

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



WEB PROGRAMMING (CO3049)

Assignment: CV creating website

Advisor(s): Nguyễn Đức Thái

Students: Mạc Hồ Do Khang - 2252297
Trương Minh Khang - 2252314
Lâm Gia Khánh - 2152651
Julien Holzer- 2360059
Đặng Việt Trung Nhân - 1852625

HO CHI MINH CITY, DECEMBER 2024



Contents

1 Members' Contribution	2
2 System requirement analysis	3
2.1 Purpose	3
2.2 Functional Requirements	3
2.3 Non-Functional Requirements	3
3 Design	3
3.1 Frontend	3
3.1.1 Main goals	3
3.1.2 Name and logo	4
3.1.3 User experience	4
3.1.4 Header	5
3.1.5 Footer	6
3.1.6 Home page	6
3.1.7 The CV	10
3.1.8 View CV page	10
3.1.9 Add CV page	11
3.1.10 Log in page	12
3.1.11 Register page	13
3.2 Backend	14
3.2.1 Database	14
3.2.2 Authentication	17
3.2.3 My CV	17
3.2.4 Add CV	17
3.2.5 View CV	17
3.2.6 Exception page	18
4 Development	18
4.1 Sprint 1: Planning and Setup	18
4.2 Sprint 2: Frontend Design	19
4.3 Sprint 3: Backend Development and Integration	19
4.4 Sprint 4: Finalization and Review	19
5 Testing	20
5.1 Strategy: User interface	20
5.2 Strategy: Authentication and authorization	20
5.3 Strategy: CVs management	20
5.4 Strategy: Database	20
6 Maintenance	20
6.1 Security Updates	21
6.2 Performance Optimization	21
6.3 Feature Enhancements	21
6.4 Database Management	21
6.5 Bug Fixes and Issue Resolution	21
6.6 Documentation Updates	22
6.7 Monitoring and Analytics	22



1 Members' Contribution

No.	Fullname	Student ID	Roles	Contribution
1	Mạc Hồ Do Khang	2252297	- Frontend	100%
2	Trương Minh Khang	2252314	- Backend	100%
3	Lâm Gia Khánh	2152651	- Backend	100%
4	Julien Holzer	2360059	- Frontend	100%
5	Dặng Việt Trung Nhân	1852625		0%



2 System requirement analysis

2.1 Purpose

The purpose of this web-based application is to enable jobseekers to create, manage, and share their CVs online while allowing viewers to access CVs via unique URLs. The system should be user-friendly, secure, and accessible.

2.2 Functional Requirements

- As for Jobseekers:
 - A jobseeker can fill out a form to create a CV.
 - A jobseeker can modify or delete their previously posted CV.
 - Jobseekers must log in or register to post a CV.
 - Jobseekers can set permissions to allow or restrict viewers from accessing their CVs.
- As for Viewers:
 - Viewers can access CVs via a unique URL
 - Viewers can only see CVs that are publicly shared or for which they have access permission.
- Each CV is assigned a unique URL containing the CV ID for easy sharing.
- Store jobseeker input data (e.g., address, phone, degrees, certificates) in a structured database.

2.3 Non-Functional Requirements

- The application must have an intuitive user interface with clear navigation for jobseekers and viewers.
- The CV retrieval process should be fast, with minimal delay in displaying CVs based on the CV ID.
- Authentication mechanisms should ensure only authorized users can post or edit CVs.
- The web application should be compatible with modern browsers and mobile devices.

3 Design

3.1 Frontend

3.1.1 Main goals

The front-end design ensures that the web application is user-friendly, visually appealing, and provides seamless interaction for jobseekers and viewers. Our website's design focuses on creating an intuitive layout for posting, managing, and viewing CVs.

Our main design features are:

- Use a desktop-first responsive approach to ensure compatibility across devices.

- Include a persistent navigation bar at the top of the page with links to key sections.
- Use JavaScript for real-time form validation (e.g., check required fields or email format).
- Provide visual cues including success messages and error alerts after form submission.

As for aesthetic design:

- We use a professional and minimalist color scheme, referenced from [Coolors](#), with the following:
 - Primary color: #1B263B
 - Tints: #415A77 and #778DA9
 - Shades: #0D1B2A
 - Background color: #E0E1DD
- We use the sans-serif font **Rubik**.
- We incorporate icons from [Ionicons](#).

3.1.2 Name and logo

Our website name is **CVForge**, and we design our logo accordingly using [Canva](#).

The **CVForge** logo's text combines two distinct colors: orange for the "CV" and blue for "ORGE," which creates a striking visual contrast. A red hammer icon replaces the "F," symbolizing the concept of forging or crafting. We consider our design clean, minimalist, and visually appealing, effectively balancing creativity with readability to leave a strong impression.

We then use that very same hammer icon as the website's favicon.



Figure 1: Side-by-side display of Favicon and Logo.

3.1.3 User experience

We also take into consideration user experience (UX), therefore we covered these following:

- When a link or a button is hovered, its color would change to indicate the user. In case of icons, they will slightly move upward. Every change takes place during a time interval of 300ms to provide a smooth experience.
- During adding a new CV:
 - We provide guidance for form fields using tooltips (e.g., "Enter your highest degree").



