# Software Requirements Specification
# For
# Resume Builder

**Version 1.0 approved**

**Prepared by: -**
**Tanna Harsh Nimesh Bhai (22SOECE13037)**
**Dhwaj Shah (22SOECE13034)**
**Jani Jeet Tapan Bhai (22SOECE13018)**

**Language: -**

**React**
**&**
**ASP.NET Core MVC**

**R.K University**

# Table of Contents

**Table of Contents** ..... **ii**
**Revision History** ..... **ii**
**1. Introduction** ..... **1**
1.1 Purpose ..... 1
1.2 Document Conventions ..... 1
1.3 Intended Audience and Reading Suggestions ..... 2
1.4 Project Scope ..... 3
1.5 References ..... 4
**2. Overall Description** ..... **5**
2.1 Product Perspective ..... 5
2.2 Product Features ..... 6
2.3 User Classes and Characteristics ..... 8
2.4 Operating Environment ..... 9
2.5 Design and Implementation Constraints ..... 11
2.6 User Documentation ..... 12
2.7 Assumptions and Dependencies ..... 13
**3. System Features** ..... **14**
3.1 System Feature 1 .....
3.2 System Feature 2 (and so on) .....
**4. External Interface Requirements** .....
4.1 User Interfaces ..... 16
4.2 Hardware Interfaces ..... 16
4.3 Software Interfaces ..... 17
4.4 Communications Interfaces ..... 17
**5. Other Nonfunctional Requirements** ..... **18**
5.1 Performance Requirements ..... 19
5.2 Safety Requirements ..... 19
5.3 Security Requirements ..... 19
5.4 Software Quality Attributes ..... 20
**6. Other Requirements** ..... **20**
**Appendix A: Glossary** ..... **21**
**Appendix B: Analysis Models** ..... **21**

# Revision History

| Name | Date | Reason For Changes | Version |
|---|---|---|---|
| | | | |
| | | | |

# 1. Introduction

## 1.1 Purpose

The "RESUME BUILDER" project is a web-based application designed for organizing information, such as educational, personal, and professional aspects of individuals for both new and seasoned users. They can easily enter their information into the fields, and a resume will be generated for them

## 1.2 Document Conventions

1.2.1Code Documentation:

- Use XML documentation comments for documenting classes, methods, properties, and parameters in the code. These comments can be processed by documentation generation tools like Doxygen or Sandcastle to produce API documentation.
- Follow a consistent style for documenting code, including descriptions of the purpose, parameters, return values, exceptions (if applicable), and usage examples for methods and classes.

1.2.2 API Documentation:

- Document the APIs exposed by the application, including RESTful endpoints if applicable.
- Use tools like Swagger/Open API Specification to automatically generate API documentation, or manually document the endpoints, request/response formats, authentication mechanisms, and usage examples.

1.2.3 User Interface (UI) Guidelines:

- Document the design principles, layout guidelines, and interaction patterns used in the user interface (UI).
- Include wireframes, mockups, or design specifications to illustrate the visual and interactive elements of the UI.
- Provide guidelines for accessibility, responsiveness, and cross-browser compatibility.

1.2.4 Database Schema Documentation:

- Document the database schema, including tables, columns, data types, constraints, relationships, and indexes.
- Use database documentation tools or data modeling software to generate graphical representations of the database schema.

1.2.5 Project Structure and Architecture:

- Document the overall architecture of the project, including the high-level components, layers, and their interactions.
- Provide an overview of the project structure, including directories, files, and their purposes.
- Include diagrams (e.g., UML diagrams) to illustrate the architecture and relationships between components.

1.2.6 Coding Standards and Conventions:

- Define coding standards and conventions for the project, covering naming conventions, formatting rules, code organization, and best practices.
- Document any specific guidelines or patterns used in the project, such as design patterns, SOLID principles, or architectural patterns.

1.2.7 Deployment and Configuration:

- Document the deployment process, including instructions for setting up the development environment, building the application, and deploying it to various environments (e.g., staging, production).
- Provide configuration guidelines for different deployment environments, including database connection strings, API keys, and other environment-specific settings.

1.2.8 Testing and Quality Assurance:

- Document testing strategies, including unit tests, integration tests, and end-to-end tests.
- Define quality assurance processes, including code reviews, static code analysis, and continuous integration/delivery (CI/CD) pipelines.

## 1.3 Intended Audience and Reading Suggestions

1.3.1 Job Seekers:

- Provide user guides or tutorials on how to use the resume builder effectively.
- Offer tips and best practices for creating professional resumes.
- Include FAQs or knowledge base articles to address common questions and concerns.

1.3.2 Developers:

- Document the architecture, code structure, and APIs used in the resume builder application.
- Provide coding standards and guidelines for contributing to the project or extending its functionality.
- Include technical documentation for setting up the development environment, building the application, and deploying it to various environments.

1.3.3 Designers:

- Provide design guidelines, including UI/UX patterns, typography, color schemes, and layout principles.
- Include design assets, such as wireframes, mockups, and design specifications.
- Offer resources or references for design inspiration and trends in resume design.

1.3.4 Recruiters and HR Professionals:

- Provide information on how recruiters can use the resume builder to streamline the hiring process.
- Offer insights into resume trends and best practices to help recruiters evaluate candidate resumes effectively.

1.3.5 Educators and Career Counselors:

- Offer educational resources, such as lesson plans or workshop materials, for teaching resume writing skills using the resume builder.
- Provide curriculum guides or training modules on resume building and job search strategies.

1.3.6 Administrators and System Operators:

- Document system requirements, installation procedures, and maintenance tasks for administrators responsible for managing the resume builder application.
- Include troubleshooting guides or FAQs to assist with resolving common technical issues.

1.3.7 General Audience:

- Offer blog posts, articles, or webinars on topics related to resume writing, job search strategies, career development, and personal branding.
- Provide resources or links to external websites with additional information on related topics.

## 1.4 Project Scope

The project scope of a resume builder typically involves defining the features, functionalities, and boundaries of the application. Here's an outline of the project scope for a resume builder:

1.4.1 Core Features:

- User Registration and Authentication: Allow users to create accounts and log in securely.
- Resume Creation: Provide tools for users to input their personal information, education, work experience, skills, and other relevant details to generate a professional-looking resume.
- Resume Templates: Offer a selection of customizable templates for users to choose from, with options for different styles, layouts, and designs.
- Editing and Formatting: Enable users to edit and format their resumes easily, including options for adding sections, rearranging content, changing fonts, colors, and styles.
- Saving and Version Control: Allow users to save their resumes, create multiple versions, and track changes over time.
- Export and Download: Provide functionality to export resumes in various formats (e.g., PDF, Word, HTML) and allow users to download or share them.
- Collaboration: Support collaboration features, such as sharing resumes with others for feedback or collaboration on a shared document.
- Printing: Enable users to print their resumes directly from the application with optimal formatting.

1.4.2 Optional Features:

- Integration with Job Boards: Allow users to upload their resumes directly to job search platforms or integrate with popular job boards for seamless application submissions.
- LinkedIn Integration: Provide options for users to import their LinkedIn profiles or export their resumes to LinkedIn.
- Analytics and Insights: Track usage metrics, such as the number of resumes created, views, downloads, and user engagement, to gain insights into user behavior and preferences.
- Customization Options: Offer advanced customization features, such as custom sections, layouts, themes, and branding options.
- Language and Localization: Support multiple languages and localization options to cater to users from different regions.

1.4.3 Administrative Features:

- Admin Dashboard: Provide administrators with a dashboard to manage users, resumes, templates, and settings.
- User Management: Allow administrators to manage user accounts, roles, permissions, and access controls.
- Content Management: Enable administrators to add, edit, or remove resume templates, sections, and other content.

1.4.4 Technical Considerations:

- Platform and Technology: Determine the technology stack and platform (e.g., web, mobile) for developing the resume builder application.
- Scalability: Design the application architecture to scale efficiently to accommodate growing user base and increasing demand.
- Security: Implement robust security measures to protect user data, prevent unauthorized access, and secure transactions.
- Performance: Optimize the performance of the application to ensure fast loading times, responsiveness, and reliability.

1.4.5 Constraints and Assumptions:

- Timeframe: Define the project timeline and constraints, including deadlines, milestones, and resource availability.
- Budget: Consider budgetary constraints and resource allocation for development, testing, and maintenance.
- Assumptions: Document any assumptions or dependencies related to user requirements, technical specifications, or external factors.

## 1.5 References

When developing a resume builder project, you may need to refer to various resources for inspiration, guidance, and technical information. Here are some references you might find useful:

1.5.1 Online Resume Builders:

- Explore existing online resume builder platforms to understand their features, user experience, and design. Examples include:
- LinkedIn Resume Builder
- Zety Resume Builder
- Canva Resume Maker
-

1.5.2 Open-Source Projects:

- Look for open-source resume builder projects on platforms like GitHub. Studying their codebase can provide insights into implementation details, best practices, and potential libraries or frameworks to use.
- Example projects:
- ResumeBuilder.NET
- Resume Builder by Aaron Shaf

1.5.3 Tutorials and Guides:

- Follow tutorials and guides on building resume builder applications using specific technologies or frameworks. Many online resources offer step-by-step instructions and code samples.
- Example tutorials:
- Building a Resume Builder with React.js
- Creating a Simple Resume Builder with ASP.NET Core MVC

1.5.4 Design Resources:

- Explore design resources for inspiration on UI/UX patterns, typography, color schemes, and layout designs. Websites like Dribble, Bedance, and Pinterest showcase design portfolios and templates.
- Example design resources:
- Bedance Resume Design Inspiration
- Dribble Resume Templates

1.5.5 Documentation and Specifications:

- Refer to official documentation and specifications for technologies and frameworks you plan to use in your resume builder project. This includes documentation for frontend libraries (e.g., React, Angular), backend frameworks (e.g., ASP.NET Core, Django), and database systems (e.g., MongoDB, PostgreSQL).
- Example documentation:
- React Documentation
- ASP.NET Core Documentation

1.5.6 Books and Courses:

- Explore books and online courses on topics related to web development, user experience design, and resume writing. These resources can provide in-depth knowledge and practical insights.
- Example books:
  - "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
  - "Don't Make Me Think: A Common-Sense Approach to Web Usability" by Steve Krug
- Example courses:
  -Udemy: "Build Responsive Real-World Websites with HTML5 and CSS3"
  -Coursera: "Interaction Design Specialization" by University of California, San Diego.

1.5.7 Community Forums and Q&A Sites:

- Participate in community forums and Q&A sites to ask questions, seek advice, and learn from other developers' experiences. Platforms like Stack Overflow, Reddit, and GitHub Discussions are great places to connect with the developer community.

# 2. Overall Description

## 2.1 Product Perspective

The product perspective of a resume builder refers to how the application fits into the broader context of the market, user needs, and technology landscape. Here are key aspects of the product perspective for a resume builder:

2.1.1 Market Analysis:

- Identify the target market segments for the resume builder, such as job seekers, professionals, students, or specific industries.
- Analyze competitors offering similar resume building solutions, including their features, pricing models, user experience, and market positioning.
- Understand market trends, demands, and emerging technologies in the resume building industry to stay competitive and meet user expectations.

2.1.2 User Needs and Requirements:

- Conduct user research to understand the needs, preferences, and pain points of the target audience regarding resume creation and job application processes.
- Gather user requirements through surveys, interviews, usability tests, and feedback sessions to prioritize features and functionalities.
- Define user personas representing different segments of the target audience to guide product development and decision-making.

