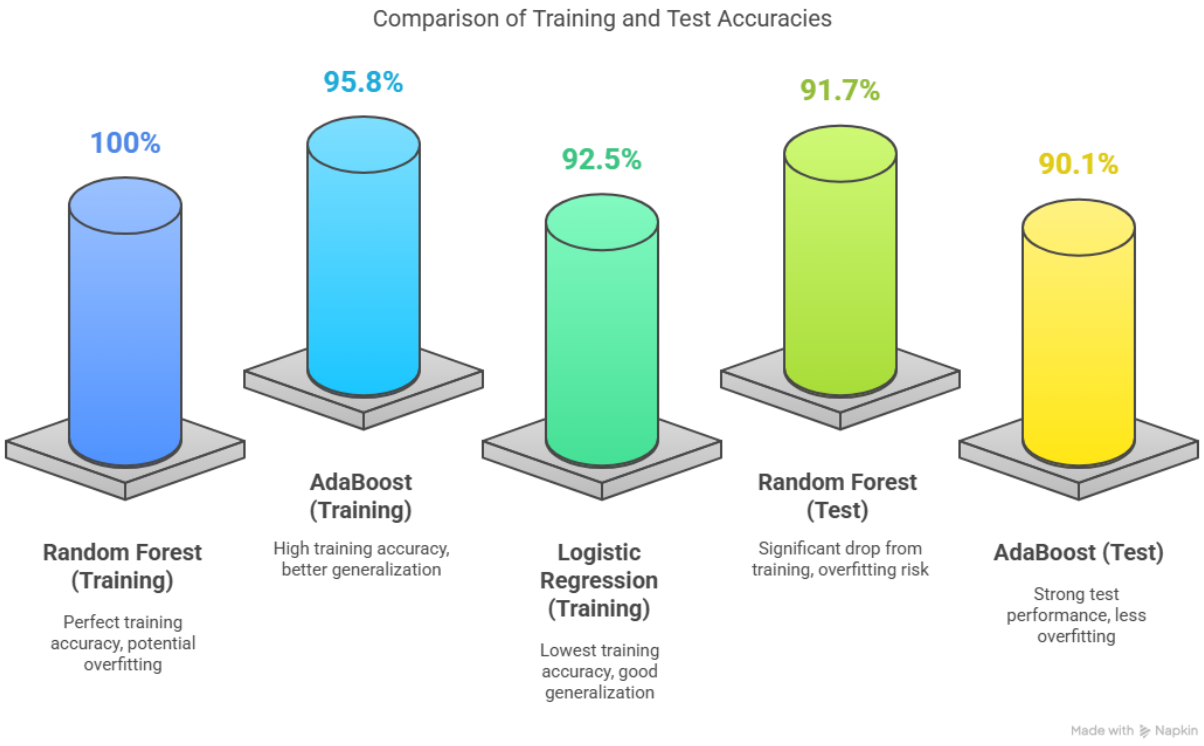


Figure 6.4 Models Comparison (ChatBot)

Summary of AI Feature Impact

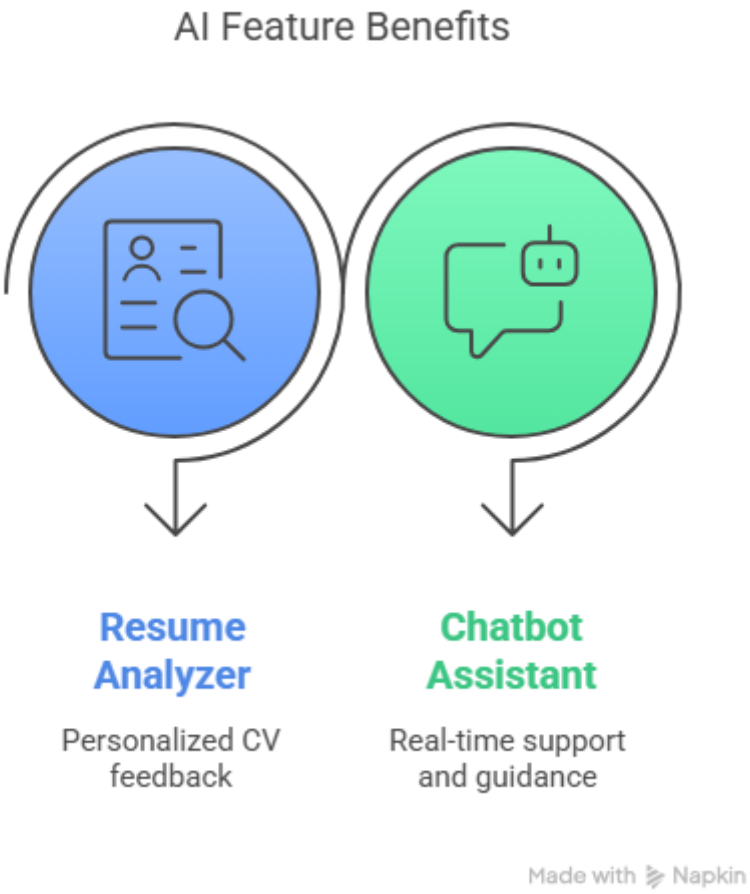


Figure 6.4 AI Features Benefits

Chapter Seven
UI / UX Design

UI/UX Design

User Interface (UI) and User Experience (UX) design play a critical role in the overall success of any web-based application, and for a career platform like Jobify, it's even more essential. Jobify's target audience—fresh graduates, students, and early-career professionals—often feel overwhelmed by traditional job portals due to cluttered interfaces, confusing workflows, and generic interactions. The goal of this chapter is to explore how design decisions were made to solve these pain points and ensure a delightful, stress-free journey for the user.

The UI/UX of Jobify was crafted with a user-first mentality, focusing on clarity, usability, consistency, and accessibility. The visual and interactive components were developed not just to make the platform look appealing, but also to ensure that users could achieve their goals with minimal effort—whether that's applying for a job, preparing for an interview, or analyzing their resume.

To achieve this, the design process adopted a human-centered design approach that was:

- Empathy-driven: Prioritizing user needs and behaviors.
- Iterative: Continuously tested and refined based on feedback.
- Collaborative: Ensuring seamless communication between designers, developers, and stakeholders.
- Accessible: Following inclusive design principles to accommodate users with different abilities.

By leveraging modern frameworks such as Angular and Bootstrap, and tools like Figma for design prototyping, Jobify's frontend was not only technically strong but also visually elegant and functionally powerful. This chapter provides an in-depth walkthrough of how the design process was carried out, supported by wireframes, prototypes, and real user feedback.

7.1 Design Process

The design process followed in Jobify was iterative, human-centered, and feedback-driven, comprising the following key stages:

1. User Research

- **Target Audience**: Fresh graduates, students, early-career professionals.
- **Research Methods**: Surveys, interviews, and analysis of competitors (e.g., LinkedIn, Glassdoor).
- **Insights Gathered**:
    - Users want quick access to jobs that match their skills.
    - They often feel lost navigating complex job portals.
    - Many lack guidance in preparing for interviews or writing CVs.

2. User Personas & Journeys

- Created personas like "Ahmed the Fresh Graduate" and "Sara the Intern Seeker."
- Mapped their journey through the platform—from signup to landing a job.

3. Information Architecture

- Defined a logical structure for navigation:
    - Home
    - Job Listings
    - CV Analyzer
    - Interview Prep
    - Profile
    - Chatbot Assistant
    - Company Directory
    - Notifications
    - Contact Us / Support

4. Wireframing & Sketching

- Low-fidelity wireframes were created using Figma to quickly prototype ideas.
- Multiple layout options were evaluated for responsiveness and usability.

5. High-Fidelity Prototypes

- Created in Figma using Jobify's color palette, typography, and spacing guidelines.
- Integrated UI kits based on Bootstrap 5 components for real implementation alignment.