(a) Links: the first link is hovered, others are not.

(b) Buttons: the first button is hovered, the second is not.

Figure 2: Illustrations of hover effects on links and buttons.

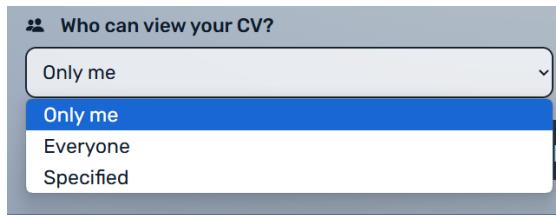


Figure 3: Dropdown for visibility setting

- During adding a new CV, jobseekers are allowed to preview their CV before submission.
- Offer an easy-to-use toggle or dropdown for setting CV visibility.

3.1.4 Header

We designed one header which is then used for every single web page. Our header is basically the navigation bar, together with the Logo on the left.

When a user is not logged in, he/she visits the website as a guest and the header will contain the links to the home page, the login page and the register page.



Figure 4: Header for guest users.

When a user is logged in, the header will change accordingly, containing the links to the Home page, the View CV page, the Add CV page and the user icon in which when hovered will reveal the current user name, user email and the Log out button.

In both cases, the logo acts exactly the same as the Home page link.

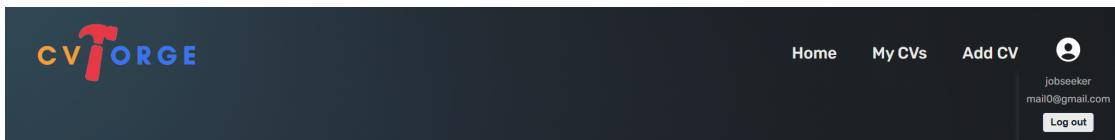


Figure 5: Header for logged in users with hovered user icon.

3.1.5 Footer

Similarly to header, we use the same footer for every single web page. It is organized into five distinct sections with clear headings and subitems, ensuring ease of navigation:

- Logo and branding. Below the logo, a row of social media icons (Facebook, TikTok, Skype, Discord, and Twitter) is displayed, allowing users to connect with the platform on various networks.
- Contact section
- Company section
- Account section
- Resources section



Figure 6: Our footer section

Note that all the links in the footer simply navigate to the beginning of the home page.

3.1.6 Home page

The home page serves as the gateway to the web-based CV management application, showcasing its features and benefits while providing easy navigation for users.

Hero section

We design our hero section to span exactly one viewport height of the screen.

There is a prominent Call-to-Action (CTA) button, "Start Now", leading to the jobseeker registration (if not logged in) or dashboard page.

There is also the featured-in section at the bottom in order to build trust and credibility.

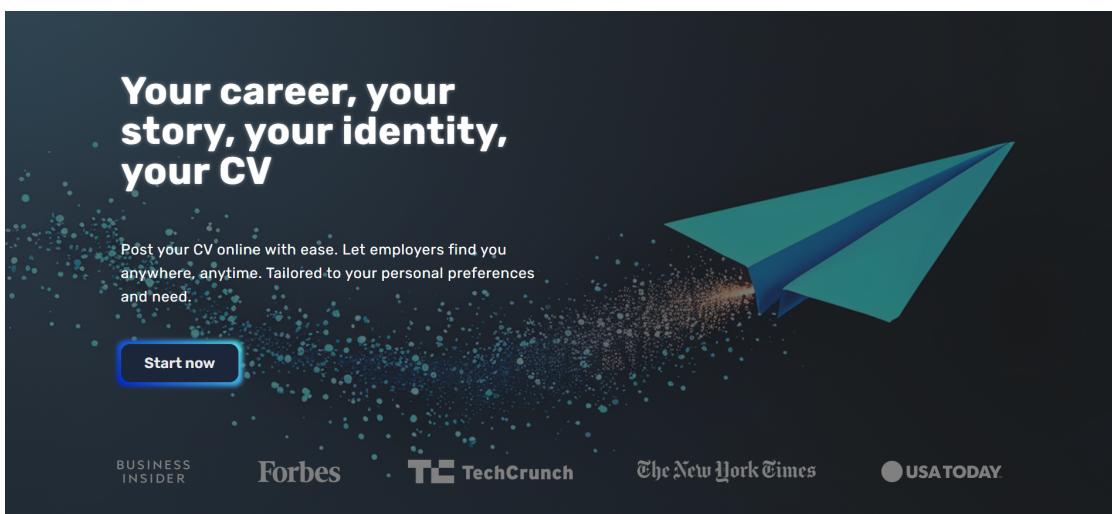


Figure 7: Hero section

Sample CVs section

This section provides visitors with a preview of what their CVs could look like using the application. We decided that our user cannot choose the layout of the CV but can select its color. Thus we only showcase 2 sample CVs of same layout with different colors.

In the middle, we list some of the strengths of our web-based application:

- Simple CV creation
- Customizable colors
- Secure data storage
- Shareable CV links
- PDF upload option
- Quick URL generation
- Fast CV updates



SAMPLES

CVForge has created 5,000+ CVs



Isabel Schumacher
Graphics Designer

About Me
I am a creative and honest connector, adapting well. I often work with others to find solutions. I have a strong desire to learn and grow.