2.1.3 Integration with Existing Systems:

- Consider how the resume builder integrates with existing systems and platforms that users may already use, such as LinkedIn profiles, job boards, applicant tracking systems (ATS), or career services platforms.
- Ensure compatibility and seamless integration with third-party APIs, data formats, and authentication mechanisms to enhance user experience and productivity.

2.1.4. Scalability and Extensibility:

- Design the resume builder application to be scalable and extensible to accommodate future growth, feature enhancements, and changes in user needs.
- Architect the system with modular components, microservices architecture, or cloud-native technologies to support scalability, flexibility, and maintainability.

2.1.5 Technological Considerations:

- Evaluate technology options and frameworks for developing the resume builder application, considering factors such as development time, cost, scalability, security, and maintainability.
- Choose appropriate frontend, backend, and database technologies based on project requirements, team expertise, and industry best practices.
- Consider cross-platform compatibility, accessibility, and performance optimization strategies to ensure a seamless user experience across different devices and environments.

2.1.6 Regulatory and Compliance Requirements:

- Consider regulatory requirements and compliance standards related to data privacy, security, and accessibility (e.g., GDPR, ADA) when designing and developing the resume builder application.
- Implement appropriate data protection measures, encryption techniques, and access controls to safeguard user data and ensure compliance with relevant regulations.

2.1.7 Business Model and Monetization Strategy:

- Define the business model and monetization strategy for the resume builder application, such as subscription-based pricing, freemium models, or one-time purchases.
- Explore additional revenue streams, such as premium features, advertising, partnerships, or affiliate programs, to maximize revenue potential and sustain the business.

## 2.2 Product Features

The features of a resume builder application can vary based on the target audience, market requirements, and the scope of the project. Here's a comprehensive list of potential features for a resume builder:

2.2.1 User Registration and Authentication:

- Allow users to create accounts and log in securely to access their resumes from any device.

2.2.2 Resume Creation and Editing:

- Provide intuitive tools for users to input their personal information, education, work experience, skills, and other relevant details.
- Offer a WYSIWYG editor or drag-and-drop interface for easy editing and formatting of resume sections.
- Support rich text formatting, bullet points, lists, and multimedia content (e.g., images, videos) within the resume.

2.2.3 Resume Templates and Customization:

- Offer a variety of professionally designed templates for users to choose from, with options for different styles, layouts, and designs.
- Allow users to customize templates by adjusting colors, fonts, styles, and layout elements to match their preferences or branding.

2.2.4 Section Management:

- Provide predefined sections for common resume elements (e.g., Summary, Education, Work Experience, Skills) and allow users to add, remove, or rearrange sections as needed.
- Support custom sections for users to include additional information relevant to their specific career goals or industries.

2.2.5 Export and Download Options:

- Enable users to export their resumes in various formats (e.g., PDF, Word, HTML) for printing, emailing, or sharing online.
- Offer options to download resumes directly to the user's device or share them via email, social media, or cloud storage platforms.

2.2.6 Version Control and History:

- Allow users to save multiple versions of their resumes and track changes over time.
- Provide a history of edits, revisions, and versioning to revert to previous versions or compare changes.

2.2.7 Integration with Online Profiles and Platforms:

- Allow users to import their existing resumes or profiles from platforms like LinkedIn, GitHub, or professional networking sites.
- Provide options to export resumes to online job boards, applicant tracking systems (ATS), or career services platforms for job applications.

2.2.8 Collaboration and Sharing:

- Support collaboration features, such as sharing resumes with others for feedback or collaboration on a shared document.
- Allow users to invite collaborators, set permissions, and track changes made by multiple contributors.

2.2.9 Printing and PDF Generation:

- Enable users to print their resumes directly from the application with optimal formatting and layout for physical distribution.
- Provide options for generating high-quality PDF versions of resumes suitable for digital sharing or archiving.

2.2.10 Analytics and Insights:

- Track usage metrics, such as the number of resumes created, views, downloads, and user engagement, to gain insights into user behavior and preferences.
- Provide analytics dashboards and reports to help users understand the effectiveness of their resumes and job application strategies.

2.2.11 Mobile Responsiveness and Cross-Platform Compatibility:

- Ensure that the resume builder application is responsive and accessible on various devices and screen sizes, including desktops, laptops, tablets, and smartphones.
- Optimize the user experience for mobile users with touch-friendly controls, gestures, and navigation patterns.

2.2.12 Security and Privacy:

- Implement robust security measures to protect user data, prevent unauthorized access, and secure transactions.
- Encrypt sensitive information, use HTTPS protocols, and follow industry best practices for data protection and privacy compliance.

2.2.13 Accessibility:

- Design the application with accessibility features to ensure that users with disabilities can navigate, nteract with, and use the resume builder effectively.
- Follow accessibility standards (e.g., WCAG) for user interface design, keyboard navigation, screen reader compatibility, and color contrast ratios.

## 2.3 User Classes and Characteristics

User classes and characteristics for a resume builder encompass the various types of individuals who interact with the application and their specific needs, preferences, and behaviors. Here are some user classes and their characteristics for a resume builder:

2.3.1 Job Seekers:

- Characteristics:
  - Typically, individuals actively seeking employment or exploring new career opportunities.
  - Varied levels of experience, education, and skill sets.
  - May have limited knowledge of resume writing best practices and formatting guidelines.
- Needs:
  - Require an easy-to-use platform to create professional-looking resumes quickly.
  - Seek guidance on content selection, formatting, and customization options.
  - Value access to templates, examples, and tips for improving their resumes.

2.3.2 Professionals:

- Characteristics:
  - Working professionals looking to update their resumes for job applications, promotions, or networking purposes.
  - Often have specific industry experience, certifications, or achievements to highlight.
  - May require customization options to tailor their resumes for different job opportunities.
- Needs:
  - Desire flexibility in editing and formatting their resumes to showcase relevant skills and experiences.
  - Value access to advanced features, such as importing LinkedIn profiles, integrating with job boards, or tracking application status.

2.3.3 Students and Recent Graduates:

- Characteristics:
  - Students or recent graduates entering the job market for the first time.
  - Limited professional experience but may possess relevant academic achievements, internships, or extracurricular activities.
  - Often lack familiarity with resume writing conventions and industry standards.
- Needs:
  - Require guidance on structuring their resumes to highlight academic achievements, projects, and relevant skills.
  - Value access to entry-level resume templates, career advice, and resources for transitioning into the workforce.

2.3.4 Career Changers:

- Characteristics:
  - Individuals seeking to transition into a new industry or role.
  - Have transferable skills and experiences from previous roles but may lack direct experience in the desired field.
  - Require assistance in crafting a compelling narrative to showcase their transferable skills and qualifications.
- Needs:
  - Need tools and guidance to emphasize relevant experiences and skills that are transferable to the new career path.
  - Value access to customizable templates, career change guides, and industry-specific resume examples.

### 2.3.5 Recruiters and Hiring Managers:

- Characteristics:
  - Professionals responsible for evaluating and shortlisting candidates for job openings.
  - Require well-formatted, easy-to-read resumes that highlight relevant qualifications and experiences.
  - Often review multiple resumes in a short amount of time and prioritize those that meet specific criteria.
- Needs:
  - Seek access to clear, concise resumes that effectively communicate candidates' qualifications and fit for the position.
  - Value standardized formats, keyword optimization, and consistent presentation of information across resumes.

## 2.4 Operating Environment

The operating environment of a resume builder refers to the technical infrastructure and conditions required for the application to function effectively. Here's an overview of the operating environment for a resume builder:

2.4.1 Web-Based Application:

- The resume builder is typically a web-based application accessible through a web browser.
- It should be compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- The application should be responsive and optimized for various screen sizes, including desktops, laptops, tablets, and smartphones.

2.4.2 Backend Technologies:

- The backend of the resume builder may be built using server-side technologies such as:
- Programming languages/frameworks like Node.js, Python (Django), Ruby on Rails, or ASP.NET Core.
- Database management systems (DBMS) like MySQL, PostgreSQL, MongoDB, or SQL Server for storing user data, resumes, and application settings.
- Server infrastructure to host the application, including cloud platforms like AWS, Microsoft Azure, Google Cloud Platform, or self-hosted servers.

2.4.3 Frontend Technologies:

- The frontend of the resume builder is developed using client-side technologies such as:
- HTML, CSS, and JavaScript for building the user interface (UI) and implementing interactive features.
- Frontend frameworks/libraries like React.js, Angular, Vue.js, or Svelte.js for building dynamic and responsive UI components.
- Design libraries or frameworks such as Bootstrap, Material Design, or Tailwind CSS for styling and layout consistency.

2.4.4 Dependencies and Libraries:

- The resume builder may rely on third-party libraries, plugins, or APIs for additional functionalities such as:
- PDF generation libraries (e.g., Puppeteer, wkhtmltopdf) to export resumes to PDF format.
- Authentication and authorization services (e.g., OAuth, JWT) for user authentication and access control.
- Integration with external services (e.g., LinkedIn API, job boards API) for importing/exporting resume data or integrating with external platforms.

2.4.5 Security Considerations:

- Implement security measures to protect user data, prevent unauthorized access, and ensure data privacy:
- Use HTTPS protocol to encrypt data transmission between the client and server.
- Implement authentication mechanisms (e.g., username/password, social login) to verify user identities.
- Apply authorization controls to restrict access to sensitive features or user data based on user roles and permissions.
- Sanitize user inputs and implement measures to prevent common web security vulnerabilities such as cross-site scripting (XSS), SQL injection, and CSRF attacks.

2.4.6 Performance and Scalability:

- Optimize the application for performance and scalability to handle concurrent user requests and growing user base:
- Use caching mechanisms (e.g., Redis, Memcached) to improve response times and reduce database load.
- Implement horizontal scaling strategies such as load balancing and auto-scaling to distribute traffic and resources across multiple server instances.
- Monitor application performance metrics (e.g., response time, throughput, error rates) using tools like Prometheus, Grafana, or New Relic, and optimize as needed.

2.4.7 Development and Testing Environment:

- Set up development and testing environments with tools and resources for coding, debugging, and testing the resume builder application:
- Integrated development environments (IDEs) such as Visual Studio Code, JetBrains IntelliJ IDEA, or Eclipse for coding and debugging.
- Version control systems (e.g., Git, SVN) for collaborative development and source code management.
- Automated testing frameworks (e.g., Jest, Selenium, JUnit) for unit testing, integration testing, and end-to-end testing.

## 2.5 Design and Implementation Constraints

Design and implementation constraints for a resume builder refer to limitations or restrictions that affect how the application is designed, developed, and deployed. Here are some common constraints you may encounter:

2.5.1 Technology Stack:

- Constraints related to the technology stack may include:
  - Compatibility requirements with existing systems or platforms.
  - Restrictions imposed by organizational policies or guidelines regarding the use of specific programming languages, frameworks, or libraries.
  - Limitations in available resources (e.g., expertise, budget) for adopting certain technologies or tools.

2.5.2 User Interface (UI) Constraints:

- Design constraints may include:
  - Compliance with accessibility standards (e.g., WCAG) to ensure that the application is usable by individuals with disabilities.
  - Consistency with branding guidelines or design specifications set by the organization or client.
  - Adherence to responsive design principles to ensure that the application is usable across various screen sizes and devices.