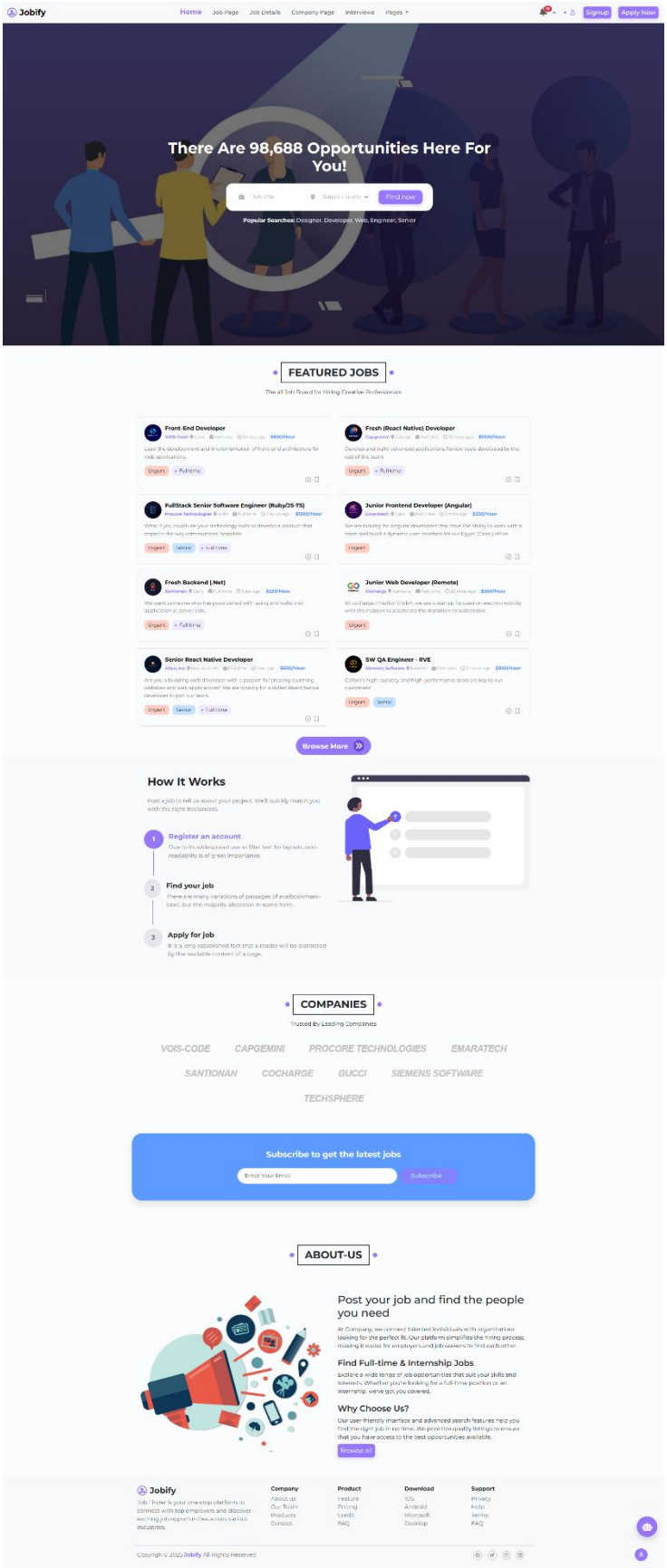
6. Usability Testing

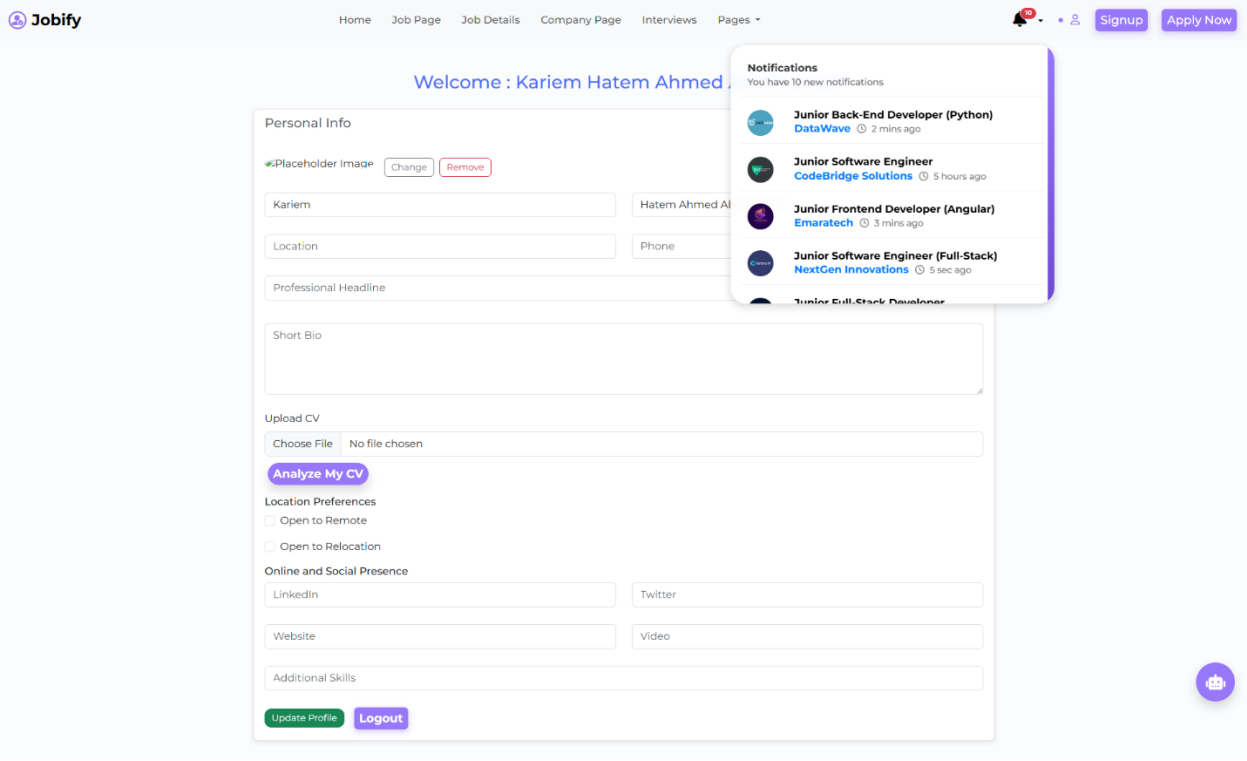
- Conducted peer reviews and A/B testing of different design versions.
- Iterated based on:
    - Ease of navigation
    - Clarity of call-to-actions
    - Form usability
    - Feedback visibility (e.g., CV analyzer results)

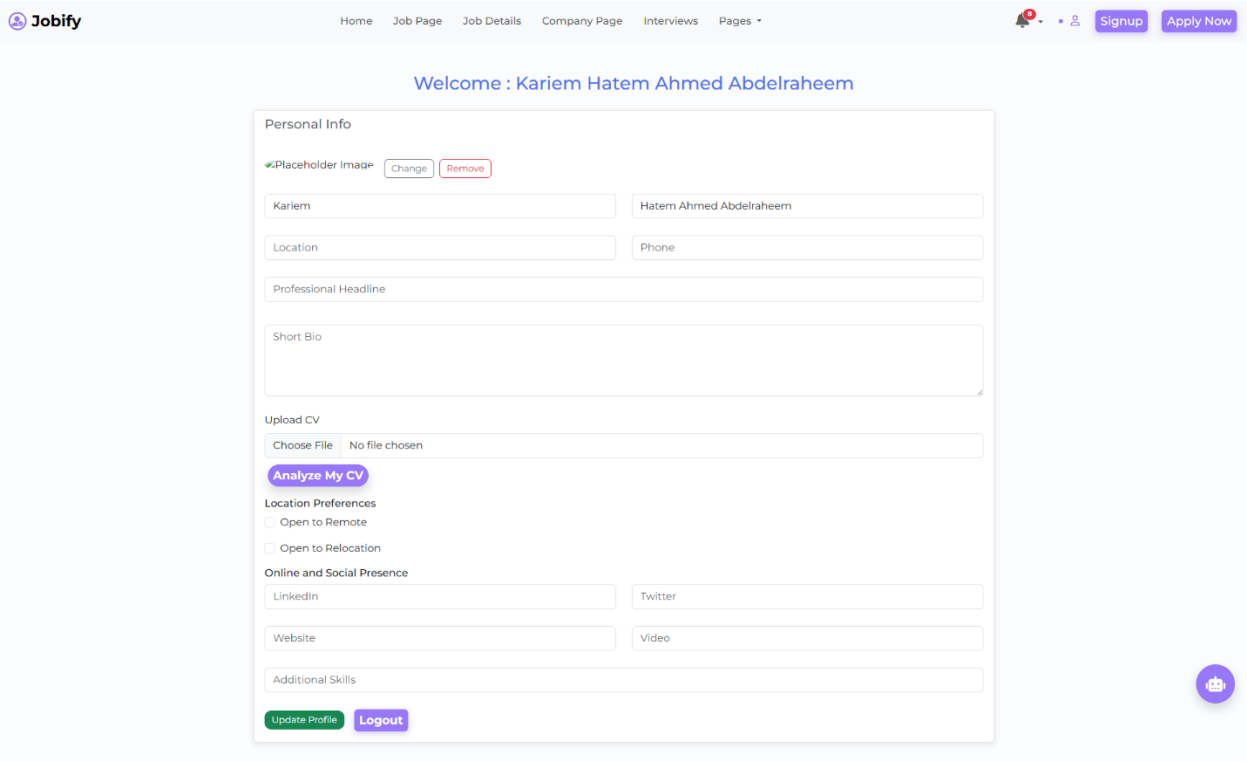
7. Implementation Sync

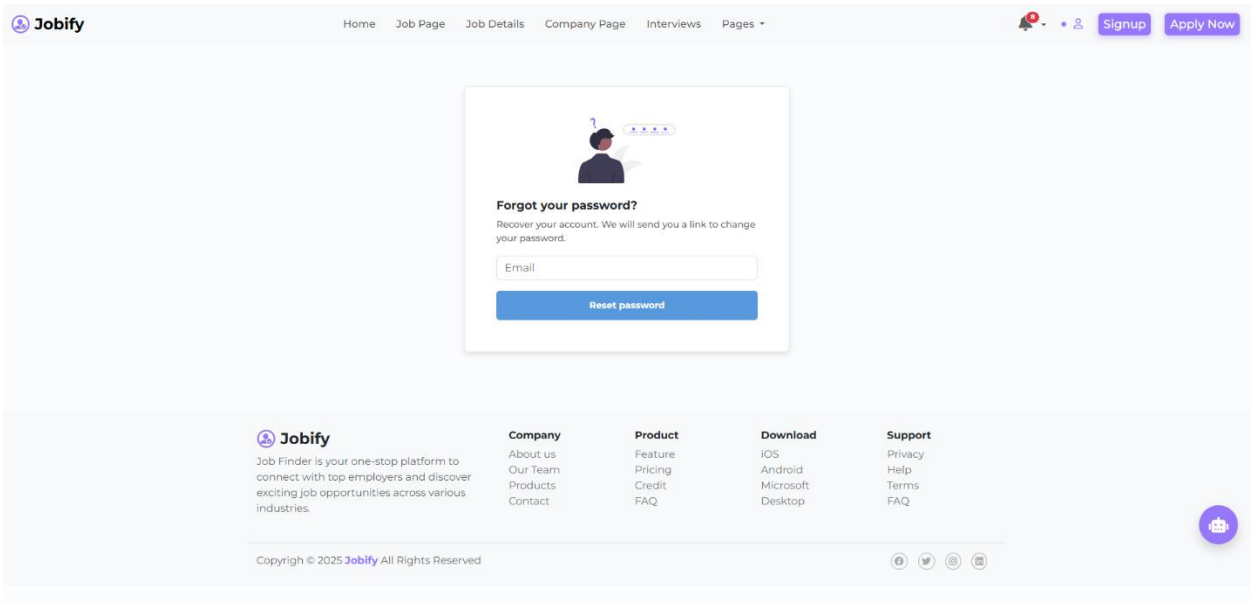
- Developers and designers worked in sync using shared Figma links and design tokens.
- Bootstrap was used in Angular to implement the designs pixel-perfectly.

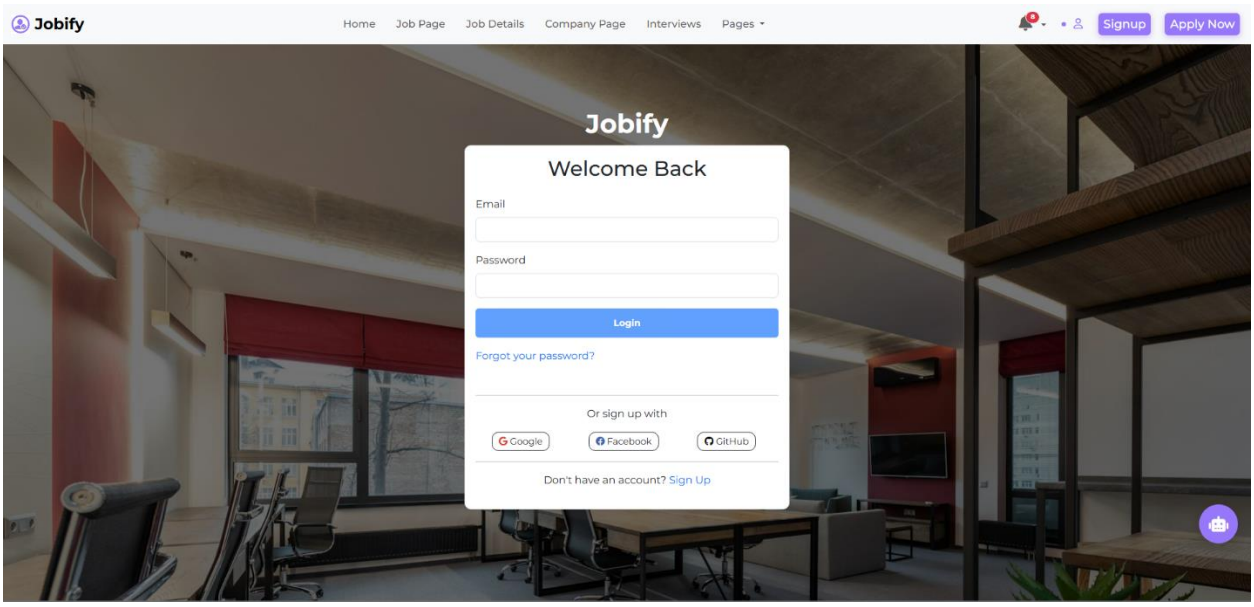
UI Design And Pages Design :