Contact
+123-456-7890
mailto:hello@mygreatsite.com
123 Anywhere Street, Any City

Skills
• Web Design
• Branding
• Graphic Design
• SEO
• Marketing

Language
• English
• French

Education
(2011 - 2015)
Wardrobe University
Bachelor of Design
3.65

(2014 - 2019)
Wardrobe University
Bachelor of Design
3.74

Experience
(2020 - 2023)
Senior Graphic Designer
Fauget studio
• create more than 100 graphic designs for big companies
• complete a lot of complicated work

(2017 - 2019)
Senior Graphic Designer
Innovo Inc.
• create more than 100 graphic designs for big companies
• complete a lot of complicated work

CVForge works with many kinds of CVs

- ✓ Simple CV creation
- ✓ Customizable colors
- ✓ Secure data storage
- ✓ Shareable CV links
- ✓ PDF upload option
- ✓ Quick URL generation
- ✓ Fast CV updates



David McAllister
Data Engineer

About Me
I am a hardworking and reliable individual who enjoys working in a fast-paced environment. I am always looking for opportunities to learn and grow.

Contact
+123-456-7890
mailto:hello@mygreatsite.com
123 Anywhere Street, Any City

Skills
• Web Design
• Branding
• Graphic Design
• SEO
• Marketing

Language
• English
• French

Education
(2011 - 2015)
Wardrobe University
Bachelor of Design
3.65

(2014 - 2019)
Wardrobe University
Bachelor of Design
3.74

Experience
(2020 - 2023)
Senior Graphic Designer
Fauget studio
• create more than 100 graphic designs for big companies
• complete a lot of complicated work

(2017 - 2019)
Senior Graphic Designer
Innovo Inc.
• create more than 100 graphic designs for big companies
• complete a lot of complicated work

Figure 8: Sample CVs section

Testimonial section

We place the testimonial section to highlight positive feedback from jobseekers who have used the application to increase user trust and motivation.

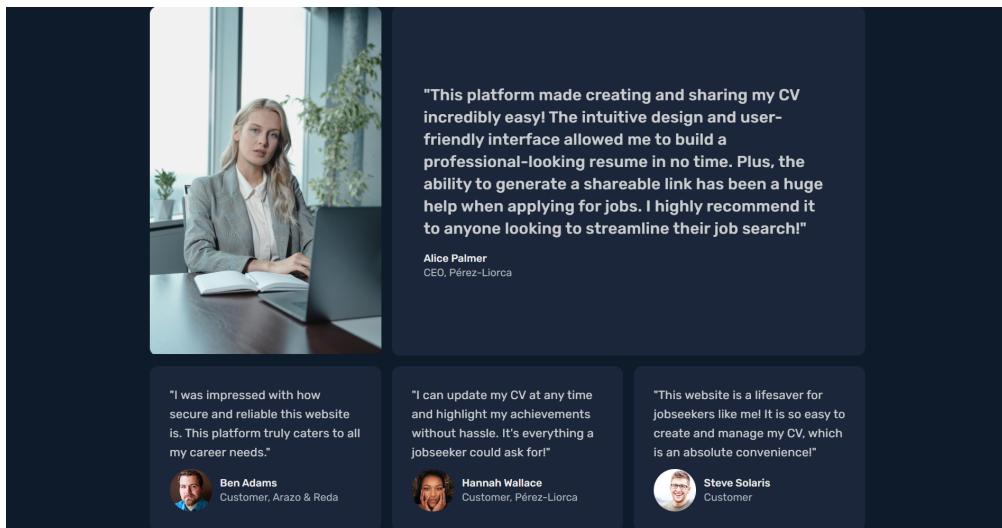


Figure 9: Testimonial section

FAQs

This section provides answers to common questions about the application to reduce confusion and enhance the user experience. This follows a collapsible accordion-style layout for questions and answers. The covered questions are:

- How do I create a CV on this website?
- Can I update my CV after posting it?
- Is my CV visible to everyone?

FAQS

Frequently asked questions

01 How do I create a CV on this website? >

02 Can I update my CV after posting it? ▾

Yes, you can edit your CV anytime to keep it up-to-date. Simply:

- Log in to your account and go to the "My CVs" section.
- Select the CV you wish to update.
- Make the necessary changes and save them.
- The changes will automatically reflect on your shared CV URL.

03 Is my CV visible to everyone? >

Figure 10: FAQs section



3.1.7 The CV

We design our CV's general layout to have two columns. On the left one, there is a portrait followed by personal information including: objective, personal contacts, skills, reference, and additional information. Everything here is on a dark background color with a white color for the letters.

In the right column, details about one's progress including his/her educational and experience history are displayed. Everything here is on the white background color with that dark color for the letters.

Isabel Schumacher
Graphics Designer

About Me
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam pharetra in lorem at laoreet. Donec hendrerit libero eget est tempor, quis tempus arcu elementum.