2.5.3 Data Privacy and Security:

- Constraints related to data privacy and security may include:
  - Compliance with data protection regulations (e.g., GDPR, HIPAA) governing the collection, storage, and processing of personal information.
  - Implementation of encryption, access controls, and other security measures to protect sensitive user data from unauthorized access or breaches.
  - Limitations in the use of third-party services or APIs that may pose security risks or require stringent compliance requirements.

2.5.4 Performance and Scalability:

- Constraints related to performance and scalability may include:
  - Limitations in hardware resources (e.g., server capacity, memory, bandwidth) that may affect the application's ability to handle concurrent users or large volumes of data.
  - Optimization challenges in improving application performance, such as reducing latency, optimizing database queries, or minimizing resource usage.
  - Scalability constraints that may arise from architectural decisions or dependencies on external services, leading to bottlenecks or limitations in scaling horizontally or vertically.

2.5.5 Regulatory and Compliance Requirements:

- Constraints related to regulatory and compliance requirements may include:
  - Legal restrictions or industry-specific regulations that dictate how certain features or functionalities should be implemented (e.g., financial regulations for handling payment information).
  - Certification or accreditation requirements that the application must meet to operate in specific sectors or markets (e.g., healthcare, finance, government).

2.5.6 Integration and Interoperability:

- Constraints related to integration and interoperability may include:
  - Compatibility issues with legacy systems or external APIs that the application needs to integrate with.
  - Dependencies on proprietary formats or protocols that require custom integration solutions or data transformation processes.
  - Challenges in ensuring seamless communication and data exchange between different systems, platforms, or applications.

2.5.7 Budget and Resource Constraints:

- Constraints related to budget and resources may include:
  - Limitations in funding or budget allocations for development, maintenance, and support of the application.
  - Constraints in staffing, expertise, or availability of skilled resources needed to design, develop, and deploy the application.
  - Trade-offs between features, quality, and time-to-market due to resource limitations or competing priorities.

## 2.6 User Documentation

User documentation for a resume builder is essential to guide users through the process of creating, editing, and managing their resumes effectively. Here's an outline of what user documentation for a resume builder might include:

2.6.1 Getting Started Guide:

- Introduction to the resume builder application.
- Overview of key features and capabilities.
- Instructions for accessing the application (e.g., web address, mobile app download).
- Guidance on creating a new account or logging in.

2.6.2 User Interface Overview:

- Description of the user interface layout and navigation elements.
- Explanation of each component and its purpose (e.g., toolbar, sidebar, main content area).
- Visual illustrations or screenshots highlighting important UI elements.

2.6.3 Creating a Resume:

- Step-by-step instructions for creating a new resume.
- Guidance on entering personal information, contact details, and career objectives.
- Instructions for adding sections (e.g., education, work experience, skills) and filling in relevant details.
- Tips for selecting appropriate resume templates, layouts, and designs.

2.6.4 Editing and Formatting:

- Instructions for editing and formatting resume content.
- Guidance on adding, deleting, or rearranging sections and subsections.
- Tips for formatting text (e.g., font styles, sizes, colors) and adjusting layout elements.

2.6.5 Saving and Version Control:

- Explanation of how to save and manage resumes within the application.
- Instructions for saving drafts, creating multiple versions, and tracking changes over time.
- Guidance on naming conventions, organizing folders, and managing file versions.

2.6.6 Exporting and Sharing:
- Instructions for exporting resumes in different formats (e.g., PDF, Word, HTML).
- Guidance on downloading, printing, or sharing resumes electronically.
- Tips for optimizing resume formatting for different export formats.

2.6.7 Advanced Features:
- Documentation for advanced features, such as importing LinkedIn profiles, integrating with job boards, or collaborating with others.
- Step-by-step instructions for using each feature and maximizing its benefits.

2.6.8 Troubleshooting and FAQs:
- Common troubleshooting tips for resolving issues encountered while using the resume builder.
- Frequently asked questions (FAQs) addressing common user inquiries and concerns.
- Contact information for customer support or technical assistance.

2.6.9 Best Practices and Tips:
- Best practices for creating effective resumes, including content selection, structure, and formatting guidelines.
- Tips for tailoring resumes to specific industries, job roles, or career stages.
- Advice on maximizing the impact of resumes for job applications and networking opportunities.

2.6.10 Glossary and Terminology:
- Definitions of key terms, acronyms, and technical jargon used within the resume builder application.
- Clarification of industry-specific terminology related to resume writing and job search.

2.6.11 Legal and Privacy Information:
- Terms of service, privacy policy, and user consent statements outlining data collection, storage, and usage policies.
- Information about user rights, data protection measures, and compliance with relevant regulations (e.g., GDPR, CCPA).

2.6.12 Updates and Release Notes:
- Notifications of application updates, new features, and enhancements.
- Release notes documenting changes, bug fixes, and improvements in each software release.

## 2.7 Assumptions and Dependencies

Assumptions and dependencies are critical considerations when developing a resume builder application. They help to define the boundaries of the project and identify factors that may impact its development and operation. Here are some assumptions and dependencies commonly associated with a resume builder:

2.7.1 Assumptions:
- User Familiarity: Assumes that users have basic computer skills and familiarity with web-based applications.
- Internet Connectivity: Assumes that users have reliable internet access to use the online resume builder.
- Device Compatibility: Assumes that the application is compatible with common web browsers and devices, such as desktops, laptops, tablets, and smartphones.

- Data Privacy: Assumes that the application complies with relevant data privacy regulations and safeguards user data.
- Resume Content Accuracy: Assumes that users provide accurate and truthful information when creating their resumes.
- Template Effectiveness: Assumes that the resume templates provided by the application are effective in helping users create professional resumes.

2.7.2 Dependencies:

- Third-Party Libraries and APIs: Dependencies on external libraries, frameworks, or APIs for features such as PDF generation, authentication, and data storage.
- Development Tools and Platforms: Dependencies on development tools, platforms, and environments (e.g., IDEs, version control systems, hosting providers).
- Database Systems: Dependencies on specific database systems (e.g., MySQL, PostgreSQL, MongoDB) for storing user data, resumes, and application settings.
- Web Technologies: Dependencies on frontend and backend technologies (e.g., HTML, CSS, JavaScript, Node.js, ASP.NET Core) for building and running the application.
- Network Infrastructure: Dependencies on network infrastructure, including servers, routers, and internet service providers, to host and deliver the application.
- Compliance Regulations: Dependencies on compliance regulations (e.g., GDPR, CCPA) that govern the collection, storage, and processing of user data.
- User Feedback and Support: Dependencies on user feedback and support channels (e.g., customer support, feedback forms) for addressing user inquiries, issues, and feature requests.
- Continuous Integration/Continuous Deployment (CI/CD): Dependencies on CI/CD pipelines and deployment processes for releasing updates, patches, and new features to the production environment.

# 3. System Features

The system features of a resume builder encompass the functionalities and capabilities that enable users to create, edit, manage, and share their resumes effectively. Here are key system features typically found in a resume builder:

- User Authentication and Account Management:
  - User registration and login functionality to create and access accounts.
  - User profile management to update personal information, contact details, and account settings.

- Resume Creation and Editing:
  - Intuitive interface for creating and editing resumes with a variety of sections (e.g., Contact Information, Objective, Education, Work Experience, Skills).
  - Customizable templates and layouts to choose from, allowing users to personalize the appearance of their resumes.
  - Ability to add, remove, or rearrange sections and subsections to tailor the resume structure to individual preferences.

- Content Input and Formatting:
  - Input fields and text editors for entering and formatting resume content, including text, lists, and bullet points.
  - Rich text editing features such as font styles, sizes, colors, alignments, and formatting options (e.g., bold, italic, underline). Support for uploading and embedding media files (e.g., images, documents) to enhance resume content.

- Preview and Review:

  - Preview mode to visualize how the resume will appear to potential employers.
  - Review functionality to check for errors, spelling mistakes, formatting inconsistencies, and content clarity.
  - Real-time feedback and suggestions for improving resume content and presentation.

- Saving and Version Control:

  - Auto-save functionality to periodically save resume drafts and changes.
  - Version control features to manage multiple versions of resumes, allowing users to revert to previous versions if needed.
  - Options to save resumes locally or in the cloud for accessibility across devices and locations.

- Export and Download:
  - Export functionality to generate resumes in various formats (e.g., PDF, Word, HTML) for printing or sharing electronically.
  - Download options for saving resumes to local storage or cloud storage services (e.g., Google Drive, Dropbox).
  - Integration with third-party platforms (e.g., LinkedIn, job boards) for seamless resume sharing and application submission.

- Collaboration and Sharing:
  - Collaboration features to invite others (e.g., mentors, peers) to review and provide feedback on resumes.
  - Sharing functionality to share resumes via email, social media, or direct links with recruiters, employers, or networking contacts.

- Analytics and Insights:
  - Analytics dashboard to track resume views, downloads, and interactions.
  - Insights and recommendations based on user activity and performance metrics to improve resume effectiveness.

- Help and Support:
  - Help documentation and tutorials to guide users through the resume creation process and application features.
  - Customer support channels (e.g., chat support, email support) for assistance with technical issues, inquiries, and feedback.

- Accessibility and Localization:
  - Accessibility features to ensure the application is usable by individuals with disabilities (e.g., screen readers, keyboard navigation).
  - Support for multiple languages and localization options to cater to users from different regions and language preferences.

## 3.1 User Interfaces

The user interfaces (UIs) of a resume builder play a crucial role in facilitating an intuitive and efficient user experience. Here are the key user interfaces typically found in a resume builder:

- Registration/Login Interface
- Dashboard Interface
- Resume Editor Interface
- Template Selection Interface
- Preview Interface
- Settings Interface
- Collaboration Interface
- Export/Download Interface
- Help and Support Interface

## 3.2 Hardware Interfaces

As a software application primarily accessed through web browsers or mobile devices, a resume builder typically doesn't have direct hardware interfaces in the traditional sense. However, there are some hardware-related considerations and interfaces that can affect its usability and performance:

- Input Devices
  - Keyboards
  - Mice or Touchpads
  - Touchscreens
- Display Devices
  - Monitors
  - Printers
  - Mobile Devices
- Storage Devices
  - Hard Drives and Solid-State Drives (SSDs)
  - Cloud Storage
- Network Interfaces
  - Ethernet/Wi-Fi
  - Mobile Data
- Printing Devices
  - Printers
  - Print Servers
- Accessibility Devices
  - Screen Readers
  - Braille Displays

## 3.3 Software Interfaces

Software interfaces of a resume builder refer to the interactions and integrations between the resume builder application and other software components, services, or platforms. Here are some common software interfaces associated with a resume builder:

- Web Browser
- Operating System
- Backend Services and APIs
- Authentication APIs
- Database APIs
- File Storage APIs
- PDF Generation APIs
- Third-Party Integrations
- Social Media APIs
- Job Board APIs
- Analytics Services
- Content Management Systems (CMS)
- Document Processing Tools

## 3.4 Communications Interfaces

Communications interfaces of a resume builder facilitate communication between different components of the application, as well as interactions with external systems or services. Here are some common communications interfaces associated with a resume builder:

- Client-Server Communication:
  - HTTP/HTTPS: The primary protocol for communication between the client (web browser or mobile app) and the server hosting the resume builder application.
  - RESTful APIs: Representational State Transfer (REST) APIs are commonly used for client-server communication, allowing the client to make requests to the server to perform operations such as creating, editing, and retrieving resumes.