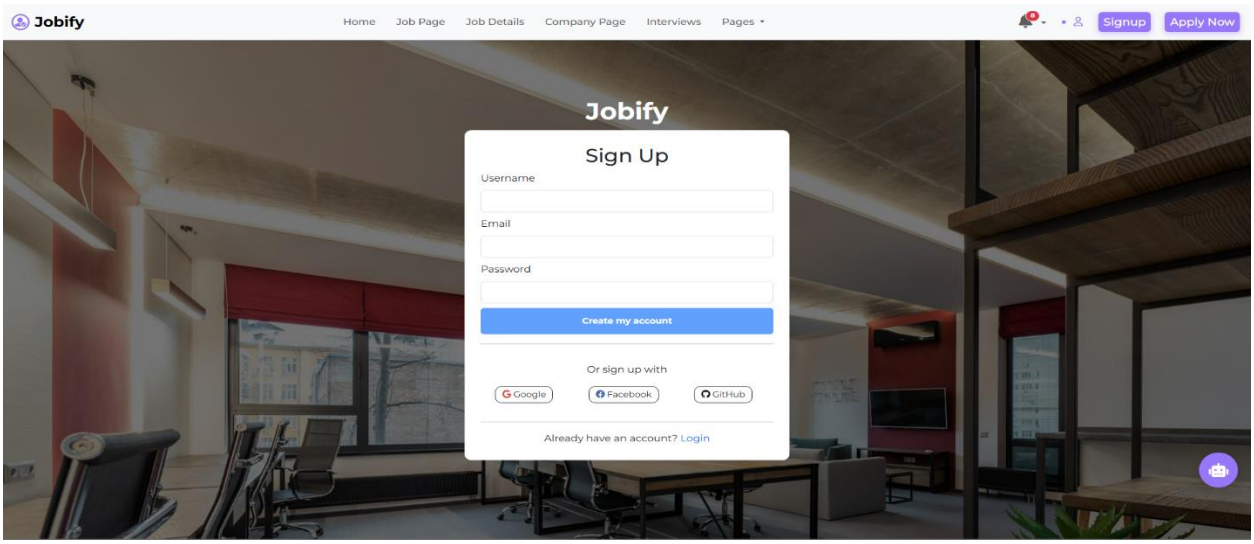

Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
There Are 98,688 Opportunities Here For You!
Find now
Popular Searches: Designer, Developer, Web, Engineer, Senior
FEATURED JOBS
Front-End Developer
Fresh (React Native) Developer
FullStack Senior Software Engineer (Ruby/JS-TS)
Junior Frontend Developer (Angular)
Fresh Backend (.Net)
Junior Web Developer (Remote)
Senior React Native Developer
SW QA Engineer - RVE
Browse More
How It Works
Register an account
Find your job
Apply for job
COMPANIES
VOIS-CODE
CAPGEMINI
PROCORE TECHNOLOGIES
EMARATECH
SANTIONAN
COCHARGE
GUCCI
SIEMENS SOFTWARE
TECHSPHERE
Subscribe to get the latest jobs
Subscribe
ABOUT-US
Post your job and find the people you need
Find Full-time & Internship Jobs
Why Choose Us?
Browse All
Jobify
Company
Product
Download
Support

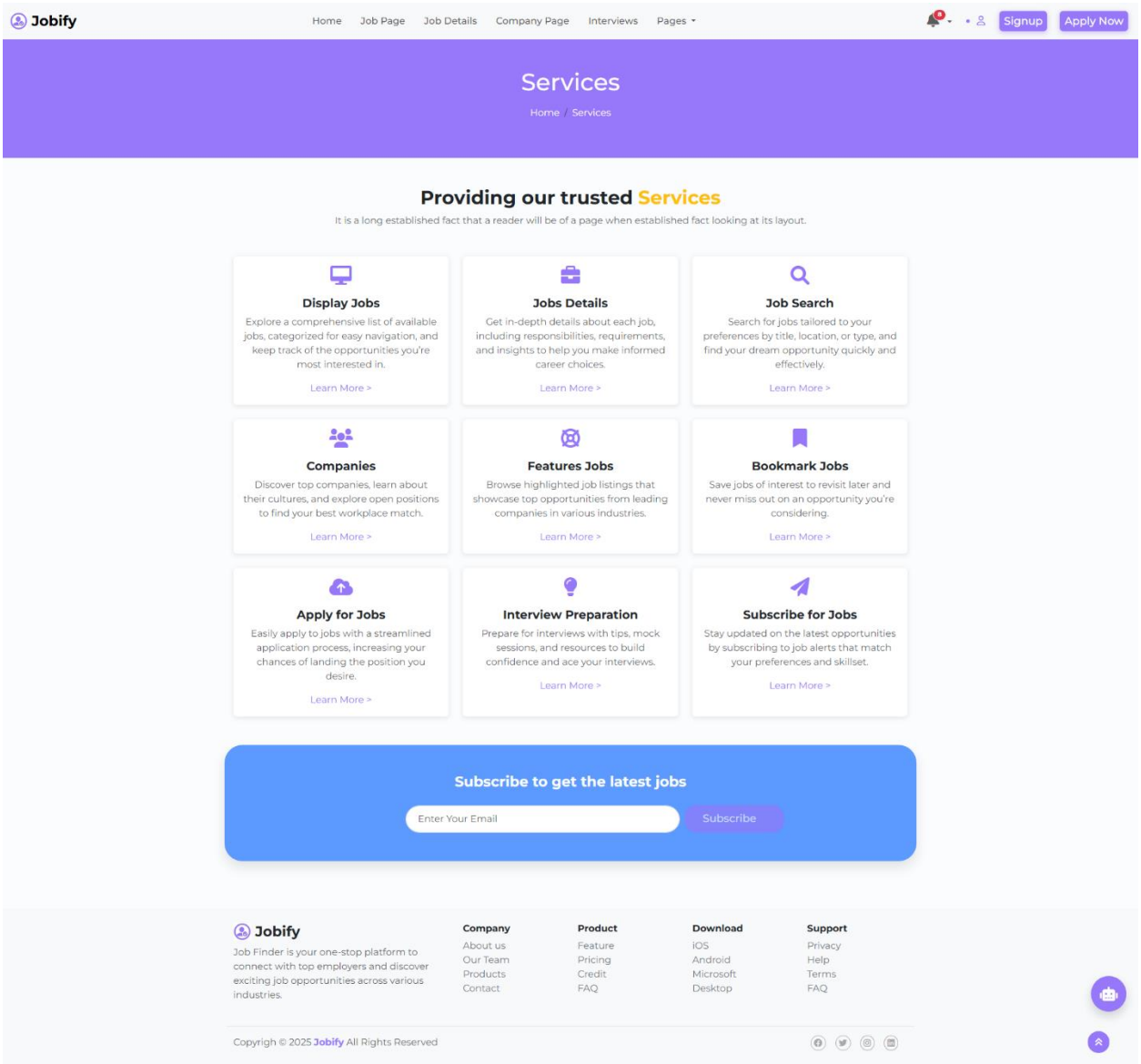

Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Welcome : Kariem Hatem Ahmed
Notifications
You have 10 new notifications
Junior Back-End Developer (Python)
DataWave 2 mins ago
Junior Software Engineer
CodeBridge Solutions 5 hours ago
Junior Frontend Developer (Angular)
Emaratech 3 mins ago
Junior Software Engineer (Full-Stack)
NextGen Innovations 5 sec ago
Personal Info
Placeholder Image
Change
Remove
Kariem
Location
Phone
Professional Headline
Short Bio
Upload CV
Choose File
No file chosen
Analyze My CV
Location Preferences
Open to Remote
Open to Relocation
Online and Social Presence
LinkedIn
Twitter
Website
Video
Additional Skills
Update Profile
Logout

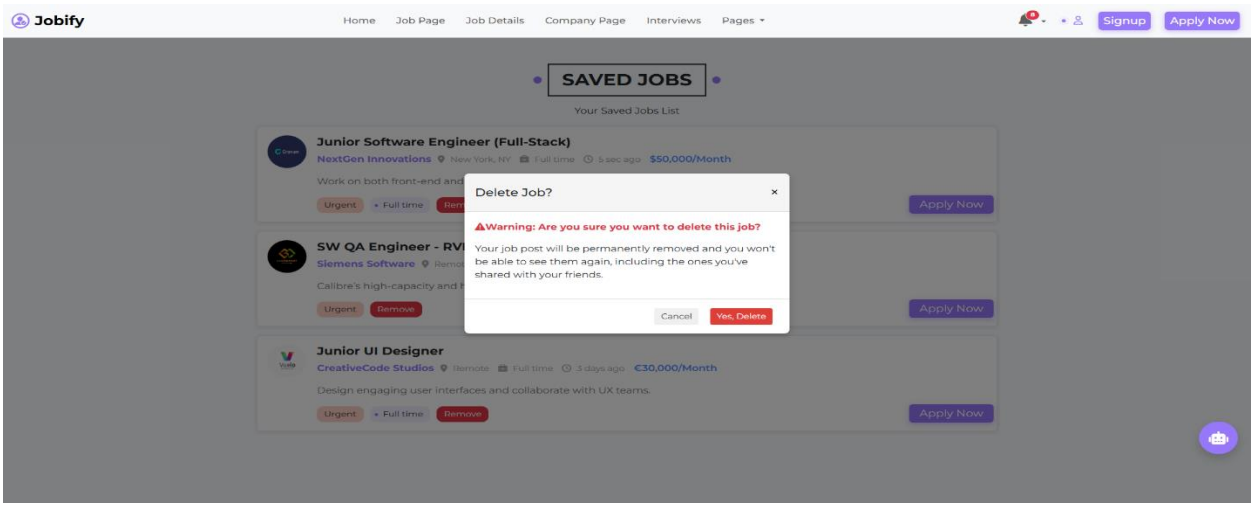

Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Welcome : Kariem Hatem Ahmed Abdelraheem
Personal Info
Placeholder Image
Change
Remove
Kariem
Hatem Ahmed Abdelraheem
Location
Phone
Professional Headline
Short Bio
Upload CV
Choose File
No file chosen
Analyze My CV
Location Preferences
Open to Remote
Open to Relocation
Online and Social Presence
LinkedIn
Twitter
Website
Video
Additional Skills
Update Profile
Logout