Contact
+123-456-7890
hello@reallygreatsite.com
123 Anywhere Street, Any City

Skills
• Web Design
• Branding
• Graphic Design
• SEO
• Marketing

Language
• English
• French

Education
(2011 - 2015)
WARDIERE UNIVERSITY
Bachelor of Design
3.65

(2014 - 2019)
WARDIERE UNIVERSITY
Bachelor of Design
3.74

Experience
(2020 - 2023)
SENIOR GRAPHIC DESIGNER
Fauget studio
• create more than 100 graphic designs for big companies
• complete a lot of complicated work

(2017 - 2019)
SENIOR GRAPHIC DESIGNER
larana, inc
• create more than 100 graphic designs for big companies
• complete a lot of complicated work

Figure 11: A sample CV

3.1.8 View CV page

The selected CV are displayed in the middle. All related details such as its URL, creating date, visibility and password (if any) are displayed on the right. On the left, there are vertical tabs that navigate to different CVs, as well as a new one.

From this page, one can edit or remove his/her CV(s).



The screenshot shows the CVForge application interface. On the left, there is a sidebar titled "ADD CV" with four items: Untitled, Untitled2, Senior Form, and Untitled. The main area displays a CV preview for "Johnny Someone". The preview includes a circular profile picture of a woman, the name "Johnny Someone" in bold, and the title "Senior Enginner". To the right of the preview are several status indicators: a link to "http://localhost:1", the date "2024-11-21 15:52:03", a lock icon indicating "No password is set.", and a "Private" status. Below these are two buttons: "Edit" and "Delete".

Figure 12: View CV page

3.1.9 Add CV page

This section simply contains a complete form for jobseekers to input their data, after which a preview of the CV is displayed and user can view different colors. Once he/she is satisfied with the final result, the CV can then be created.

The screenshot shows the "Add CV" form for "PRIMARY INFORMATION". The form includes fields for "Objective (*)" (with placeholder text "Lorem ipsum dolor sit amet..."), "Full Name (*)" (input: "John M. Doe"), "Email (*)" (input: "john@example.com"), "Address (*)" (input: "123 Any Street"), "Current Job (*)" (input: "Graphic Designer"), and "Phone Number (*)" (input: "919-263-1770"). There are dropdown menus for Country, State, and City.

Figure 13: The form in which user input their data

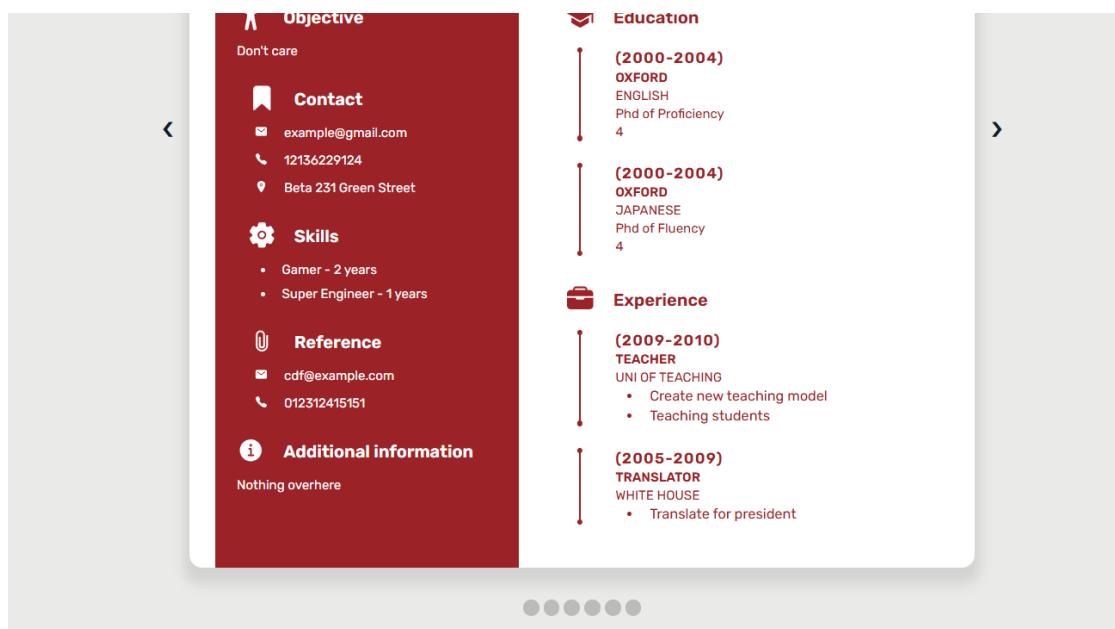


Figure 14: Selection between different colors

3.1.10 Log in page

The login page of the website is designed with a clean and minimalistic layout. On the left side, there is a background image depicting a hand pointing at a resume, divided into a grid-like structure for a visually appealing effect.

On the right side, a login form is presented, consisting of labeled input fields for "Username" and "Password.". Below the form, a prominent blue gradient button labeled "Login" facilitates user interaction.

Navigation options provide links to "Home" and "Sign up." The "Sign up" button is highlighted with a dark blue background, contrasting with the page's overall white and light gray color scheme, creating a clear call to action.

