All Good CV Generator

Team _SaaS_

Functional Requirements:

User Authentication: The system should have a user authentication mechanism to ensure only authorized users can access the system.

Data Collection: The system should be able to collect data from the user, such as personal information, work experience, education, and skills.

Template Selection: The system should offer various templates for the user to choose from, based on their job industry and personal preferences.

Automatic Formatting: The system should be able to automatically format the data collected from the user into the selected template.

Preview and Editing: The user should be able to edit the generated CV to make any necessary changes.

Proofreading tools: Integrated tool for proofreading i.e. reviews texts and seeks any spelling corrections, grammar mistakes, punctuation marks etc.

Export Formats: The user should be able to download the generated CV in various formats, such as PDF or Word.

Non-Functional Requirements:

Performance: The system should be able to generate a CV within a reasonable amount of time, depending on the amount of data provided.

Usability: The system should be easy to use and navigate, even for users with little to no technical experience.

Reliability: The system should be reliable and able to handle a large number of users simultaneously without crashing.

Security: The system should be secure and protect the user's personal data from unauthorized access.

Compatibility: The system should be compatible with various web browsers and devices.

Accessibility: The system should be accessible to users with disabilities, such as those who use screen readers or other assistive technologies

User-friendly Interface: The application should have a user-friendly interface that is easy to navigate and use.

Scalability: The application should be scalable to handle a large number of users and generate CVs simultaneously.

Solution Description:

The proposed solution is to develop an automated CV generation web application that generates professional and visually appealing resumes within a few minutes. The target platform for the application will be a web browser. The application will be developed using the T3 Stack. The following are the features and methods that will be included in the application:

1. **User Registration**: Users will be able to login using their Google or Discord account. More authentication options will be added later on, namely: Github, LinkedIn.

   1.1. **Guest Users**: Guest users can choose templates and generate CV by providing their information. But they will not be able to save their works and export files.

2. **Personal and Professional Details**: The application will provide users with input fields to enter their personal and professional details, such as education, work experience, skills, and achievements. The input fields will be customizable, allowing users to add or remove fields as needed.

   2.1. **Saving Information**: Registered Users can save their works in the database for future purpose.

3. **Customization Options**: The application will provide users with various customization options for the layout and design of the CV. This will include options for font, color, and layout, among others.

3.1. **Customization Tools**: The application will use Tailwind-CSS properties and TypeScript to provide these customization options. We will also be looking into RichText Component libraries such as Jodit React, React Quill etc.

4. **Preview and Editing**: The application will allow users to preview the generated CV and make any necessary edits before finalizing it. This will be done using TypeScript to dynamically update the preview as the user makes changes.

5. **Export Formats**: The application will primarily focus on exporting in PDF format. Additional formats will be added later.

   5.1. **Exporting Criteria**: The user has to be registered in the database to export their CV.

6. **User-friendly Interface**: The application will have a user-friendly interface that is easy to navigate and use. This will be achieved by using a modern and intuitive design, with clear labeling and instructions.

7. **Performance**: The application will generate the CV within a few minutes without any delay or lag. This will be achieved by using optimized algorithms and caching techniques to speed up the generation process.

8. **Compatibility**: The application will be compatible with different devices and web browsers. This will be achieved by using responsive design and ensuring compatibility with popular browsers such as Chrome, Firefox, and Safari.

9. **Security**: The application will ensure the security of user data by using encryption and other security measures. This will be achieved by using HTTPS for secure data transmission and storing user information in a secure Supabase (PostgreSQL) database.

10. **Scalability**: The application will be scalable to handle a large number of users and generate CVs simultaneously. This will be achieved by using load balancing techniques and cloud hosting services to handle increased traffic.

Target platform:

The proposed solution is a web-based application that can be accessed through any web browser on desktop, laptop, tablet, or mobile devices.

Methods:

1. **Front-end Development**: NextJS, ReactJS and TypeScript will be used for the front-end development of the application. Tailwind-CSS will be used for styling.

2. **Back-end Development**: TypeSCript will be used for server-side scripting, and Supabase (PostgreSQL) will be used as the database management system to store user data securely. Prisma ORM will be used to interface with the database.

3. **API Integration**: The application will be integrated with third-party APIs, such as Google, Github and Discord for authentication and authorization purposes using OAuth2 through NextAuth. tRPC will be used for fetching data. We will also use the Grammarly API to proofread in the application.

4. **Cloud Hosting**: The application may be hosted on Vercel.

5. **Version Control**: Git and Github will be used for version control to maintain code history and collaboration.

6. **Testing and Debugging**: The application will be thoroughly tested and debugged using tools such as Postman, Mozilla/ Chrome Developer Tools to ensure functionality and usability.

7. **Deployment**: The application will be deployed using Vercel's continuous integration/continuous deployment (CI/CD) pipeline.

8. **Documentation**: The whole workflow will be documented in Jira for smooth work progress. Also, project issues will be updated in GitHub issues.