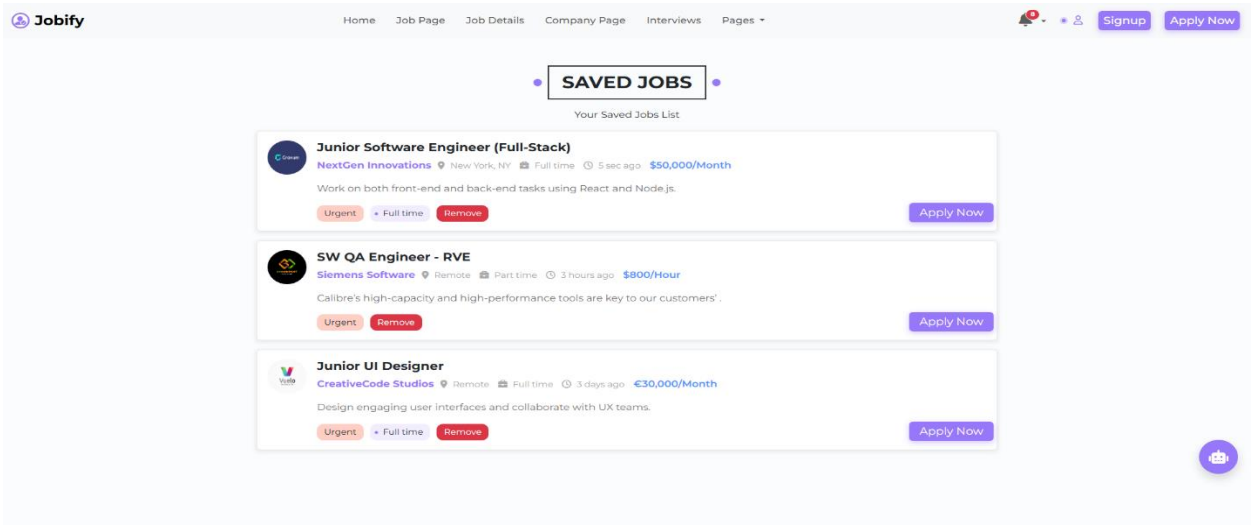

Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Forgot your password?
Recover your account. We will send you a link to change your password.
Email
Reset password
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved

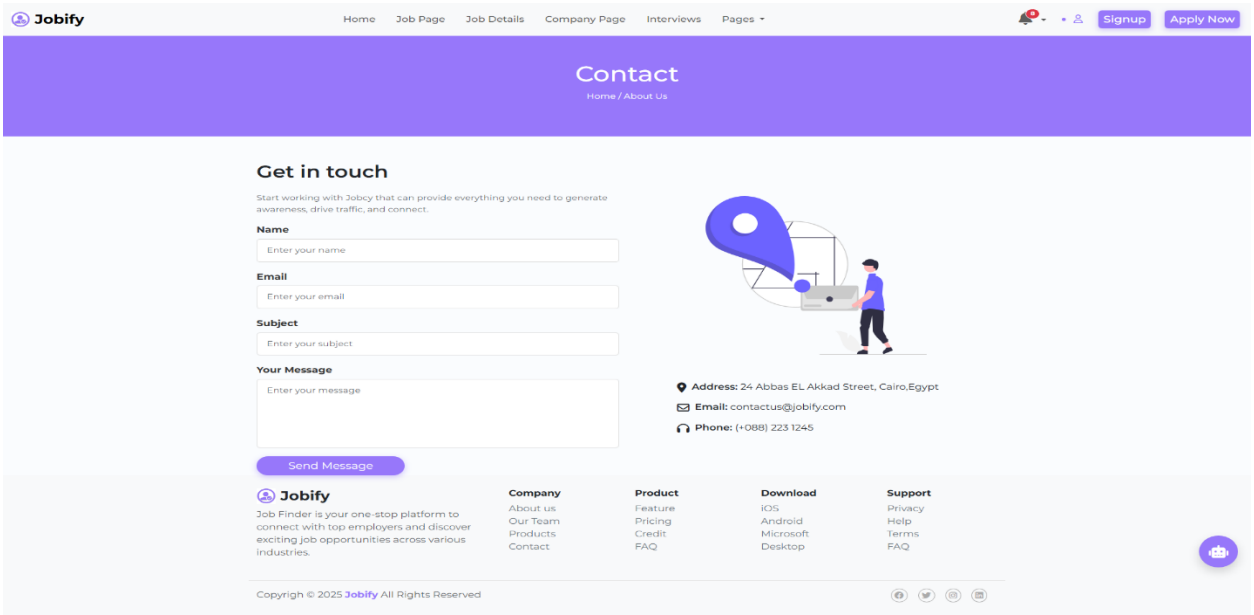

Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Jobify
Welcome Back
Email
Password
Login
Forgot your password?
Or sign up with
Google
Facebook
GitHub
Don't have an account? Sign Up


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Jobify
Sign Up
Username
Email
Password
Create my account
Or sign up with
Google
Facebook
GitHub
Already have an account? Login


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Services
Home / Services
Providing our trusted Services
It is a long established fact that a reader will be of a page when established fact looking at its layout.
Display Jobs
Explore a comprehensive list of available jobs, categorized for easy navigation, and keep track of the opportunities you're most interested in.
Learn More >
Jobs Details
Get in-depth details about each job, including responsibilities, requirements, and insights to help you make informed career choices.
Learn More >
Job Search
Search for jobs tailored to your preferences by title, location, or type, and find your dream opportunity quickly and effectively.
Learn More >
Companies
Discover top companies, learn about their cultures, and explore open positions to find your best workplace match.
Learn More >
Features Jobs
Browse highlighted job listings that showcase top opportunities from leading companies in various industries.
Learn More >
Bookmark Jobs
Save jobs of interest to revisit later and never miss out on an opportunity you're considering.
Learn More >
Apply for Jobs
Easily apply to jobs with a streamlined application process, increasing your chances of landing the position you desire.
Learn More >
Interview Preparation
Prepare for interviews with tips, mock sessions, and resources to build confidence and ace your interviews.
Learn More >
Subscribe for Jobs
Stay updated on the latest opportunities by subscribing to job alerts that match your preferences and skillset.
Learn More >
Subscribe to get the latest jobs
Enter Your Email
Subscribe
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
SAVED JOBS
Your Saved Jobs List
Junior Software Engineer (Full-Stack)
NextGen Innovations
New York, NY
Full time
5 sec ago
$50,000/Month
Work on both front-end and
Urgent
Full time
Apply Now
SW QA Engineer - RV
Siemens Software
Calibre's high-capacity and
Urgent
Remove
Apply Now
Junior UI Designer
CreativeCode Studios
Remote
Full time
3 days ago
€30,000/Month
Design engaging user interfaces and collaborate with UX teams.
Urgent
Full time
Remove
Apply Now
Delete Job?
Warning: Are you sure you want to delete this job?
Your job post will be permanently removed and you won't be able to see them again, including the ones you've shared with your friends.
Cancel
Yes, Delete


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
SAVED JOBS
Your Saved Jobs List
Junior Software Engineer (Full-Stack)
NextGen Innovations
New York, NY
Full time
5 sec ago
$50,000/Month
Work on both front-end and back-end tasks using React and Node.js.
Urgent
Full time
Remove
Apply Now
SW QA Engineer - RVE
Siemens Software
Remote
Part time
3 hours ago
$800/Hour
Calibre's high-capacity and high-performance tools are key to our customers'.
Urgent
Remove
Apply Now
Junior UI Designer
CreativeCode Studios
Remote
Full time
3 days ago
€30,000/Month
Design engaging user interfaces and collaborate with UX teams.
Urgent
Full time
Remove
Apply Now


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Contact
Home / About Us
Get in touch
Start working with Jobcy that can provide everything you need to generate awareness, drive traffic, and connect.
Name
Enter your name
Email
Enter your email
Subject
Enter your subject
Your Message
Enter your message
Send Message
Address: 24 Abbas EL Akkad Street, Cairo,Egypt
Email: contactus@jobify.com
Phone: (+088) 223 1245
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved

Jobify Home Job Page Job Details Company Page Interviews Pages Signup Apply Now

About Us

Home / About Us

ABOUT US

Why 10,000+ People Trust On Jobify?

Start working with Jobcy that can provide everything you need to generate awareness, drive traffic, connect. Dummy text is used by designers to occupy space.

- Your feuture job is here
- Creative Design
- New Job Innovations
- Create Resume
- Many Opportunities

Learn More

10,000 Available Jobs

7,500 Applications

8,850 Positive Feedback

9,875 Users

Key Features

Start working with Jobify that can provide everything you need to generate awareness, drive traffic, connect.

Job Listing | CV Analysis | Privacy Policy

Companies Page | Display Jobs | For Agencies

Quick Support | Bookmark Jobs | Real-time Analytics

See everything about your employee at one place.

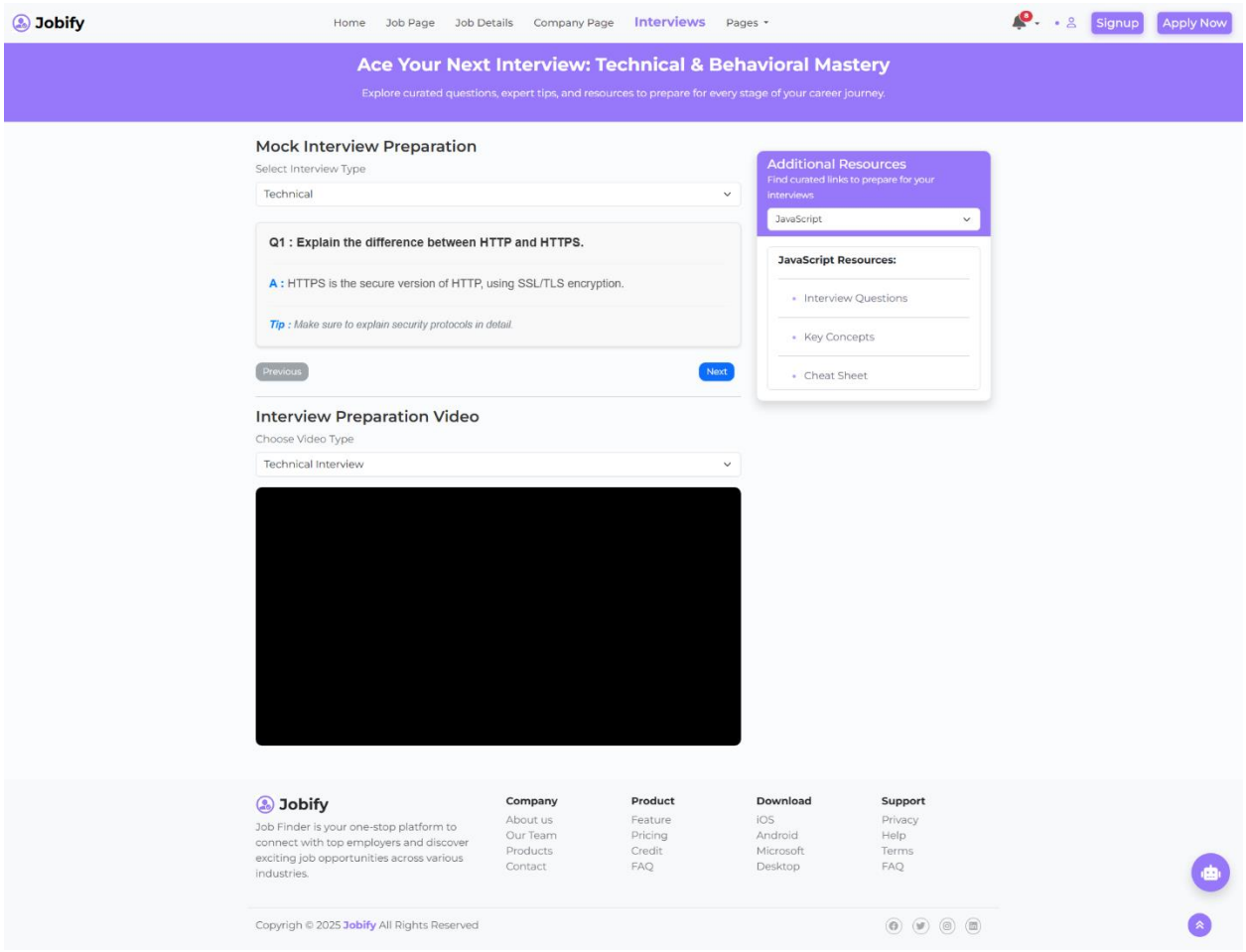
Start working with Jobify that can provide everything you need to generate awareness, drive traffic, connect