If something goes wrong, a red message gets echoed to notify the users the problem

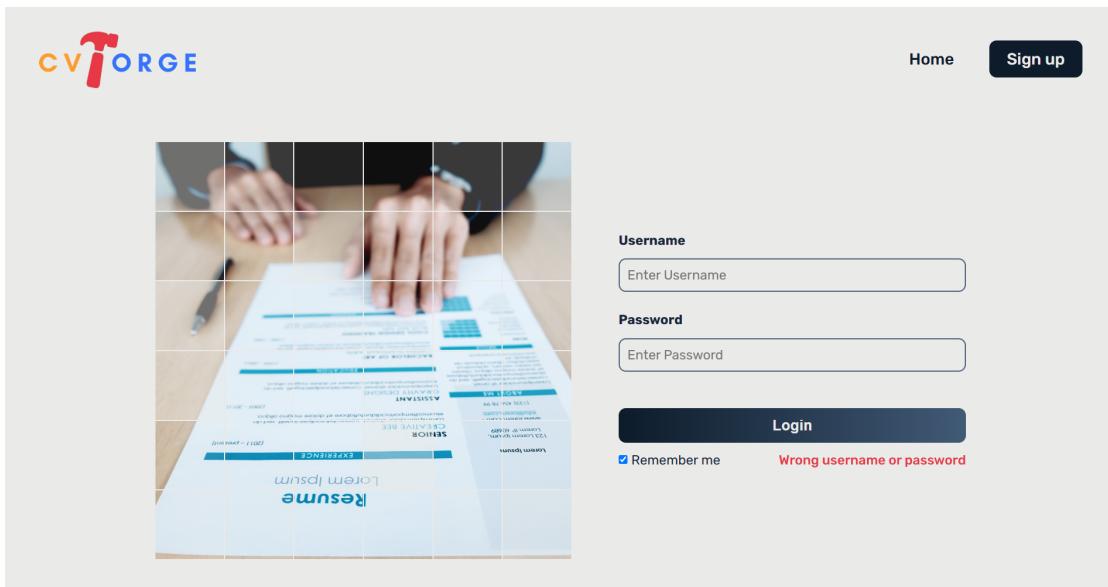


Figure 15: Log in page

3.1.11 Register page

We design our sign-up page in a way that can encourage user engagement with a welcoming and professional design. Navigation links for "Home" and "Login" are placed in the top right corner, with the "Login" button highlighted in dark blue.

The main content is centered around a call-to-action message on a gradient blue background. The sign-up form includes input fields for "Email," "Password," and "Full name," with placeholder text providing examples of what to enter.

To the right of the form, we place a contextual image which shows a person reviewing a clipboard with a professional resume, reinforcing the theme of career development.

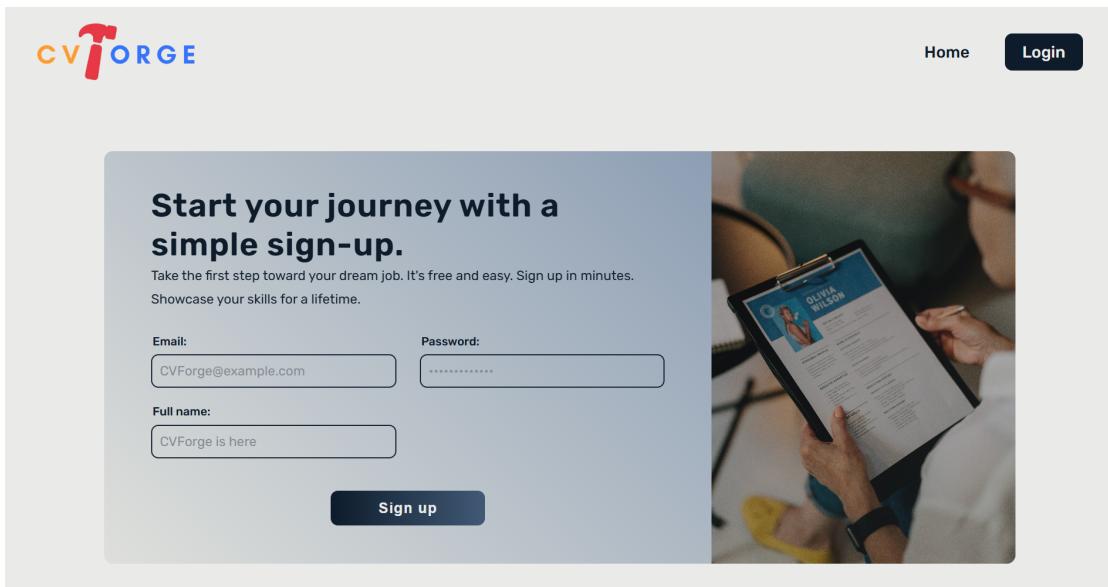


Figure 16: Register page

3.2 Backend

3.2.1 Database

In this project, there are two important entities to store data for: User and CV.

- User: contains data for authentication.
- CV: contains all necessary data to construct a proper CV.
 - "Own" relationship: Show which CV a user own.
 - "Allow" relationship: Store data for which user can access which CV.
 - Access level: Indicate the authorization mode of the CV.
 - * 1: Private. No one other than the owner is allowed to view.
 - * 2: Public. Any login user can view it.
 - * 3: Specified. Only those who specified by the owner can view. Permission is stored in the form of "Allow" relationship.
 - : Color: Store data about the template of the CV.
 - : Other field: Data of the CV.