- User Interface (UI) Interactions:
  - Event-driven communication: Interaction between the user interface elements (e.g., buttons, forms) and the application logic, triggering actions such as submitting form data or navigating between pages.
  - Asynchronous updates: Real-time updates to the UI in response to user actions or server-side events, providing feedback and improving responsiveness.

- Authentication and Authorization:
  - Authentication protocols: Communication between the client and server to authenticate users, typically using mechanisms such as username/password, OAuth, or JSON Web Tokens (JWT).
  - Authorization headers: Transmission of authorization tokens or credentials with each request to the server to verify the user's identity and permissions.

- Data Transfer and Storage:
  - Data exchange formats: Communication between the client and server using standardized data formats such as JSON (JavaScript Object Notation) or XML (eXtensible Markup Language) for transmitting resume data and metadata.
  - Database queries: Communication between the application server and the database management system (DBMS) to perform CRUD (Create, Read, Update, Delete) operations on resume data stored in the database.

External Service Integration:
- API calls: Communication with external APIs (e.g., social media platforms, job boards) to import or export data, share resumes, or retrieve additional information relevant to the resume building process.
- Webhooks: Communication with external services via webhooks to receive real-time notifications or trigger actions based on specific events (e.g., resume submission, job application).

Error Handling and Logging:
- Error messages: Communication of error messages or status codes from the server to the client to indicate issues with the resume building process (e.g., validation errors, server-side exceptions).
- Logging mechanisms: Communication of log messages from the application server to logging services or files for monitoring, debugging, and troubleshooting purposes.

User Feedback and Support:
- Feedback forms: Communication between the client and server to submit user feedback, bug reports, or feature requests to improve the resume builder application.
- Support chat or messaging: Real-time communication channels (e.g., chat widgets, messaging APIs) for users to seek assistance from customer support representatives or technical experts.

# 4. Other Nonfunctional Requirements

Nonfunctional requirements define the quality attributes or constraints that a system must meet beyond its functional capabilities. For a resume builder, several nonfunctional requirements are crucial for ensuring the usability, performance, security, and maintainability of the application. Here are some examples:

- Usability
- Performance
- Security
- Reliability
- Maintainability
- Compatibility
- Performance Metrics
- Deployment and Operations

## 4.1 Performance Requirements

Performance requirements for a resume builder application are essential to ensure that it meets user expectations regarding speed, responsiveness, and scalability. Here are some performance requirements typically associated with a resume builder:

- Response Time
- Page Load Time
- Concurrency and Scalability
- Database Performance
- Caching Mechanisms
- Network Latency
- Error Handling and Recovery
- Third-Party Integrations
- Scalability Testing
- Monitoring and Optimization

## 4.2 Safety Requirements

Safety requirements for a resume builder primarily focus on protecting user data, ensuring secure interactions within the application, and safeguarding against potential security threats. Here are some safety requirements that should be considered:

- Data Encryption
- Secure Authentication
- Authorization Controls
- Data Privacy Compliance
- Protection Against Injection Attacks
- Session Management
- Secure File Handling
- Security Testing
- Incident Response Plan
- User Education and Awareness

## 4.3 Security Requirements

Security requirements for a resume builder are crucial to protect user data, prevent unauthorized access, and ensure the integrity and confidentiality of information. Here are some security requirements that should be considered:

- Authentication and Authorization
- Data Encryption
- Secure Storage and Handling of User Data
- Input Validation and Sanitization
- Protection Against Common Vulnerabilities
- Session Management
- Logging and Monitoring
- Incident Response Plan
- Third-Party Integrations
- User Education and Awareness

### 4.4 Software Quality Attributes

- Usability
- Performance
- Security
- Reliability
- Maintainability
- Compatibility
- Performance Metrics
- Deployment and Operations

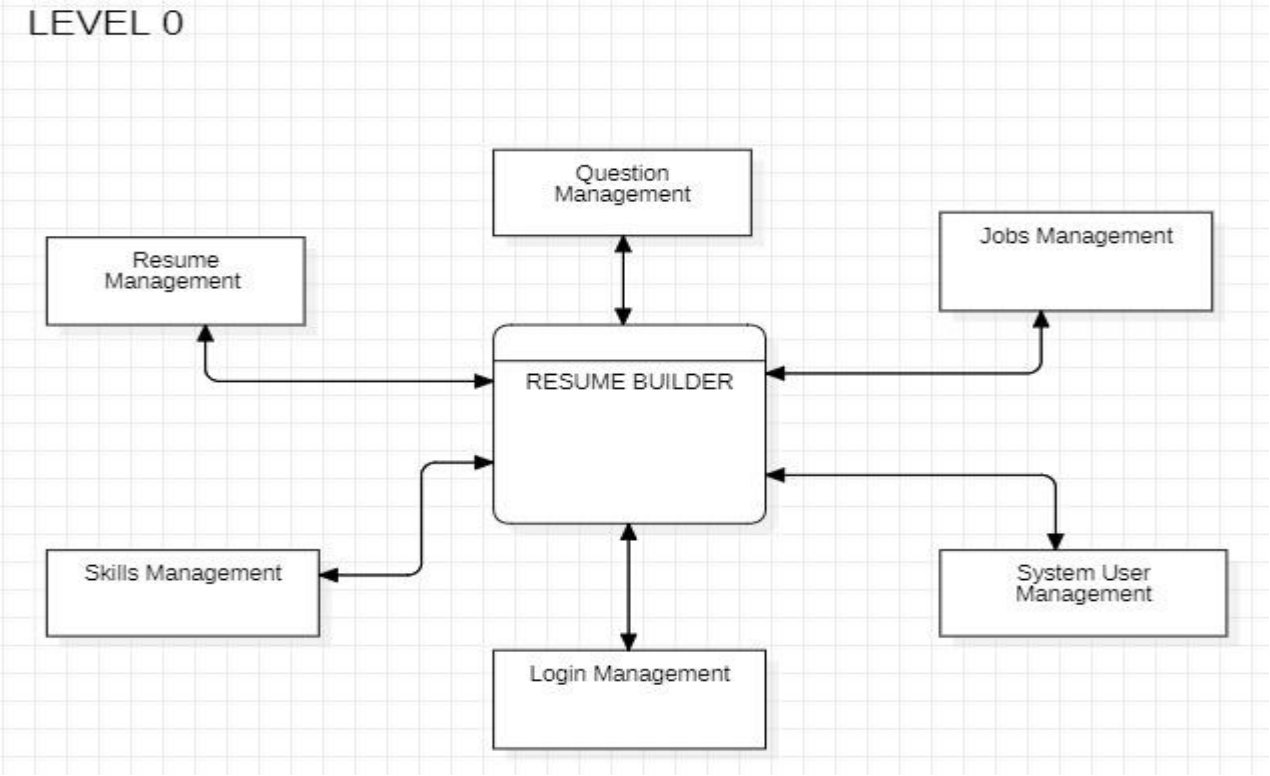
# 5. Other Requirements

- Usability Requirements
- Compatibility Requirements
- Performance Requirements
- User Experience (UX) Requirements
- Integration Requirements
- Documentation Requirements
- Support and Maintenance Requirements

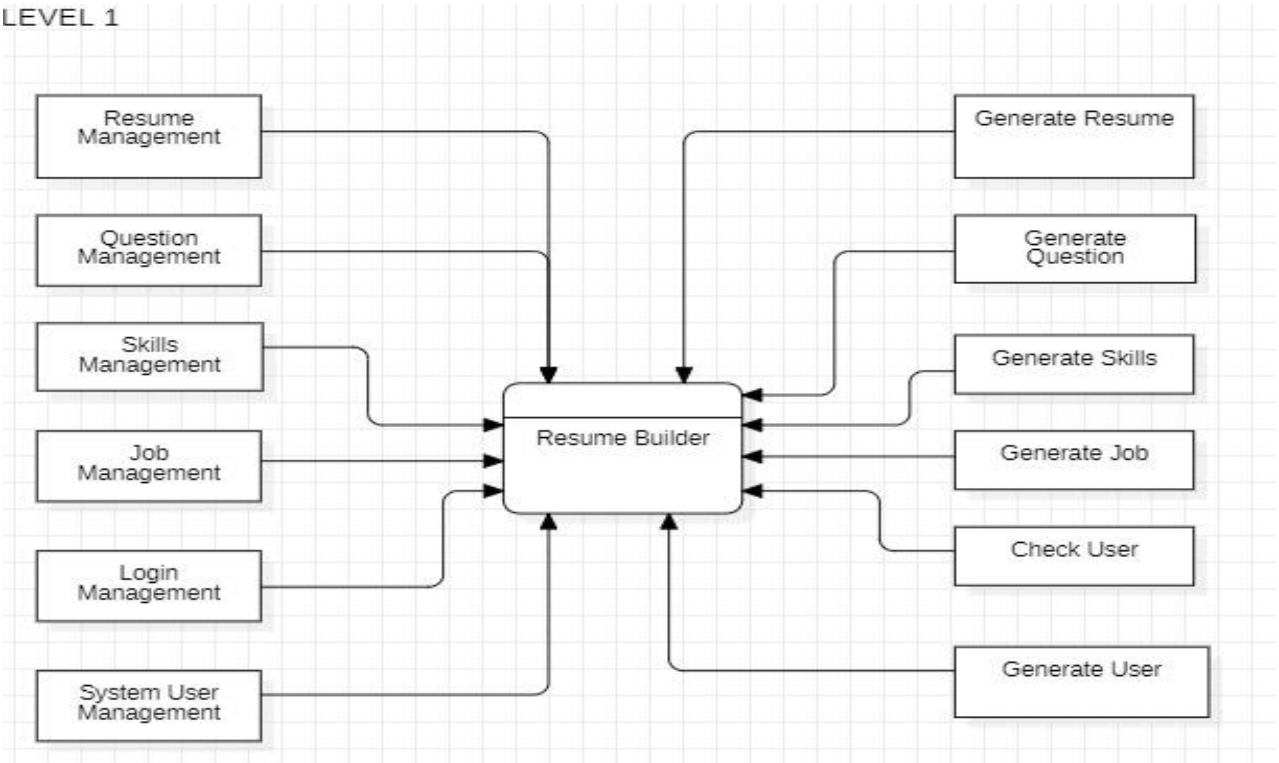
# Appendix A: Glossary

- Resume Builder
- Template
- Section
- Contact Information
- Objective Statement
- Work Experience
- Education
- Skills
- Certifications
- References
- Formatting
- Export
- Preview
- Edit
- Save