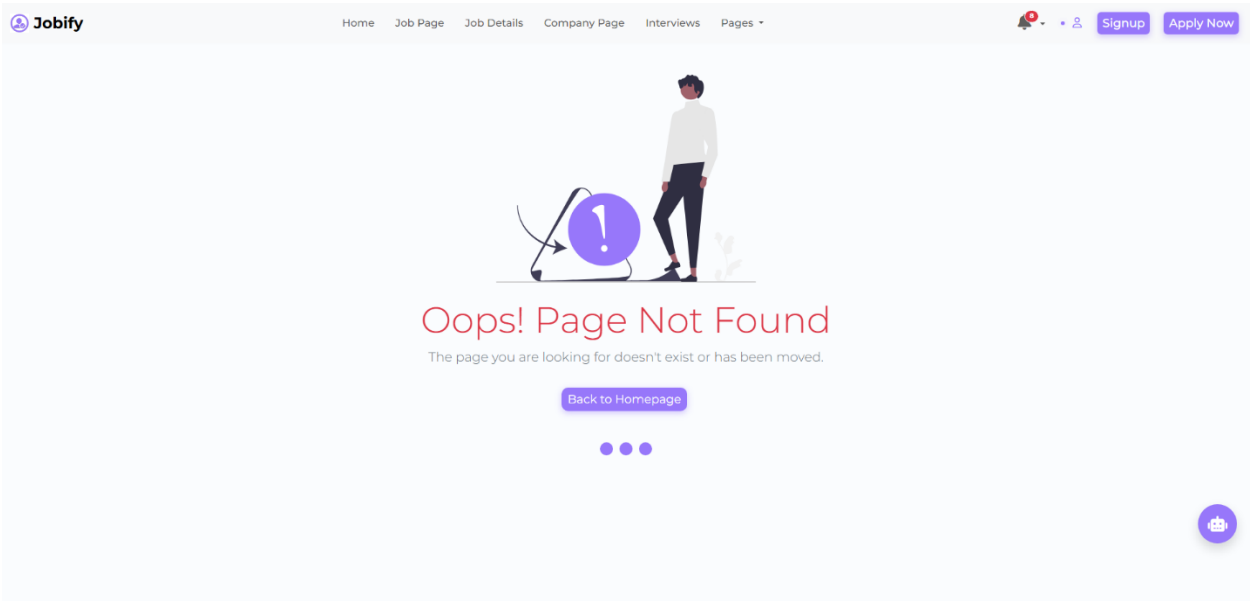
Get Started Now

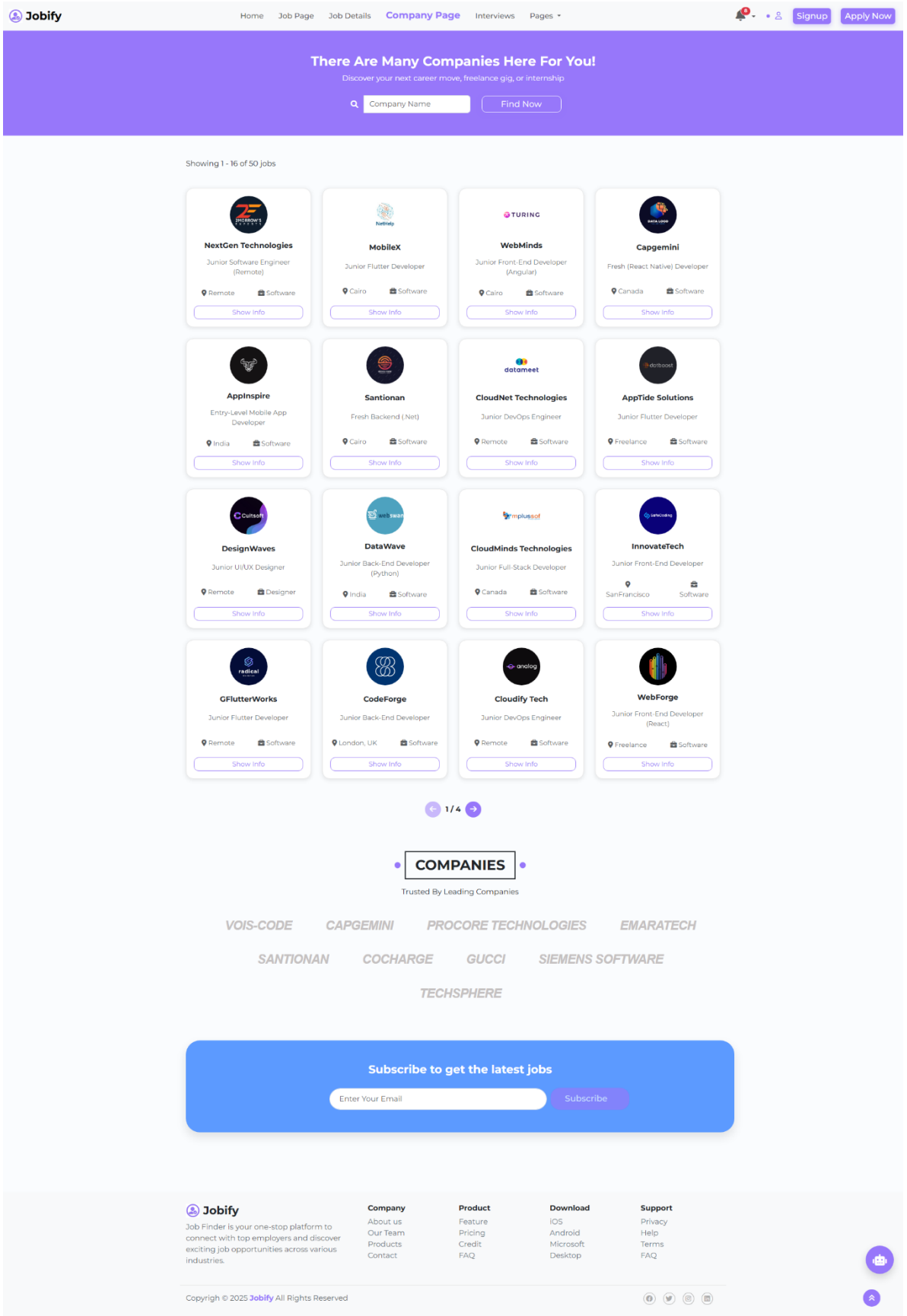
Jobify

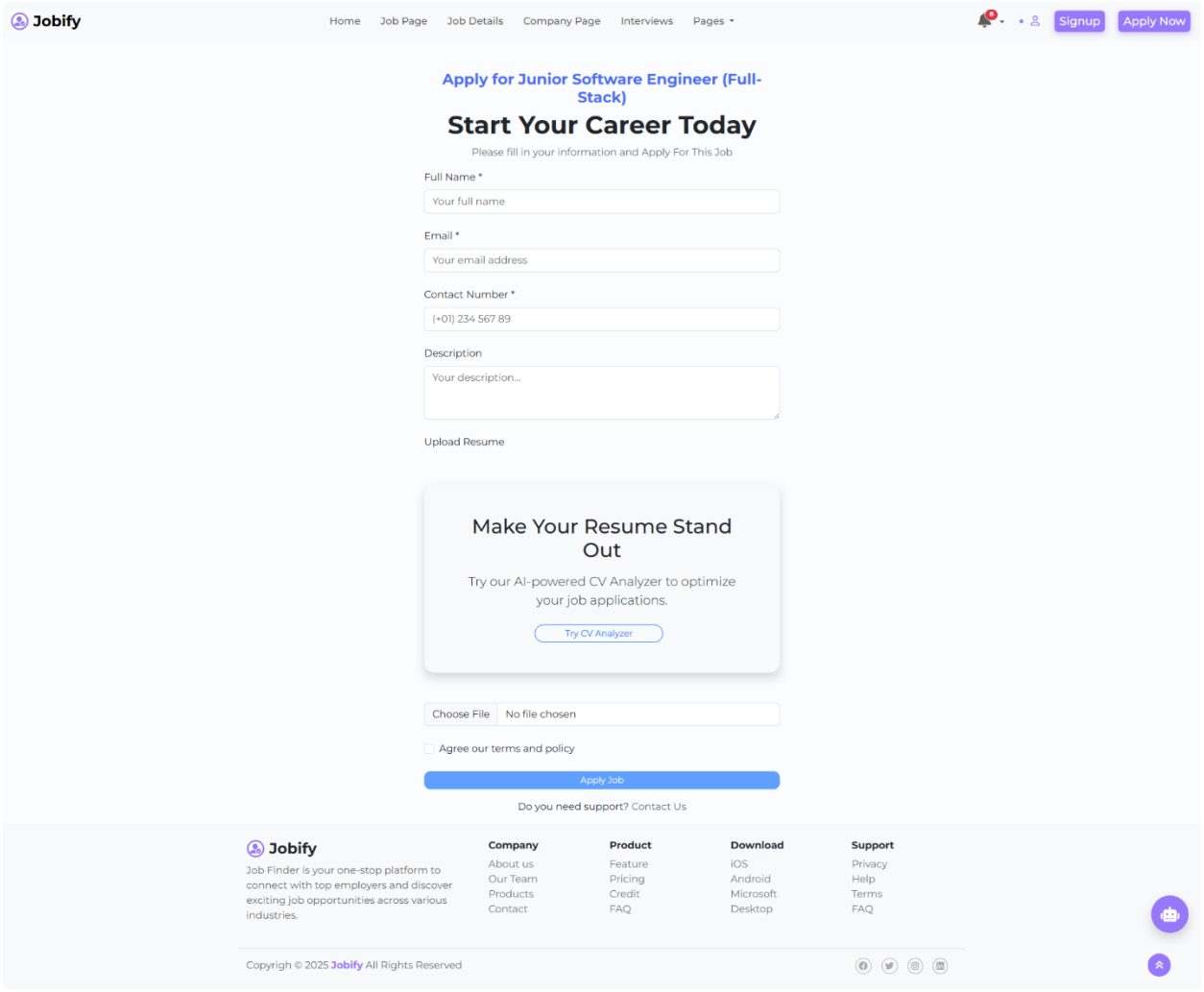
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.