In implementation, it will be implemented according to Figure 18.

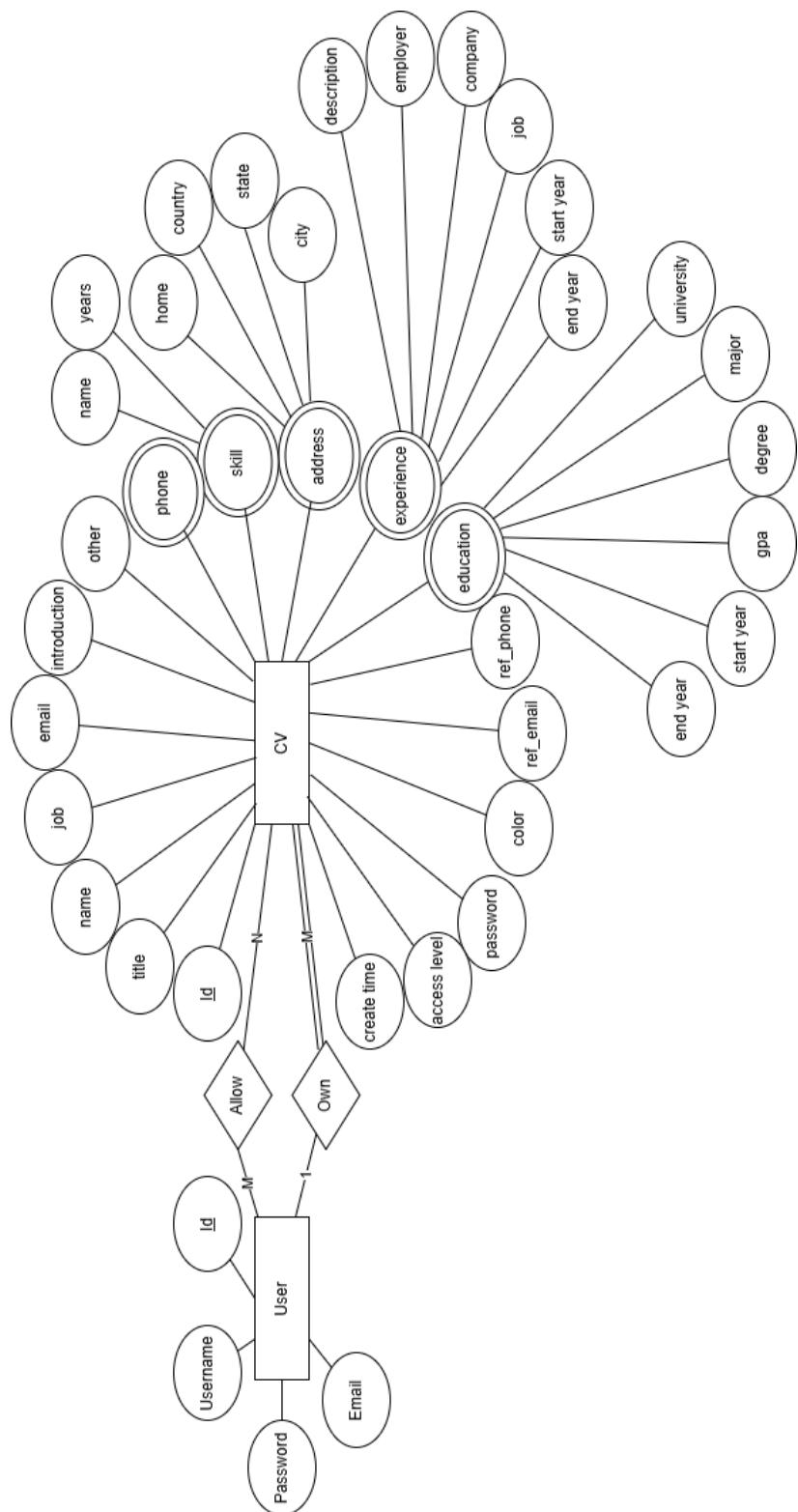


Figure 17: ERD diagram

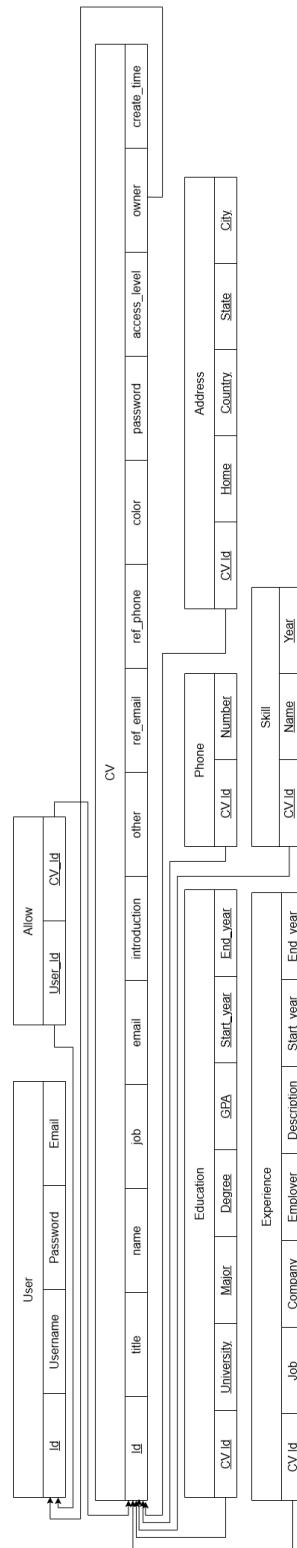


Figure 18: ERD mapping



3.2.2 Authentication

Register: After user inputs all fields, check for any missing field and verified if the email is valid. Passing all of the validation, the data will be inserted as a new row in table "User" in database.