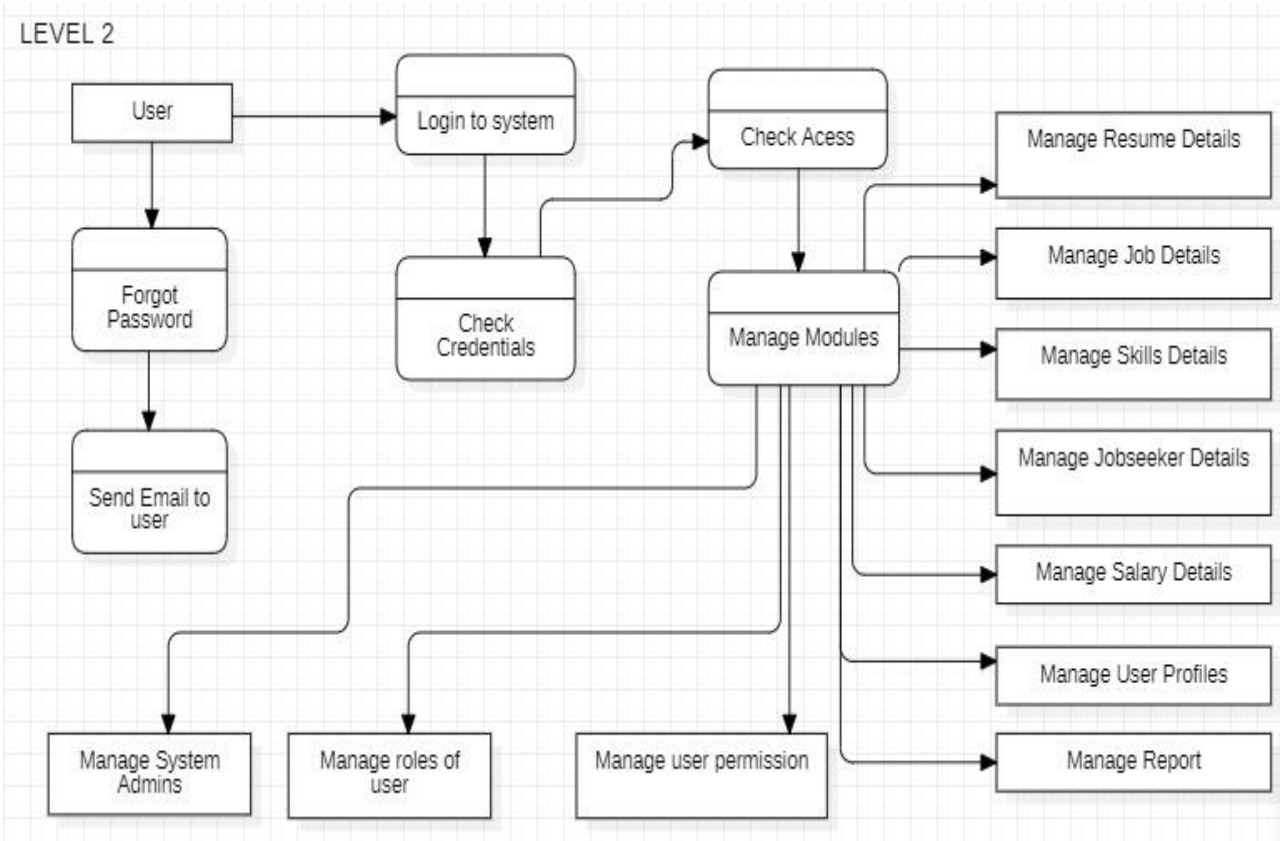
# Appendix B: Analysis Models



LEVEL 1

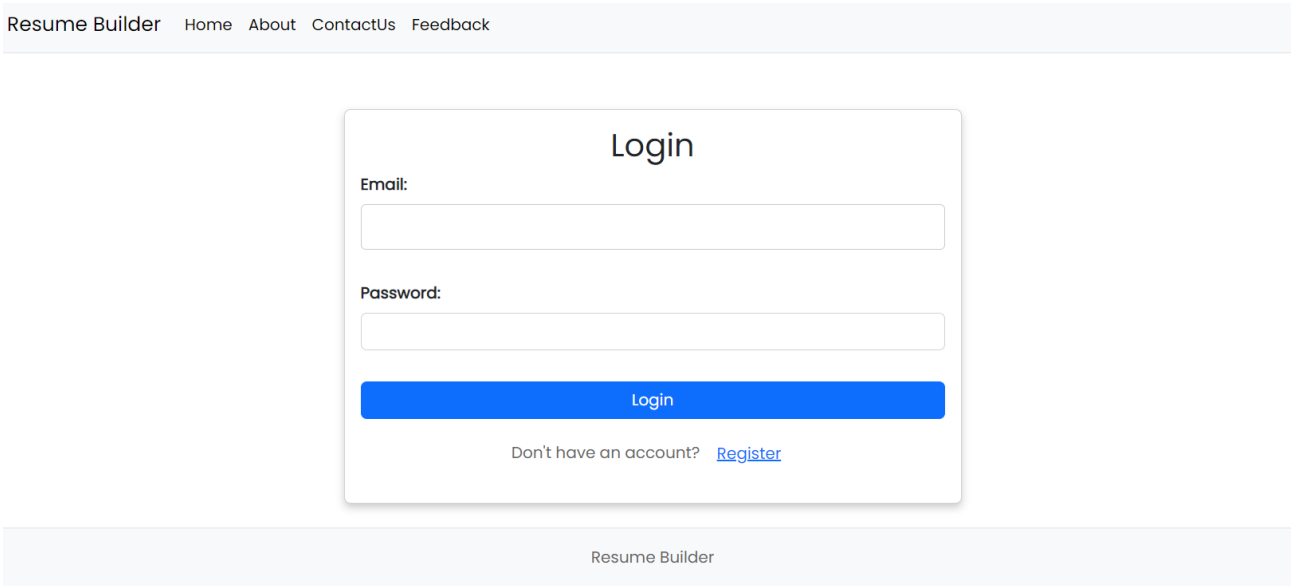



LEVEL 2



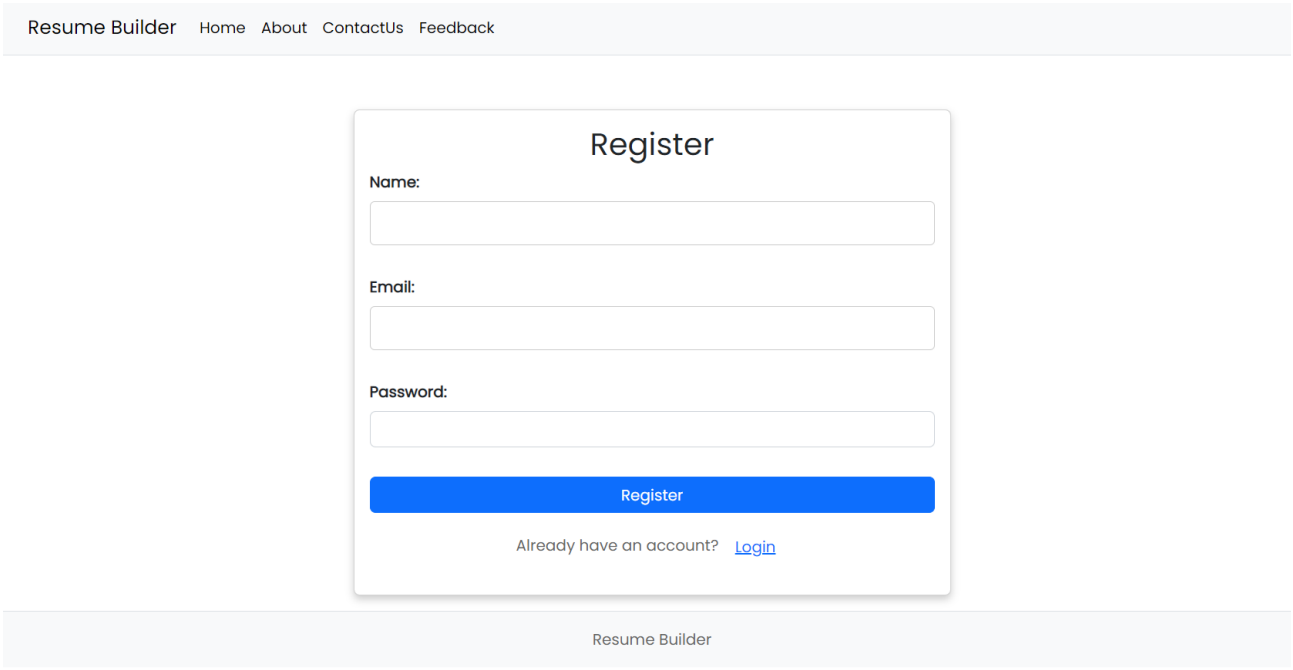
## Short Description:

A login page is a web interface or application screen that serves as the entry point for users to access a secure system or platform. Typically, users are required to input their credentials, such as a email and password, to authenticate and gain access to the protected area. The login page plays a crucial role in ensuring the security of user accounts and sensitive information by verifying the identity of individuals before granting them access.



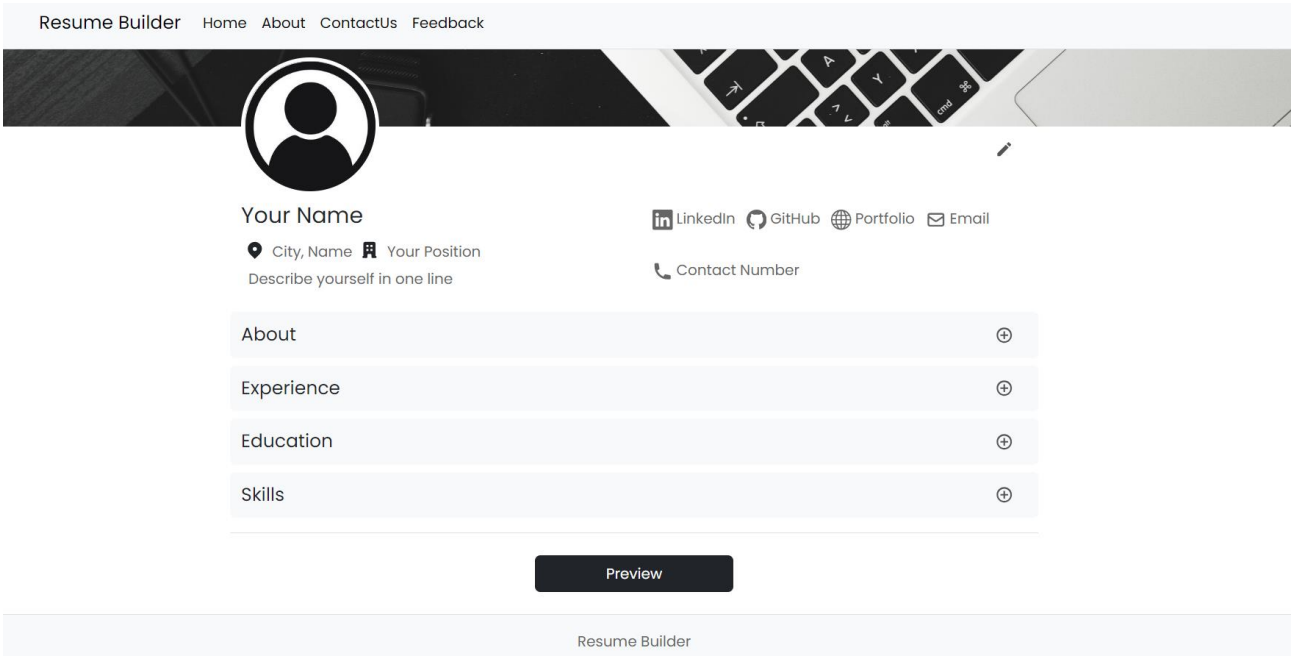
## Short Description:

A register page, also known as a signup page, is a web interface or application screen were new users can create accounts to gain access to a specific platform or service. It typically involves filling out a registration form with required information, which may include a name, email, password, and other relevant details. The purpose of a register page is to collect and validate user information, enabling the creation of a unique account for each individual.