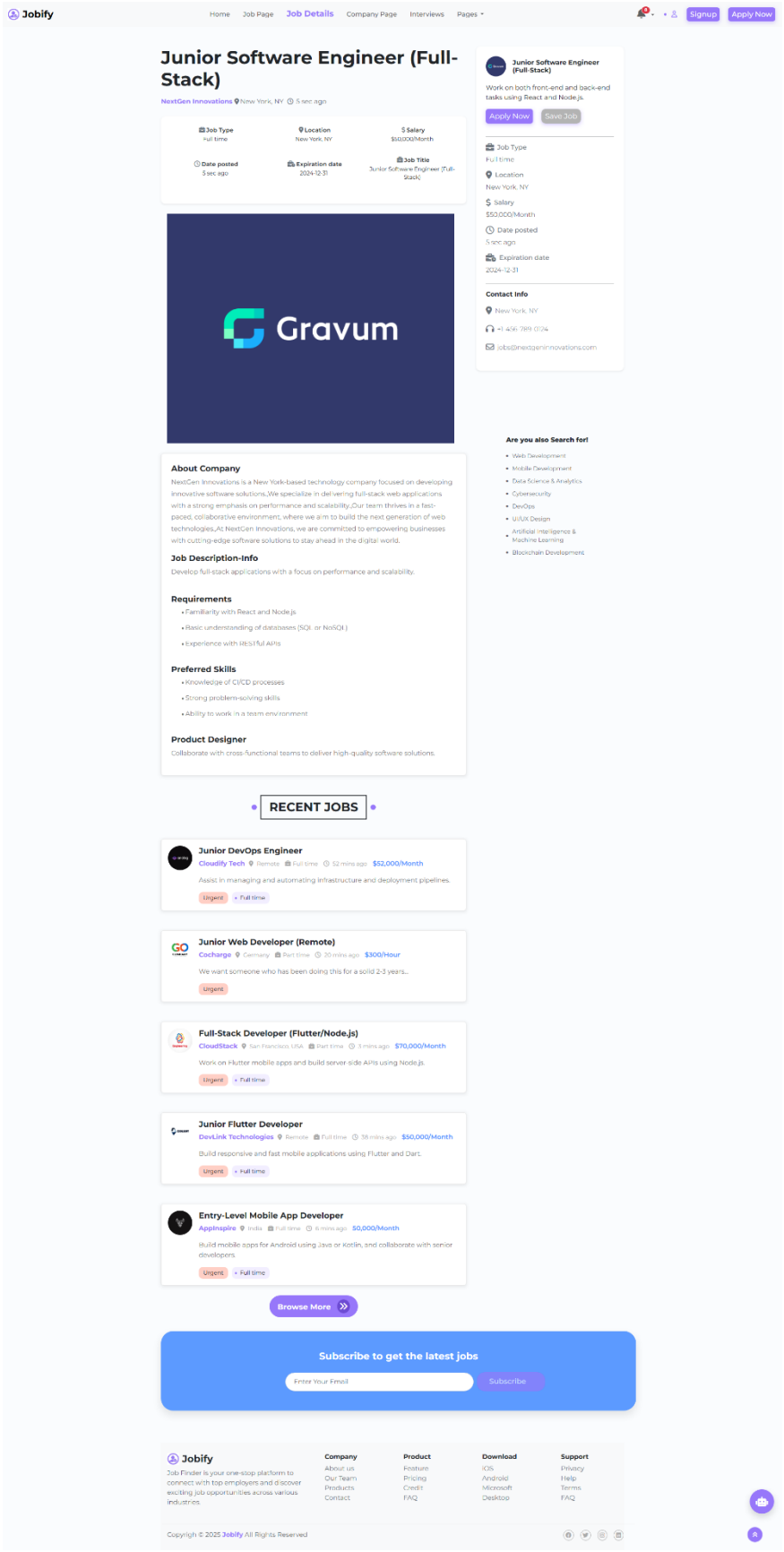
Company	Product	Download	Support
About us	Feature	iOS	Privacy
Our Team	Pricing	Android	Help
Products	Credit	Microsoft	Terms
Contact	FAQ	Desktop	FAQ

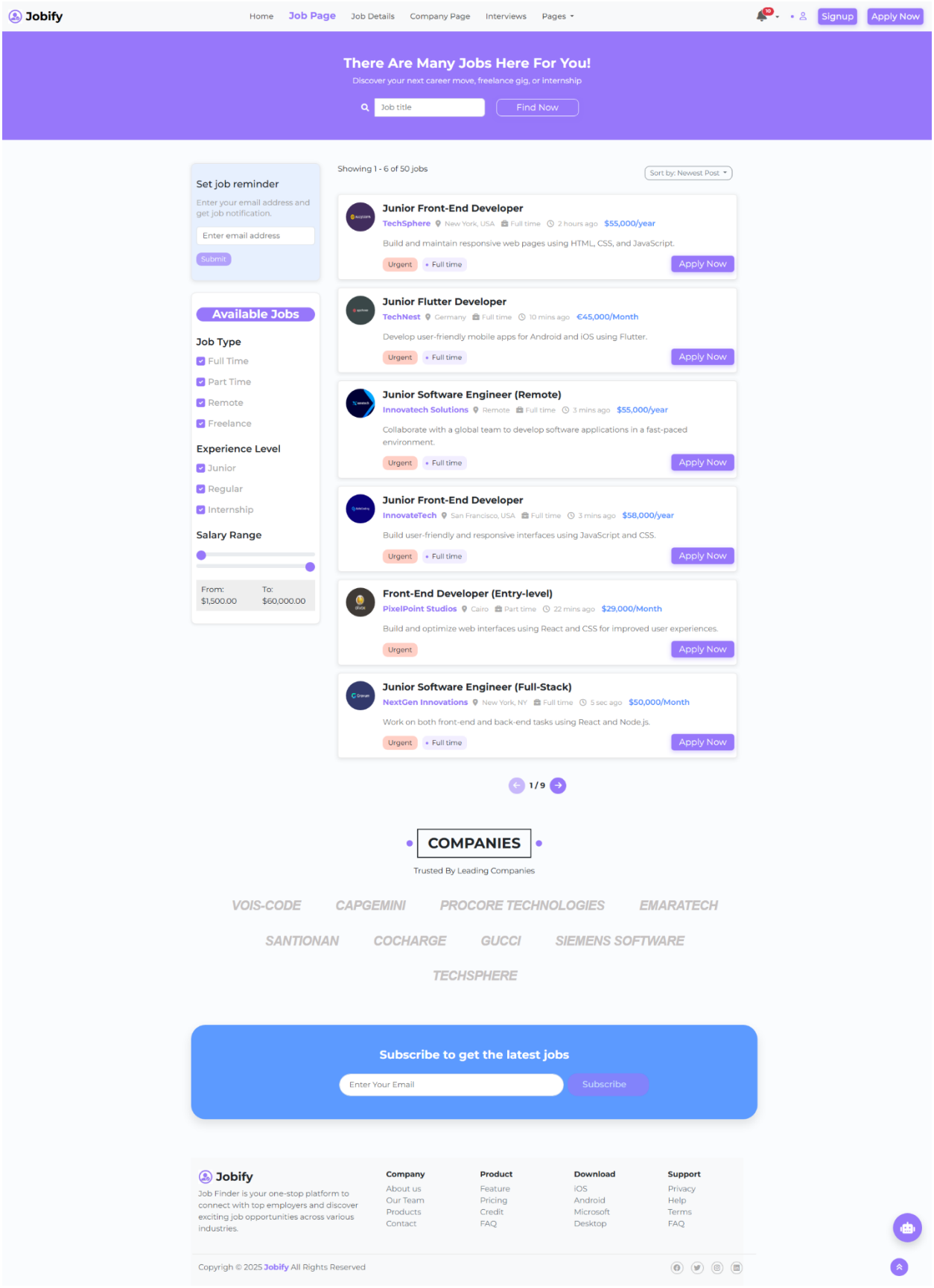

Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Ace Your Next Interview: Technical & Behavioral Mastery
Explore curated questions, expert tips, and resources to prepare for every stage of your career journey.
Mock Interview Preparation
Select Interview Type
Technical
Q1 : Explain the difference between HTTP and HTTPS.
A : HTTPS is the secure version of HTTP, using SSL/TLS encryption.
Tip : Make sure to explain security protocols in detail.
Previous
Next
Additional Resources
Find curated links to prepare for your interviews
JavaScript
JavaScript Resources:
Interview Questions
Key Concepts
Cheat Sheet
Interview Preparation Video
Choose Video Type
Technical Interview
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Oops! Page Not Found
The page you are looking for doesn't exist or has been moved.
Back to Homepage


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
There Are Many Companies Here For You!
Discover your next career move, freelance gig, or internship
Company Name
Find Now
Showing 1 - 16 of 50 jobs
NextGen Technologies
Junior Software Engineer (Remote)
Remote
Software
Show Info
MobileX
Junior Flutter Developer
Cairo
Software
Show Info
WebMinds
Junior Front-End Developer (Angular)
Cairo
Software
Show Info
Capgemini
Fresh (React Native) Developer
Canada
Software
Show Info
AppInspire
Entry-Level Mobile App Developer
India
Software
Show Info
Santionan
Fresh Backend (.Net)
Cairo
Software
Show Info
CloudNet Technologies
Junior DevOps Engineer
Remote
Software
Show Info
AppTide Solutions
Junior Flutter Developer
Freelance
Software
Show Info
DesignWaves
Junior UI/UX Designer
Remote
Designer
Show Info
DataWave
Junior Back-End Developer (Python)
India
Software
Show Info
CloudMinds Technologies
Junior Full-Stack Developer
Canada
Software
Show Info
InnovateTech
Junior Front-End Developer
SanFrancisco
Software
Show Info
GFlutterWorks
Junior Flutter Developer
Remote
Software
Show Info
CodeForge
Junior Back-End Developer
London, UK
Software
Show Info
Cloudify Tech
Junior DevOps Engineer
Remote
Software
Show Info
WebForge
Junior Front-End Developer (React)
Freelance
Software
Show Info
1 / 4
COMPANIES
Trusted By Leading Companies
VOIS-CODE
CAPGEMINI
PROCORE TECHNOLOGIES
EMARATECH
SANTIONAN
COCHARGE
GUCCI
SIEMENS SOFTWARE
TECHSPHERE
Subscribe to get the latest jobs
Enter Your Email
Subscribe
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
IOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Apply for Junior Software Engineer (Full-Stack)
Start Your Career Today
Please fill in your information and Apply For This Job
Full Name *
Your full name
Email *
Your email address
Contact Number *
(+01) 234 567 89
Description
Your description...
Upload Resume
Make Your Resume Stand Out
Try our AI-powered CV Analyzer to optimize your job applications.
Try CV Analyzer
Choose File
No file chosen
Agree our terms and policy
Apply Job
Do you need support? Contact Us
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
Junior Software Engineer (Full-Stack)
NextGen Innovations
New York, NY
5 sec ago
Job Type
Full time
Location
New York, NY
Salary
$50,000/Month
Date posted
5 sec ago
Expiration date
2024-12-31
Job Title
Junior Software Engineer (Full-Stack)
Gravum
About Company
NextGen Innovations is a New York-based technology company focused on developing innovative software solutions. We specialize in delivering full-stack web applications with a strong emphasis on performance and scalability. Our team thrives in a fast-paced, collaborative environment, where we aim to build the next generation of web technologies. At NextGen Innovations, we are committed to empowering businesses with cutting-edge software solutions to stay ahead in the digital world.
Job Description-Info
Develop full-stack applications with a focus on performance and scalability.
Requirements
Familiarity with React and Node.js
Basic understanding of databases (SQL or NoSQL)
Experience with RESTful APIs
Preferred Skills
Knowledge of CI/CD processes
Strong problem-solving skills
Ability to work in a team environment
Product Designer
Collaborate with cross-functional teams to deliver high-quality software solutions.
Junior Software Engineer (Full-Stack)
Work on both front-end and back-end tasks using React and Node.js.
Apply Now
Save Job
Job Type
Full time
Location
New York, NY
Salary
$50,000/Month
Date posted
5 sec ago
Expiration date
2024-12-31
Contact Info
New York, NY
+1 456 789 0124
jobs@nextgeninnovations.com
Are you also Search for!
Web Development
Mobile Development
Data Science & Analytics
Cybersecurity
DevOps
UI/UX Design
Artificial Intelligence & Machine Learning
Blockchain Development
RECENT JOBS
Junior DevOps Engineer
Cloudify Tech
Remote
Full time
52 mins ago
$52,000/Month
Assist in managing and automating infrastructure and deployment pipelines.
Urgent
Full time
Junior Web Developer (Remote)
Cocharge
Germany
Part time
20 mins ago
$300/Hour
We want someone who has been doing this for a solid 2-3 years...
Urgent
Full-Stack Developer (Flutter/Node.js)
CloudStack
San Francisco, USA
Part time
5 mins ago
$70,000/Month
Work on Flutter mobile apps and build server-side APIs using Node.js.
Urgent
Full time
Junior Flutter Developer
DevLink Technologies
Remote
Full time
38 mins ago
$50,000/Month
Build responsive and fast mobile applications using Flutter and Dart.
Urgent
Full time
Entry-Level Mobile App Developer
AppInspire
India
Full time
6 mins ago
$0,000/Month
Build mobile apps for Android using Java or Kotlin, and collaborate with senior developers
Urgent
Full time
Browse More
Subscribe to get the latest jobs
Enter Your Email
Subscribe
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyright © 2025 Jobify All Rights Reserved