Login: We only require user to input username and password. After it's done, both field will be compared to the data in table "User" in the database. If there's no corresponding row, it will prompt an error and make user input again. Otherwise, the user is granted access to the main page and all of the functions the web can provide. Additionally, session will save user's id, username and email for later convenience.

3.2.3 My CV

This is the page where jobseeker can view all of their CV without using password. First, the page sends a request to the server to fetch all CVs that the user owns. Depending on the data, the web will construct the UI to display all data of the CV. In addition to data that appears on the CV, it will also display information such as the URL to view that CV, the date it was created, password if one does set it up and all users who are permitted to view this CV. Additionally, user can also edit and delete after selecting a CV. Edit will send the user to "Add" page with an additional argument for the current CV's id. For deletion, clicking on "Delete" will erase all related data to the current CV and the UI will update to reflect this. If the user's currently hold no CV, the page will request the user to create on in "Add CV".

3.2.4 Add CV

This is the page where jobseeker creates new CV. It has two stages: inputting data in the form and selecting template. All field that are marked with "(*)" are required to input, other are optional to do. In addition to that, all phone number and email field are checked to see whether if they are valid. After user filling the form, they can presses on "Prepare CV" to validate form's data and move on the next stage. Fail the validation and user has to fix all the error displayed on screen. When selecting template start, it will displayed a preview CV. All data from the input form will be transferred on to this CV and display to user. For selecting template, one has already been chosen for user but they can choose another one from the given choices. Done with everything, "Add CV" can be clicked to save the CV on the database. Data to save included everything in the form and the template that's the user choose.

If this page has additional arguments "cvid", this means it's in edit mode. All data of the CV identified with "cvid" variable will be fetched from the database and filled in the form and the reviewed CV when initializing. User can make any change as long as it still satisfies the validation. If user fail to do so, it will force the user to fix all error and will not change the data on the database when clicking on "Edit CV". If not, the changes will update on the database and user is sent back to the "myCV" page.

3.2.5 View CV

This page cannot be accessed through normal UI. Instead, user has to type a corresponding URL on browser's address bar to see it. This URL is displayed on "MyCV" page for each CV. However, attempt to do this while not logging in, the system will instead redirect the user to login. The URL has the form "<hostname>/<cvid>" which will be translated to "<hostname>/page=viewCVs&cvid=<cvid>". This is achieved by modifying the ".htaccess" file as in the figure 20 .



```
RewriteEngine On  
RewriteRule ^([0-9])+$ /index.php?page=viewCVs&cvid=$1 [L,R=302]
```

Figure 19: .htaccess's URL redirect modification

Redirecting to viewCV page, it will first look for the access level of the user for this CV. Accessing with an invalid <cvid>, the page will raise no CV exception. If the user's is the owner, grant access regardless immediately to the user. If not, it will further check the access level of the CV. A private CV will not show any information and the page will display "Access Deny" message. It will also be the case for unauthorized people to access to CV set with a specified permission. Only public CV and authorized people to the CV will be able to access the page. If the CV is rigged with the password, user is required to input it in a form. When they input it right, the server will fetch CV's data and fills it in the appeared preview CV. If there is no password, user can see the CV filled with data from the beginning.

3.2.6 Exception page

In case the server has some error, the user will be redirected to this page. To do so, ".htaccess" file is added with the following line:

```
ErrorDocument 404 /index.php?page=error&errMessage=pageNotFound
```

Figure 20: .htaccess's error handling modification

4 Development

Our project followed a Scrum methodology to ensure effective collaboration, iterative progress, and the ability to adapt to changes as needed. The process was divided into several sprints, each with a clear focus and deliverables, allowing us to stay on track and meet the project objectives efficiently.

4.1 Sprint 1: Planning and Setup

We began the project with a kickoff meeting to establish its foundation. During this initial meeting, we discussed the overall goals, assigned roles within the team, and decided on the tools and technologies to use. To streamline our work, we adopted a modular approach, splitting the team into two groups: one focusing on the frontend and the other on the backend. This allowed parallel progress on different components of the project.

- Team Meetings: We scheduled meetings twice a week to review progress, share updates, and address challenges. These meetings ensured clear communication between the two groups and helped us align our efforts effectively.
- Task Breakdown: We identified key tasks for the frontend and backend teams, such as designing user interfaces, implementing the database structure, and integrating authentication features. Although we didn't create a formal backlog, these tasks were informally prioritized to guide our work during the development phase.



By focusing on collaboration and regular check-ins, we were able to maintain steady progress and address any issues promptly, laying a solid foundation for the subsequent sprints.