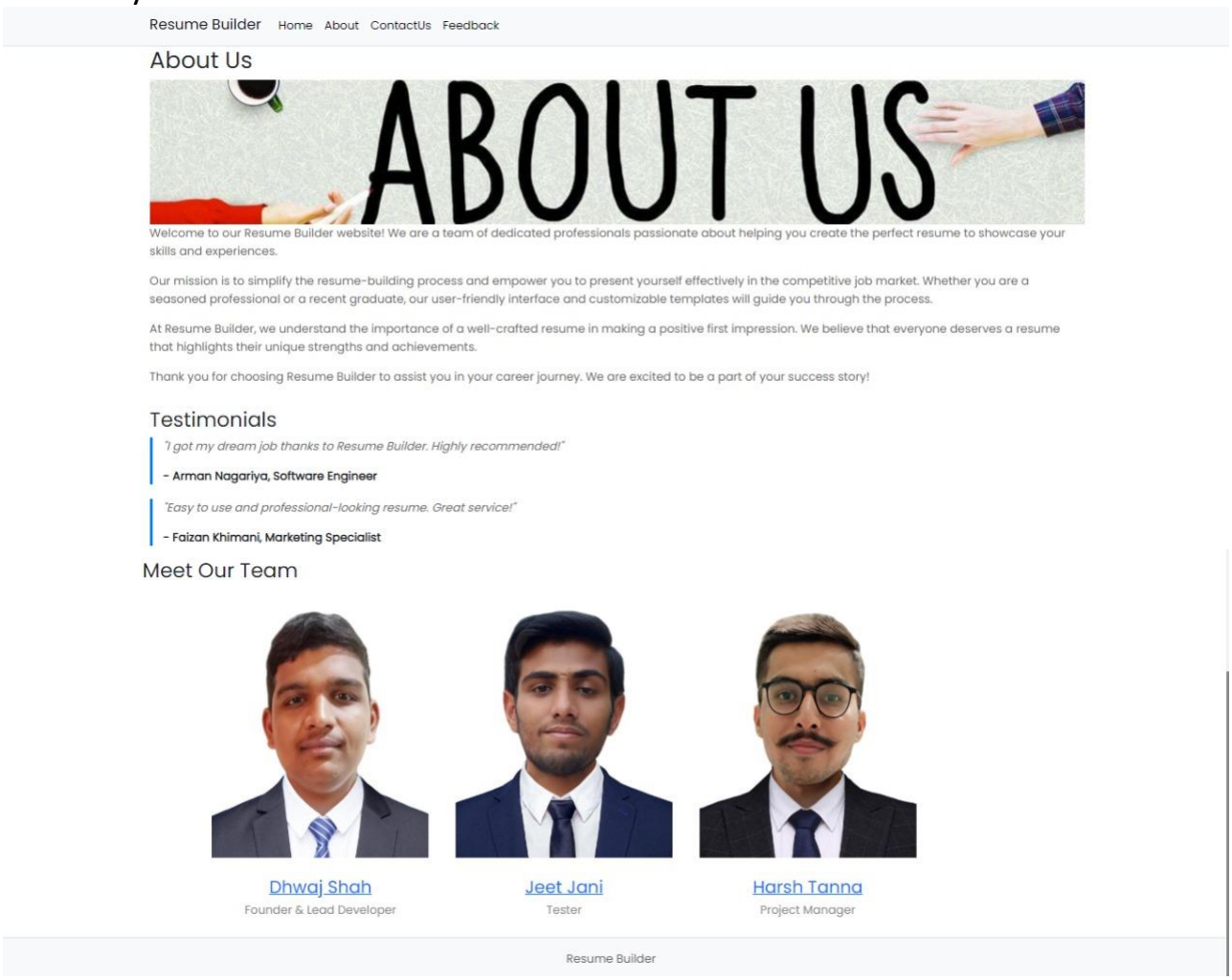
**Short Description:**

The home page of a website serves as the main entry point and often sets the tone

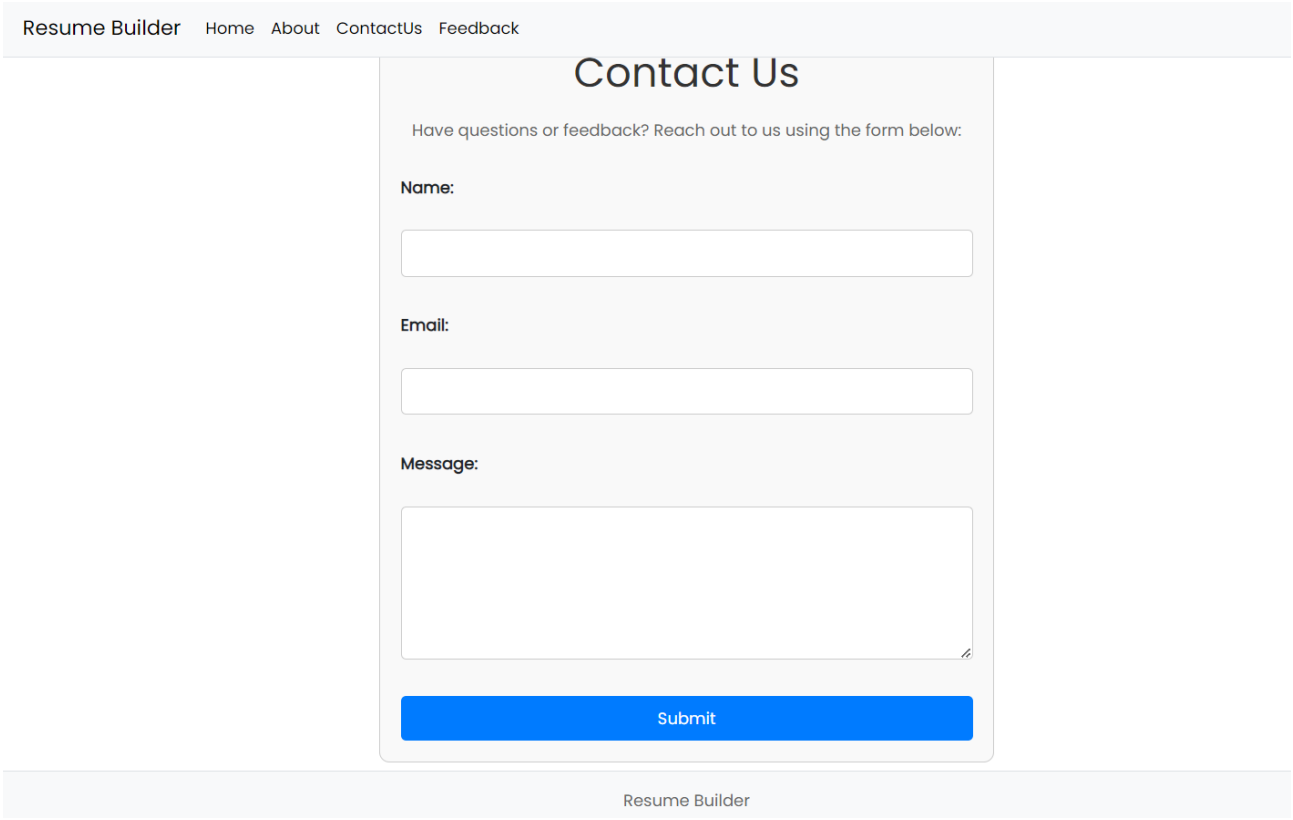


**Short Description:**

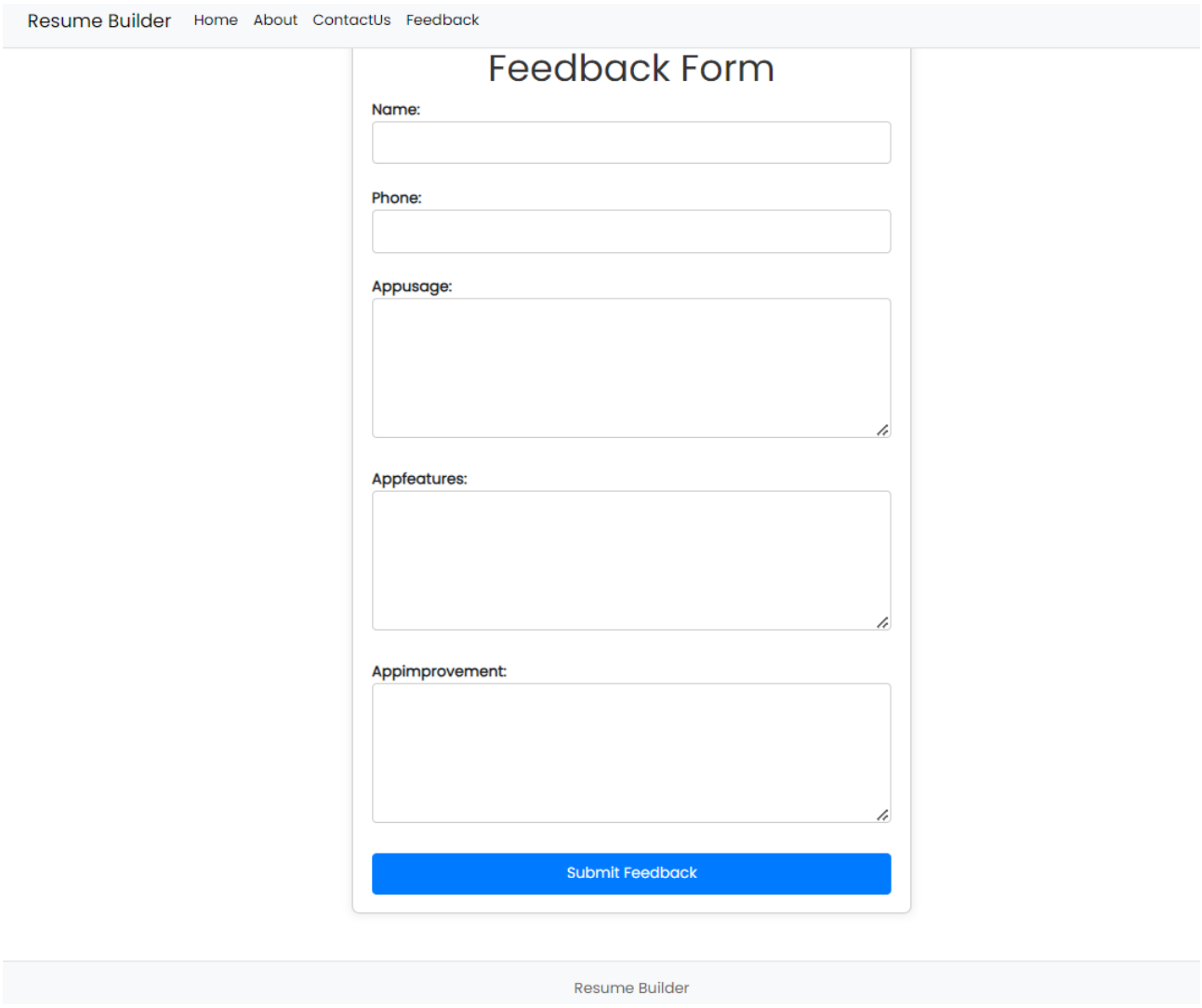
An "About" page on a website provides visitors with information about the organization, company, or individual behind the site. It typically includes details that help establish credibility, build trust, and give users a better understanding of the entity.