Jobify
Home
Job Page
Job Details
Company Page
Interviews
Pages
Signup
Apply Now
There Are Many Jobs Here For You!
Discover your next career move, freelance gig, or internship
Job title
Find Now
Set job reminder
Enter your email address and get job notification.
Enter email address
Submit
Available Jobs
Job Type
Full Time
Part Time
Remote
Freelance
Experience Level
Junior
Regular
Internship
Salary Range
From: $1,500.00
To: $60,000.00
Showing 1 - 6 of 50 jobs
Sort by: Newest Post
Junior Front-End Developer
TechSphere New York, USA Full time 2 hours ago $55,000/year
Build and maintain responsive web pages using HTML, CSS, and JavaScript.
Urgent Full time
Apply Now
Junior Flutter Developer
TechNest Germany Full time 10 mins ago €45,000/Month
Develop user-friendly mobile apps for Android and iOS using Flutter.
Urgent Full time
Apply Now
Junior Software Engineer (Remote)
Innovatech Solutions Remote Full time 3 mins ago $55,000/year
Collaborate with a global team to develop software applications in a fast-paced environment.
Urgent Full time
Apply Now
Junior Front-End Developer
InnovateTech San Francisco, USA Full time 3 mins ago $58,000/year
Build user-friendly and responsive interfaces using JavaScript and CSS.
Urgent Full time
Apply Now
Front-End Developer (Entry-level)
PixelPoint Studios Cairo Part time 22 mins ago $29,000/Month
Build and optimize web interfaces using React and CSS for improved user experiences.
Urgent
Apply Now
Junior Software Engineer (Full-Stack)
NextGen Innovations New York, NY Full time 5 sec ago $50,000/Month
Work on both front-end and back-end tasks using React and Node.js.
Urgent Full time
Apply Now
1 / 9
COMPANIES
Trusted By Leading Companies
VOIS-CODE
CAPGEMINI
PROCORE TECHNOLOGIES
EMARATECH
SANTIONAN
COCHARGE
GUCCI
SIEMENS SOFTWARE
TECHSPHERE
Subscribe to get the latest jobs
Enter Your Email
Subscribe
Jobify
Job Finder is your one-stop platform to connect with top employers and discover exciting job opportunities across various industries.
Company
About us
Our Team
Products
Contact
Product
Feature
Pricing
Credit
FAQ
Download
iOS
Android
Microsoft
Desktop
Support
Privacy
Help
Terms
FAQ
Copyrigh © 2025 Jobify All Rights Reserved

7.2 Wireframes & Prototypes

Below are brief overviews of the major screens with design goals and interactions.

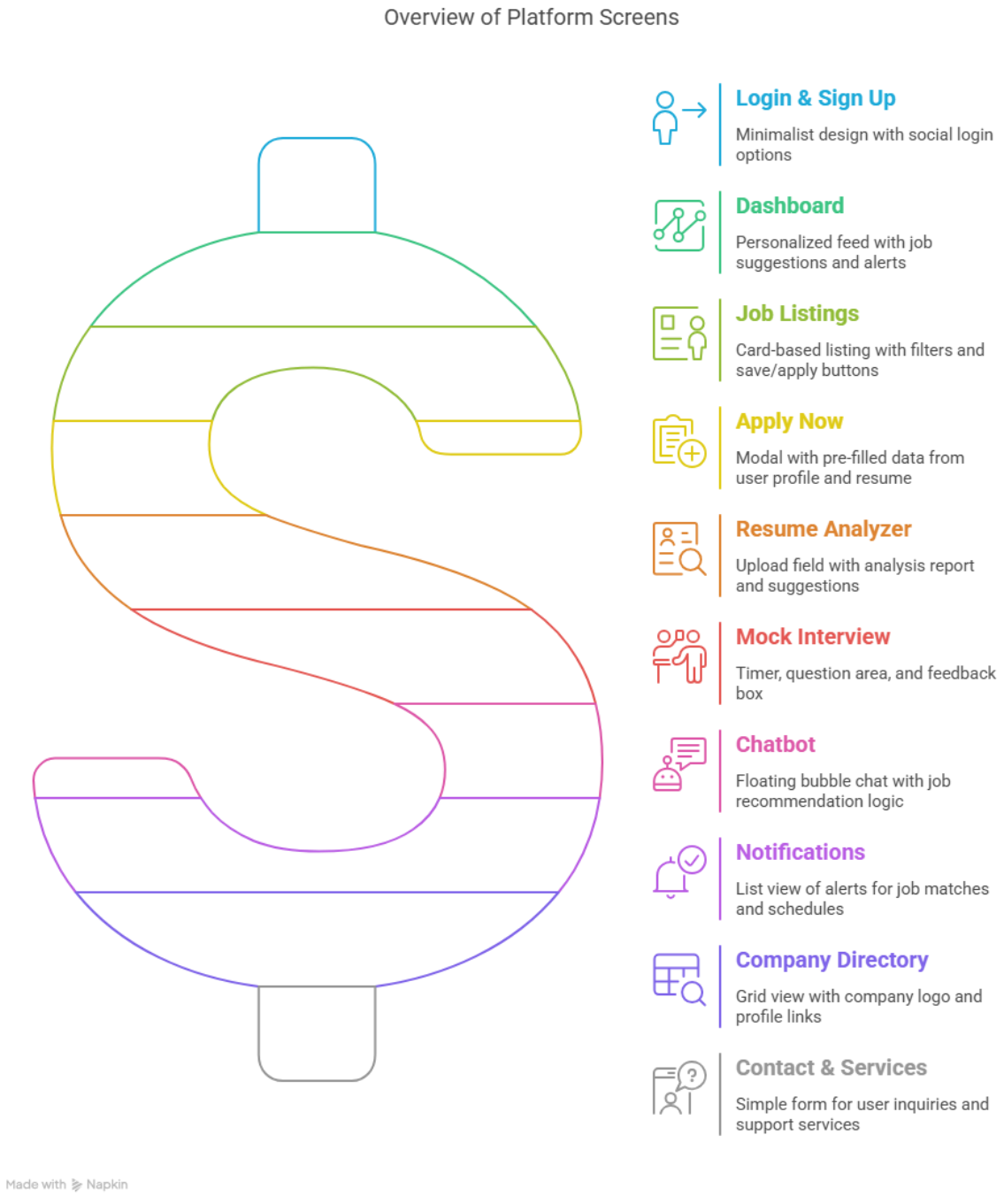


Figure 7.2 Overview Screens

Chapter Eight
Result & Discussion

Result & Discussion

The development of Jobify brought a variety of technical, design, and operational challenges. These obstacles tested the robustness of our stack and our ability to deliver a high-quality, AI-integrated career platform.

1. **Inaccurate Resume-to-Job Matching**

Challenge:

The initial AI model provided job matches based on simple keyword comparison, leading to irrelevant suggestions.

Solution:

Integrated NLP techniques to understand context, used TF-IDF and spaCy for semantic analysis, and improved dataset diversity. Fine-tuned resume scoring based on skill alignment, experience level, and job type.

---

2. **Difficulties in AI Model Integration with .NET Backend**

Challenge:

Python-based AI modules (resume analyzer and chatbot) needed to be called from a .NET Core backend, which caused communication issues.

Solution:

Used Flask to expose Python services as RESTful APIs**,** consumed them in .NET using `HttpClient`, and implemented fallback handlers for service failure tolerance.

3. **Chatbot Response Delays and Irrelevance**

Challenge:

Chatbot either delayed responses or gave general answers that didn't help the user.

Solution:

Improved natural language understanding using transformer models (like DistilBERT) and combined it with rule-based logic for instant, FAQ-style replies.

---

5. **Resume Upload Format Compatibility**

Challenge:

Some users uploaded unsupported file types or corrupt resumes, breaking parsing.

Solution:

Restricted uploads to **PDF and DOCX**, used Python's `docx2txt` and `PyMuPDF` to parse resumes, and validated content before analysis.

---

6. **Scalability & Performance Issues with SQL Server**

Challenge:

As user data and job listings grew, performance dropped and queries took longer.

Solution:

Implemented query optimization**,** indexing, stored procedures**,** and asynchronous data access**.** Future plans include horizontal sharding for large-scale growth.

7. **Interview Feedback Inaccuracy**

Challenge:

Mock interview module gave vague or inconsistent performance ratings.

Solution:

Trained AI model with labeled mock interview datasets. Integrated tone and sentiment analysis to better evaluate verbal responses.

---

8. **Frontend Performance Lag on Mobile Devices**

Challenge:

Some pages and forms were not responsive, affecting user experience on smartphones.

Solution:

Refined Bootstrap layouts, added media queries**,** lazy-loaded Angular components, and conducted real-device testing.

---

9. **Search Filtering Logic Was Too Rigid**

Challenge:

Job seekers couldn’t mix filter types or adjust salary ranges dynamically.