4.2 Sprint 2: Frontend Design

During this sprint, the frontend team focused on designing and building the user interface for the project. We took inspiration from the personal websites we had developed in a previous lab, which provided us with a solid foundation for the design. This prior experience allowed us to reuse and adapt ideas, speeding up the design process and ensuring a consistent structure.

- Leveraging Previous Experience: We drew from the personal websites we built in a previous lab, adapting the structure and styling to fit the current project's requirements. This not only saved time but also ensured we had a proven base for the interface design.
- Prioritizing User-Friendly Design: The frontend team worked to create a clean, intuitive, and responsive interface. We aimed to make sure the design was easy to navigate and accessible across different devices, especially focusing on mobile compatibility to ensure a positive user experience.

By building upon familiar design concepts, we were able to quickly develop an interface that met the project's needs, allowing us to focus on enhancing its functionality and overall usability.

4.3 Sprint 3: Backend Development and Integration

In this sprint, the backend team focused on implementing the core functionalities and connecting them to the frontend. Their tasks included building user authentication, creating unique URLs for CVs, and managing database interactions.

- Independent Development: The backend team worked on their assigned tasks, ensuring that all features were functional and met the project's requirements.
- Testing and Debugging: Throughout the sprint, testing and debugging were carried out to ensure the backend functions integrated seamlessly with the frontend. Issues were identified and resolved as they arose, allowing for smooth collaboration between the two teams during the integration process. This sprint established the essential features and ensured the application was ready for final adjustments and deployment.

This sprint established the essential features and ensured the application was ready for final adjustments and deployment.

4.4 Sprint 4: Finalization and Review

The final sprint was dedicated to reviewing, testing, and documenting the project.

- Comprehensive Testing: We simulated various user scenarios to ensure the application worked as intended. This included checking for security vulnerabilities, such as unauthorized access to CVs.
- Report Writing: As a team, we collaborated to draft the project report, ensuring all aspects of the development process were accurately documented.

By following Scrum principles, we were able to maintain clear communication, adapt to challenges effectively, and deliver a project that met the initial requirements and expectations.



5 Testing

We test the system using test cases based on some strategies

5.1 Strategy: User interface

We test if the website is user-friendly on multiple environments. Appearance must be clear, straight forward. If hyperlinks exists, they must point to the correct destination.

Test case	Result	Number of cases
Zooming browser	OK	3
Multiple browsers	OK	2
Button and clickables	OK	10

5.2 Strategy: Authentication and authorization

We test on multiple registers, logging in and logouts, some must fail and some must succeed. There are also tests in accessibility, which mean allowing users on different rights.

Test case	Result	Number of cases
Valid logging in	OK	3
Invalid logging in	OK	2
Logging out	OK	1
Register	OK	3
Access rights on CVs	OK	5

5.3 Strategy: CVs management

We also test the ability to manage CVs, including certain operation on CVs. The database is also involved in this process.

Test case	Result	Number of cases
Create CV	OK	30
Remove CV	OK	5
Edit CV	OK	25
List CV	OK	5

5.4 Strategy: Database

Test case	Result	Number of cases
Multi-value data	OK	2

6 Maintenance

Although the web application is a project and will not be maintained beyond its completion, we outlined a potential maintenance plan to ensure its long-term usability and relevance if required. This section discusses maintenance strategies and best practices that we recommend based on the project's design and implementation.



6.1 Security Updates

We recognize the importance of keeping the application secure. If maintenance were to continue:

- We would periodically review and update the authentication system to ensure it aligns with current security standards.
- Regular checks for vulnerabilities, such as SQL injection or cross-site scripting (XSS), would be part of the maintenance routine.
- HTTPS would be implemented to provide encrypted communication, ensuring user data remains protected during transmission.

6.2 Performance Optimization

To maintain a seamless user experience:

- We would monitor and optimize the performance of database queries to handle larger datasets efficiently as more CVs and users are added.
- Enhancing page loading times and ensuring compatibility across various devices and browsers would remain a priority.

6.3 Feature Enhancements

If users request additional features or modifications:

- We would adopt an iterative development approach to implement updates without disrupting existing functionalities.
- Collecting user feedback would guide the prioritization of new features, such as customizable CV templates or advanced search capabilities for viewers.

6.4 Database Management

For ongoing stability and reliability:

- We would perform regular backups of the database to prevent data loss in case of hardware or software failures.
- Archiving older CVs or inactive user data could help maintain optimal database performance.

6.5 Bug Fixes and Issue Resolution

We understand that any application requires continued bug fixes. If the project were actively maintained:

- We would establish a system for tracking and prioritizing reported bugs.
- Quick resolution of issues would ensure the application remains functional and reliable for its users.



6.6 Documentation Updates

To ensure the project remains maintainable for future developers:

- We would keep the project's technical documentation up to date, reflecting any changes or additions made during maintenance.
- This would include updating diagrams, code comments, and user manuals.

6.7 Monitoring and Analytics

To understand user behavior and identify potential improvements:

- We would integrate analytics tools to monitor application usage patterns.
- These insights would help us proactively address areas needing enhancement or optimization.

While this maintenance plan outlines what we could do if the application were maintained, it is not expected to be carried out since the project was intended as a learning experience. However, we believe these strategies provide a solid foundation for ensuring the application's sustainability if required.