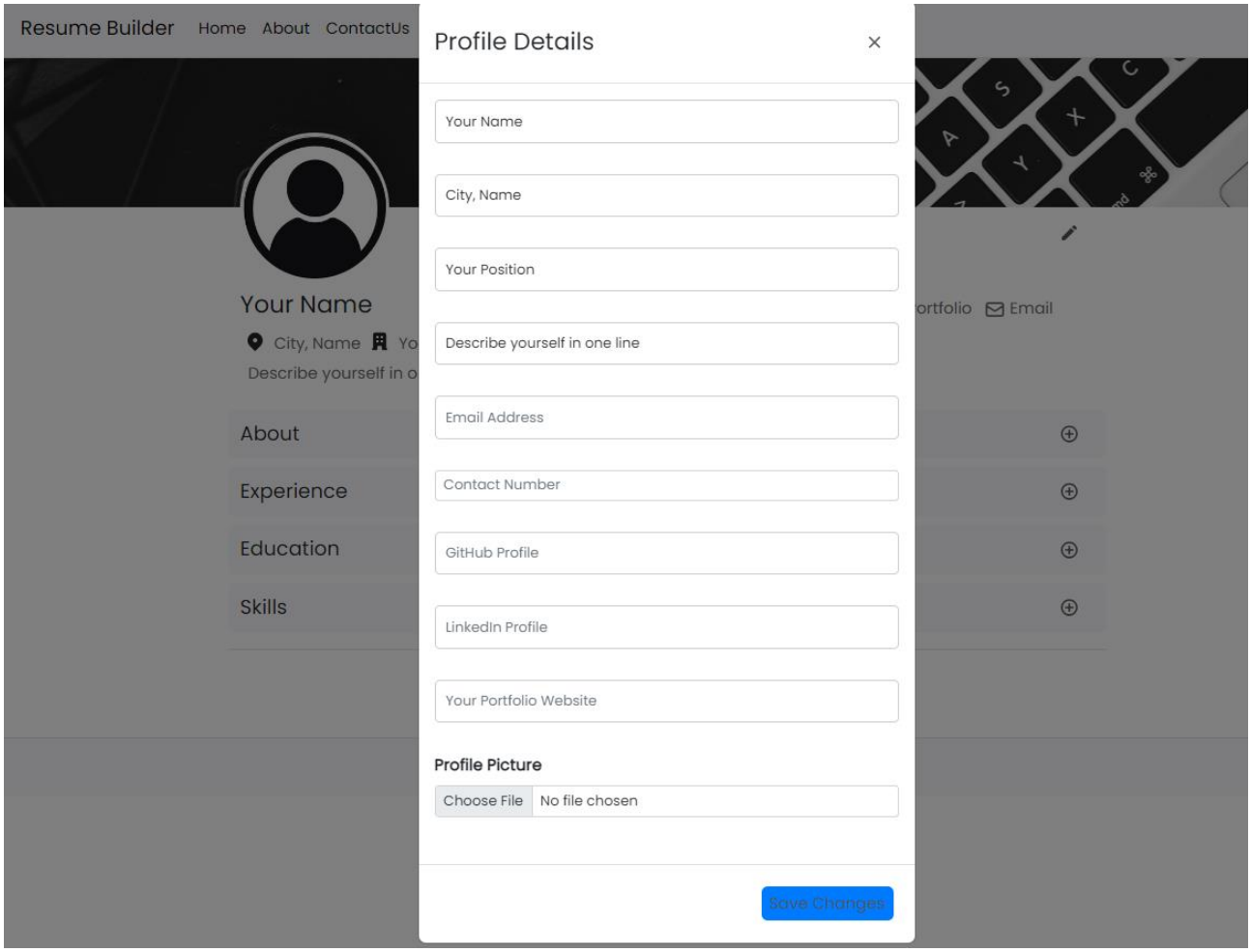
**Short Description:**
A "Contact" page on a website is a crucial component that facilitates communication between the website owner and its visitors or customers**.**



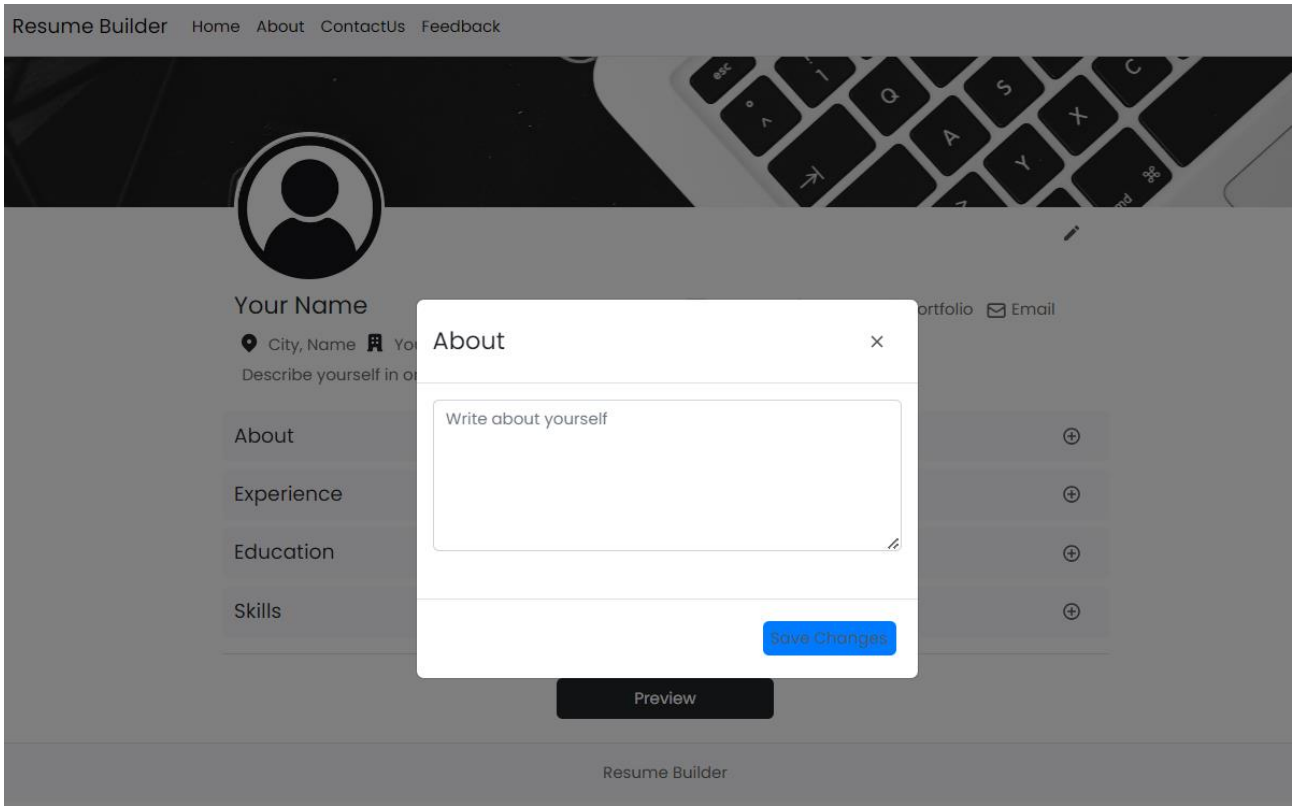
**Short Description:**
A "Feedback" page on a website is a crucial component that facilitates communication between the website owner and gives feedback to improve the platform.



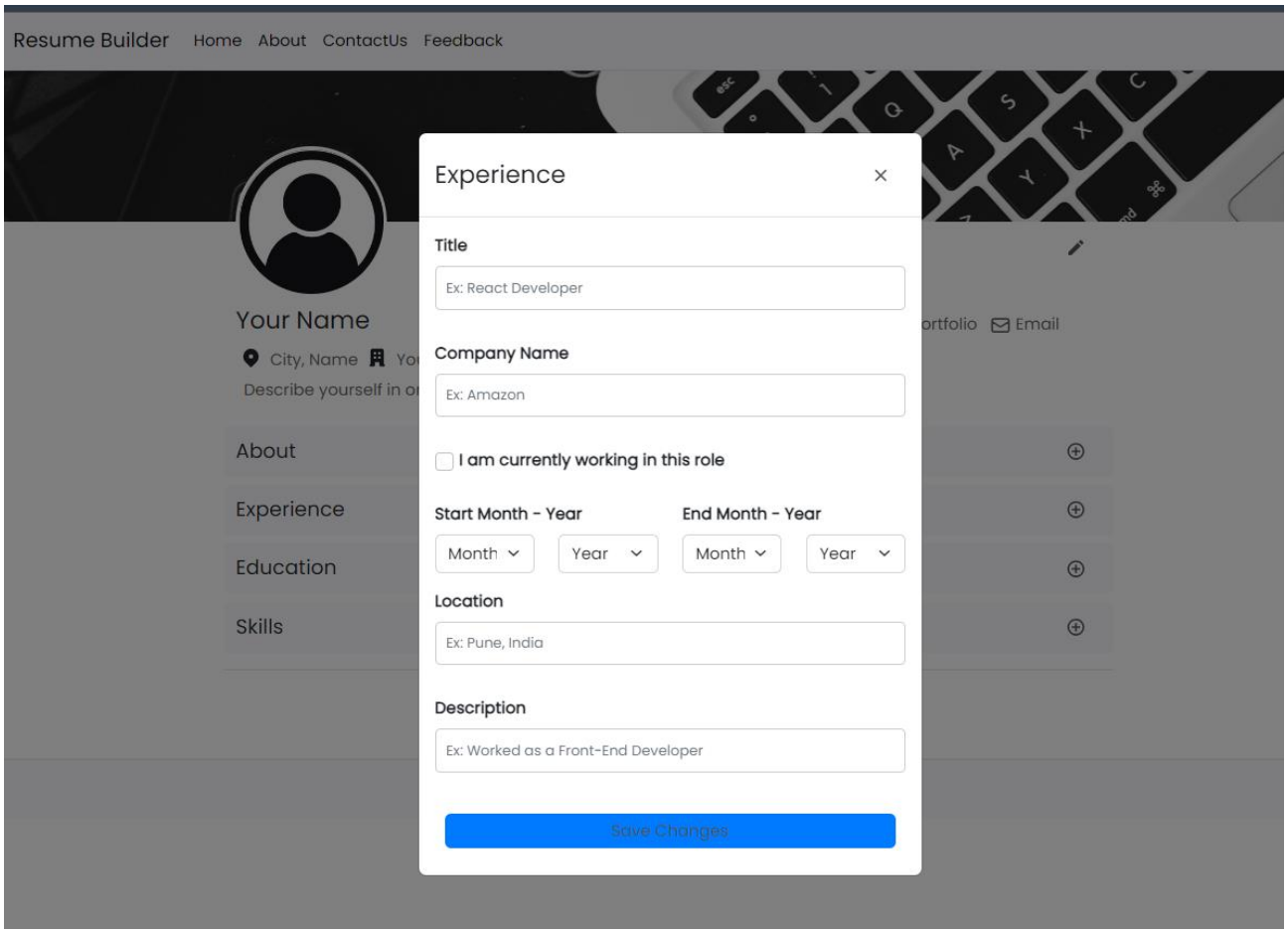
**Short Description:**
A candidate profile provides a concise summary of an individual's qualifications, skills, and professional background. It is commonly used in the context of job applications or professional networking platforms.