Solution:

Used Angular reactive forms with live filtering and backend query strings to allow multi-layered filter combinations**.**

10. **Password Reset Function Was Vulnerable to Abuse**

Challenge:

Password reset emails could be triggered unlimited times.

Solution:

Added rate-limiting, one-time reset tokens with expiry, and CAPTCHA for added security.

---

11. **Keeping Angular Validation Consistent with Backend Rules**

Challenge:

Form validations passed in Angular but failed at the backend due to mismatch in logic.

Solution:

Centralized validation logic in shared models and replicated rules using Angular services to mirror .NET server logic.

---

12. **Incomplete Data in Employer Profiles**

Challenge:

Recruiters often left out key data like company logos, addresses, or job descriptions.

Solution:

Added step-by-step employer onboarding wizard**,** made certain fields required, and included inline data tips.

13. **Difficulty in Managing Notification System**

Challenge:

Job seekers weren't receiving timely job or application status notifications.

Solution:

Used SignalR (for real-time) and email triggers via SendGrid. Stored notification states in SQL and added a user notification panel.

---

14. **Wireframe and Final UI Inconsistencies**

Challenge:

The final UI drifted away from original wireframes, causing confusion in implementation.

Solution:

Used Figma components, version control, and collaboration between designers and Angular developers for synchronized updates.

---

15. **Testing AI Feedback Without Real Users**

Challenge:

Hard to evaluate resume feedback or mock interview analysis without real data.

Solution:

Created synthetic CV and Q&A datasets for testing, ran closed beta with real users, and collected qualitative feedback.

Chapter Nine
Future Work & Recommendations

Future Work & Recommendations

As technology continues to evolve and user expectations grow, platforms like Jobify must remain adaptive, scalable, and forward-thinking. While the current version of Jobify delivers a strong foundation through AI-driven career services tailored for fresh graduates and early professionals, the digital employment landscape is constantly shifting—with new challenges, innovations, and opportunities emerging every day.

This chapter outlines the strategic roadmap and forward-looking recommendations to future-proof Jobify, enhance its intelligence, expand its features, and create a more immersive, impactful, and inclusive user experience.

These recommendations are built on insights gathered during the design, development, and testing phases, as well as on current market trends, user feedback, and the evolving needs of both job seekers and employers.

From AI advancements and mobile-first strategies to blockchain integration and global language support, this chapter acts as a vision blueprint for Jobify's long-term sustainability and innovation. Each point discussed aims to increase user engagement, boost platform credibility, and maximize job placement success rates through intelligent design and technology integration.

By proactively planning future enhancements, Jobify can not only stay competitive but also redefine the standards for digital career platforms in the modern era.

1. Enhanced AI Personalization

What to Improve:

While the current AI models provide solid recommendations, there's potential to make Jobify more personalized.

Future Work:

- Incorporate reinforcement learning to adapt job recommendations based on real-time user behavior.
- Use deep learning models to create more nuanced profile-job matches.
- Implement sentiment analysis to understand user satisfaction from feedback and interactions.

---

2. Video Interview Assessment System

What to Improve:

Mock interviews are currently text or voice-based, lacking non-verbal feedback.

Future Work:

- Develop a video-based AI interview analyzer that evaluates facial expressions, eye contact, and confidence.
- Integrate with platforms like WebRTC or Twilio for video streaming.
- Use computer vision techniques to detect body language cues and stress indicators.

---

3. Integration with Job Market APIs

What to Improve:

Currently, job listings are platform-specific or manually added.

Future Work:

- Integrate with APIs from LinkedIn, Indeed, or Glassdoor to automatically fetch updated listings.
- Leverage web scraping with permission where APIs aren't available.

---

4. Advanced Analytics Dashboard for Users

What to Improve:

Users have access to basic insights but no deep career analytics.

Future Work:

- Add visual dashboards that track job application status, resume health score, interview trends, etc.
- Provide career trajectory forecasting using machine learning.
- Introduce salary benchmarking tools and competitor analysis.

---

5. Employer Dashboard and Analytics

What to Improve:

Employers need better tools to evaluate applicants and hiring campaigns.

Future Work:

- Develop a dedicated recruiter dashboard for tracking job post performance.
- AI-powered tools to rank applicants based on role compatibility.
- Visual heatmaps of active regions and candidate engagement.

6. Gamification of Learning and Career Growth

What to Improve:

User engagement can be increased through motivational design.

Future Work:

- Introduce badges**,** achievements, and career milestones**.**
- Provide AI-powered suggestions to unlock new badges (e.g., “Top CV Score”, “Interview Master”).

---

7. Real-Time Collaboration Tools

What to Improve:

Collaboration for referrals, networking, and group interview prep isn’t supported.

Future Work:

- Add chat rooms**,** video group discussions, and peer review spaces.
- Create mentorship rooms where experienced users guide newcomers.

---

8. Multi-Language and Localization Support

What to Improve:

The current system supports English only.

Future Work:

- Add Arabic, French, and Spanish interfaces and content.
- Enable multi-language NLP models for resume parsing and chatbot interaction.

9. Mobile App Development

What to Improve:

Jobify is web-based and lacks a mobile-native solution.

Future Work:

- Develop native Android and iOS apps using Flutter or React Native.
- Ensure offline resume editing, push notifications, and local storage sync.

---

10. Blockchain for Credential Verification

What to Improve:

Verifying user qualifications and resumes is manual and prone to fraud.

Future Work:

- Integrate blockchain to allow credential storage and verification.
- Partner with universities and institutions to issue blockchain-based certificates.

---

11. Voice-Enabled Job Search & Guidance

What to Improve:

Accessibility for visually impaired users is limited.

Future Work:

- Add voice-enabled chatbot and search features.
- Use text-to-speech (TTS) and speech-to-text (STT) tools for platform navigation.

12. Employer Rating and Review System

What to Improve:

Users can't view previous experiences with employers.

Future Work:

- Add rating and review system for companies.
- Include flags for unethical practices or misleading job descriptions.

---

13. Internship-to-Job Pipeline

What to Improve:

No clear pathway from internships to full-time roles.

Future Work:

- Allow employers to flag high-potential interns for conversion.
- Introduce progress tracking and evaluation tools for intern performance.

---

14. AI Career Coach

What to Improve:

Career navigation is generic.

Future Work:

- Introduce an AI career advisor that helps users build personalized development plans, learn new skills, and align with market demands.

15. Feedback Loop for Continuous Improvement

What to Improve:

User feedback is collected passively, not actionable.

Future Work:

- Add in-app prompts, surveys, and heatmaps to analyze feature usage.
- Apply A/B testing to measure improvements and deploy only the best variants.

Chapter Ten
Conclusion

Conclusion

Jobify was developed with a clear vision: to bridge the gap between fresh graduates and meaningful employment by leveraging the power of AI and modern web technologies. Throughout this project, we addressed key challenges faced by early-career job seekers—such as resume optimization, job matching, interview preparation, and access to part-time or internship opportunities.

The platform integrates intelligent features like AI-driven resume analysis, real-time chatbot support, mock interview simulations, and advanced job filtering tools—all within a responsive and user-friendly interface. By combining Angular, .NET, SQL Server, and Python-based AI models, Jobify delivers a seamless experience that not only connects users with opportunities but also equips them with the insights and confidence needed to succeed in the competitive job market.

From system design to implementation, every phase of this project was driven by user needs and a commitment to usability, scalability, and innovation. While challenges arose during the development lifecycle—such as integrating AI components and ensuring data security—they were met with practical solutions and forward-thinking approaches.

Looking ahead, Jobify has immense potential for expansion, with possible additions including employer dashboards, mobile applications, and global language support. With continuous improvement, Jobify can evolve into a comprehensive employment ecosystem that supports users throughout their entire career journey.

In conclusion, Jobify is not just a job portal—it is a career empowerment platform, and this project marks the beginning of its journey toward transforming the future of job hunting.

References

1. **Angular** – Angular Documentation. Retrieved from https://angular.io
2. **Bootstrap** – Responsive web design framework. Retrieved from https://getbootstrap.com
3. **Font Awesome** – Icon toolkit and vector icons. Retrieved from https://fontawesome.com
4. **Google Fonts** – Free and open-source fonts for web and mobile apps. Retrieved from https://fonts.google.com
5. **HTML5 & CSS3** – W3C standards for web structure and styling. Retrieved from https://www.w3.org
6. **ASP.NET Core** – Microsoft's cross-platform, high-performance framework. Retrieved from https://learn.microsoft.com/en-us/aspnet/core
7. **Node.js** – JavaScript runtime for scalable network applications. Retrieved from https://nodejs.org
8. **Express.js** – Fast and minimalist Node.js web framework. Retrieved from https://expressjs.com
9. **Microsoft SQL Server** – Enterprise-grade relational database. Retrieved from https://learn.microsoft.com/en-us/sql/sql-server
10. **Entity Framework Core** – ORM for .NET Core applications. Retrieved from https://learn.microsoft.com/en-us/ef/core
11. **Python** – General-purpose programming language. Retrieved from https://www.python.org

12. **Google Colab** – Cloud-based Jupyter notebook environment. Retrieved from **https://colab.research.google.com**
13. **Scikit-learn** – Machine learning library for Python. Retrieved from **https://scikit-learn.org**
14. **spaCy** – NLP library in Python. Retrieved from **https://spacy.io**
15. **TensorFlow** – End-to-end open-source machine learning platform. Retrieved from **https://www.tensorflow.org**
16. **Flask** – Lightweight Python web framework for APIs. Retrieved from **https://flask.palletsprojects.com**
17. **OpenAI** – ChatGPT & NLP models API. Retrieved from **https://platform.openai.com/docs**
18. **Figma** – Interface design & prototyping tool. Retrieved from **https://www.figma.com**
19. **Freepik** – Free vectors, photos, and PSDs. Retrieved from **https://www.freepik.com**
20. **Pexels** – Free stock photos and videos. Retrieved from **https://www.pexels.com**
21. **Unsplash** – High-quality royalty-free images. Retrieved from **https://unsplash.com**
22. **FlatIcon** – Icon resources for UI design. Retrieved from **https://www.flaticon.com**
23. **IconFinder** – Search engine for icons and illustrations. Retrieved from **https://www.iconfinder.com**

24. **Visual Studio 2022** – IDE for .NET and C#. Retrieved from https://visualstudio.microsoft.com
25. **Visual Studio Code** – Source code editor for all platforms. Retrieved from https://code.visualstudio.com
26. **Postman** – API platform for building and using APIs. Retrieved from https://www.postman.com
27. **Swagger UI** – API documentation and testing tool. Retrieved from https://swagger.io
28. **Microsoft Azure** – Cloud computing and hosting platform. Retrieved from https://azure.microsoft.com
29. **Docker** – Platform for containerizing applications. Retrieved from https://www.docker.com
30. **GitHub** – Version control and code collaboration. Retrieved from https://github.com
31. **Vercel** – Frontend hosting platform optimized for React, Angular, and Jamstack. Retrieved from https://vercel.com
32. **Git-Hup Repo :** https://github.com/KariemHatem/Jobify-Platform
33. **Live Site Demo :** https://jobify-platform.vercel.app/