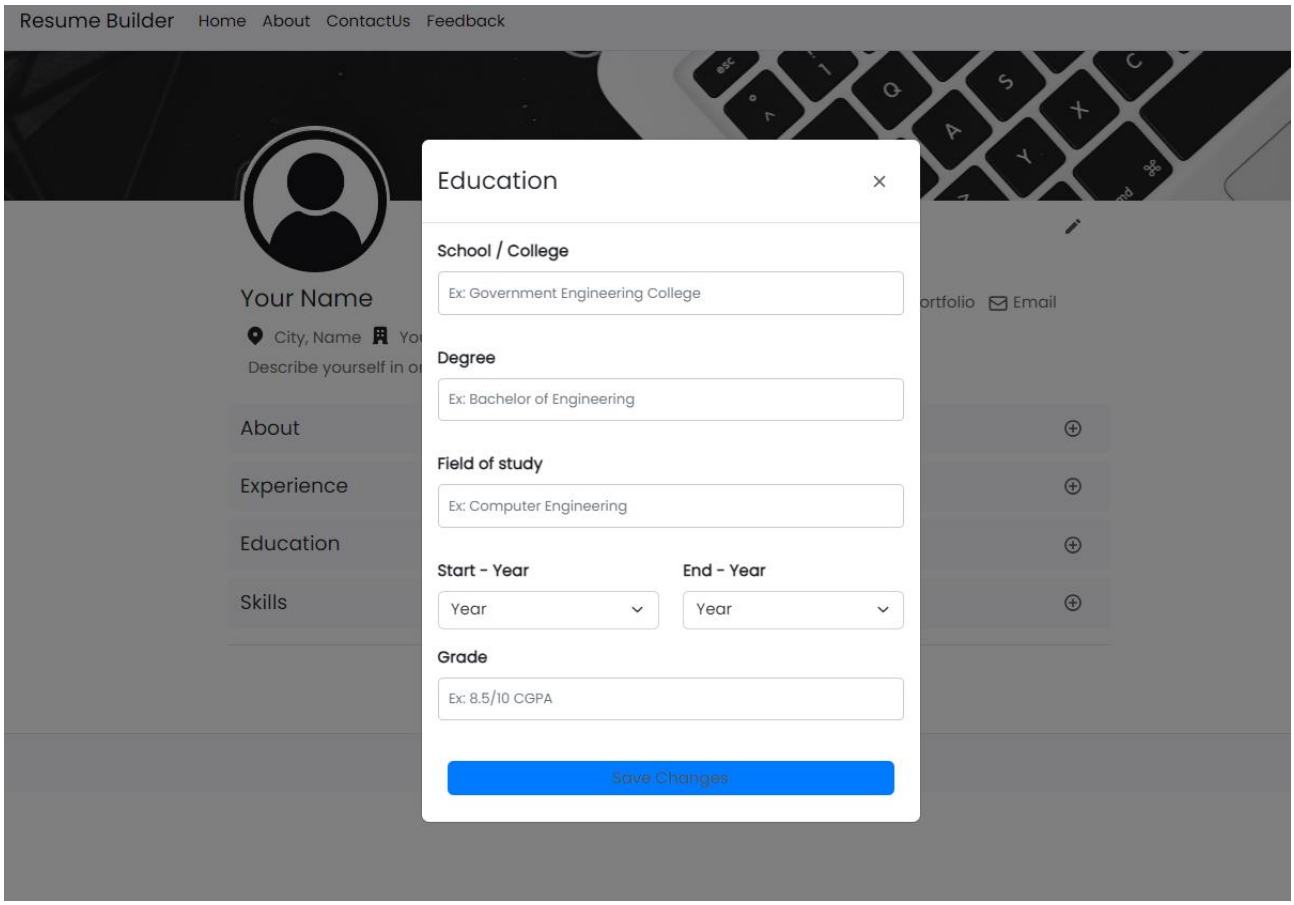
**Short Description:**
A candidate's "About Me" or "Summary" section in a professional context is a brief self-description intended to provide an overview of their background, skills, and career aspirations.



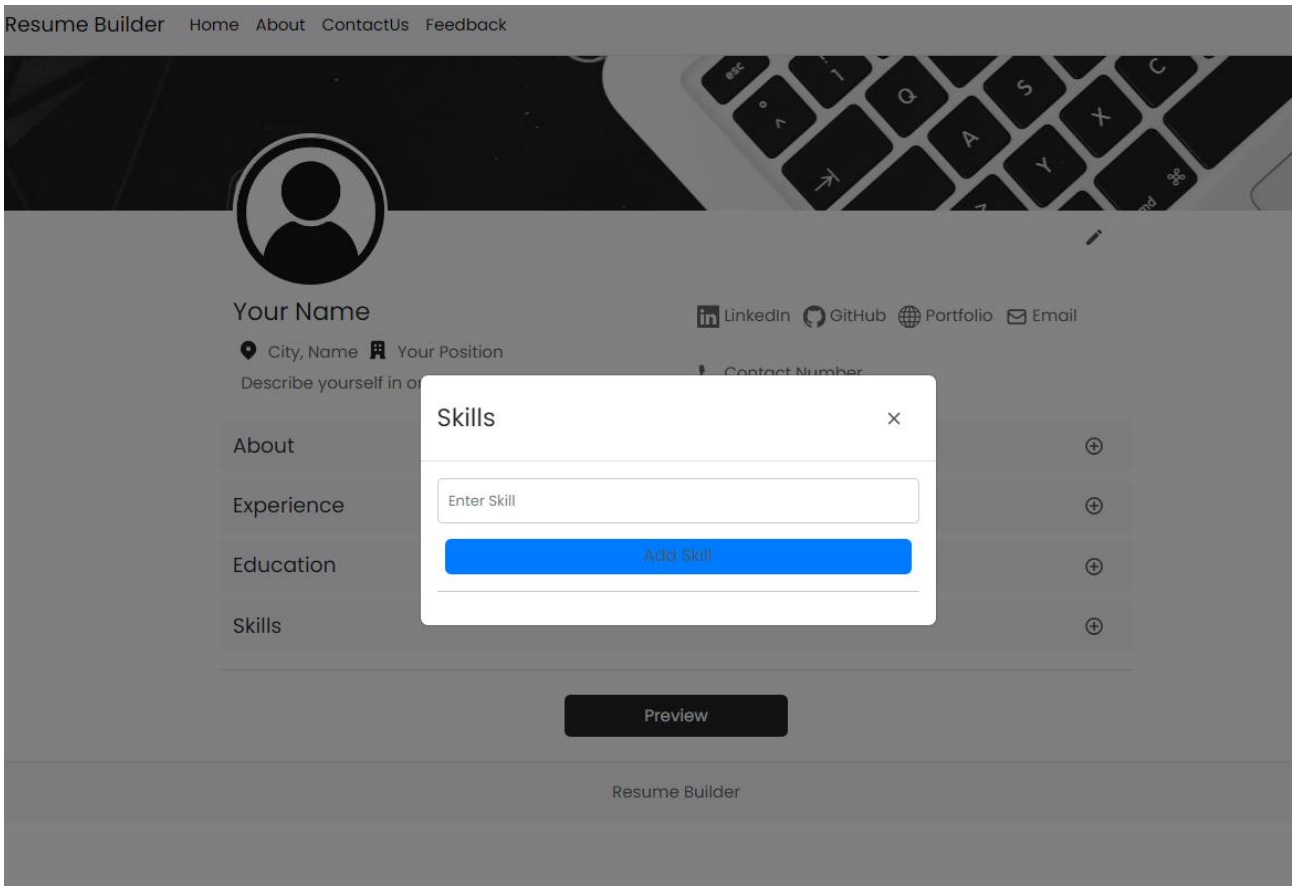
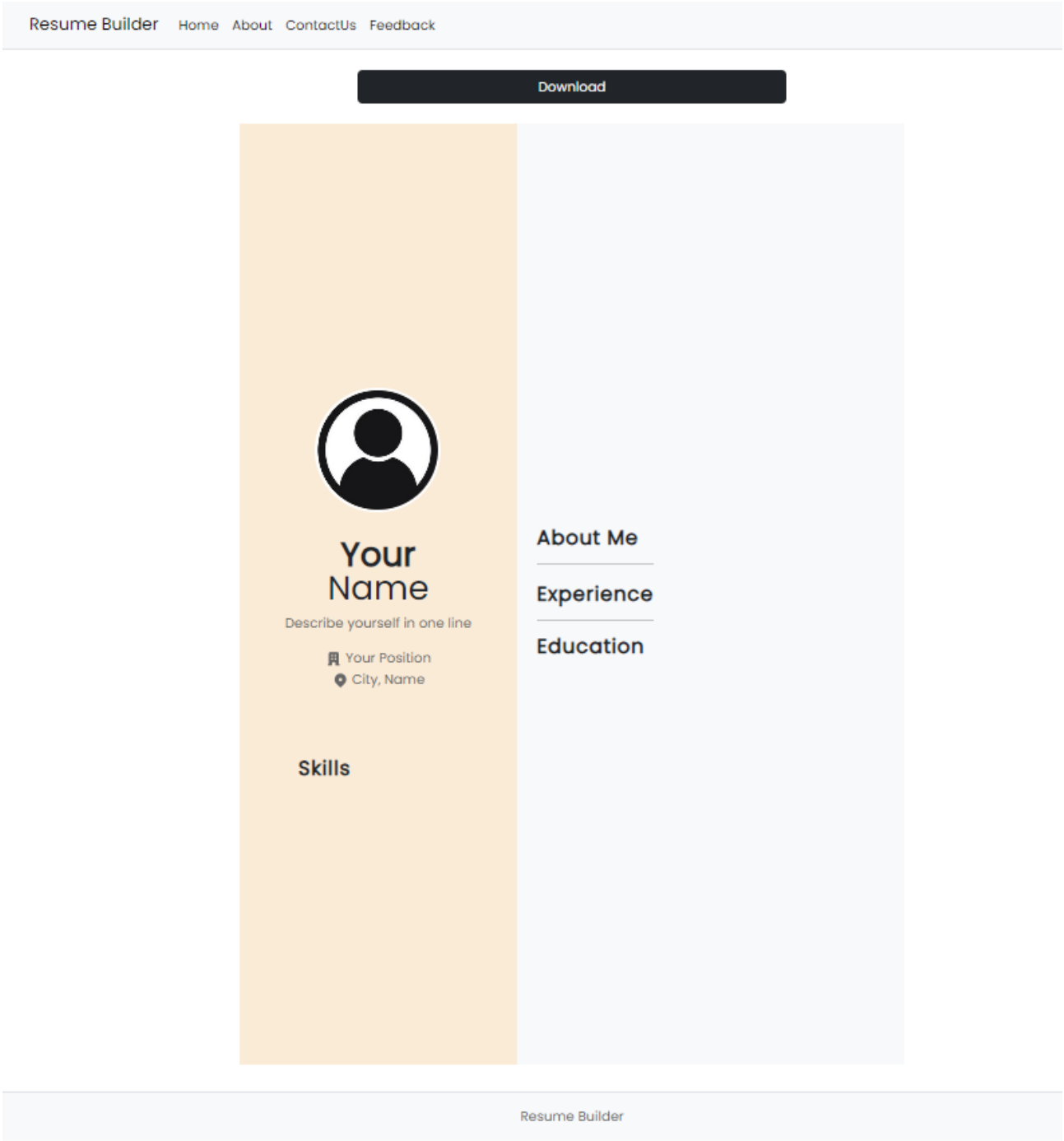
**Short Description:**
A candidate's experience section on a resume or professional profile outlines their work history and accomplishments.



**Short Description:**
A candidate's education section on a resume or professional profile outlines their academic background and qualifications.



**Short Description:**
A candidate's skills section on a resume or professional profile highlights the specific competencies and capabilities they possess.



**Short Description:**
A preview page on a resume builder website is a crucial step in the resume creation process. It offers users a glimpse of how their final resume will appear before they decide to download or share it.