Project Progress Report on

CV GENERATING WEB APP

Submitted in fulfillment of the requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATION

Submitted By:

Ankit Kumar Thakur **2102144**

Under the Guidance of

Mr. Aditya Joshi

Assistant Professor

Department of Computer Application

Department of Computer Application

Graphic Era (Deemed to be University)

Dehradun, Uttarakhand

Jun-2024

Graphic Era
Deemed to be
University
DEHRADUN

CANDIDATE’S DECLARATION

I hereby certify that the work which is being presented in the Project progress report entitled **“CV Generating web app”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Computer Applications in the Department of Computer Application of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **Mr. Aditya Joshi, Assistant Professor,** Department of Computer Application, Graphic Era (Deemed to be University), Dehradun.

Ankit Kumar Thakur	2102144	Sign.

The above mentioned student shall be working under the supervision of the undersigned on the **“CV Generating web app”**

Supervisor **Head of the Department**

<u>Examination</u>

Name of the Examiners: **Signature with Date**

1.

2.

Work Satisfactory: Yes/ No

Table of Contents

List of Figures:

ABSTRACT

In today's competitive job market, a well-crafted CV (Curriculum Vitae) is essential for job seekers to stand out and secure employment opportunities.[1] This web application provides a user-friendly platform for creating professional CVs with ease and efficiency. The web application features a guided step-by-step process, customization options to cater to various industries and personal preferences. Users can input their personal information, work experience, education, skills, and other relevant details, which are then automatically formatted into a polished, professional-looking CV. The platform also includes the ability to export the final document in multiple formats (PDF, Word, etc.). By leveraging modern web technologies, this application aims to simplify the CV creation process, helping users present their qualifications effectively and increase their chances of landing their desired jobs.

Keywords—CV Generator; Curriculum Vitae; WebApplication; Web Development

Chapter 1 Introduction and Problem Statement

1.1 Introduction

In the modern job market, a standout Curriculum Vitae (CV) is a crucial tool for job seekers striving to capture the attention of potential employers. A CV not only highlights a candidate's qualifications, skills, and experience but also serves as the first impression in the recruitment process.[2] However, crafting a compelling and professional CV can be a daunting task for many individuals, especially those who lack design skills or are unfamiliar with industry standards.

This report introduces a web application designed to streamline the CV creation process, making it accessible and straightforward for users of all backgrounds. The CV generating web app offers a comprehensive solution by providing a user-friendly interface, customizable templates, and automated formatting tools. Users can easily input their personal information, work history, education, and other pertinent details, which the application then organizes into a visually appealing and professional document.

The primary goal of this web application is to empower job seekers by simplifying the complexities of CV writing, thereby enhancing their chances of securing employment. By leveraging advanced web technologies and an intuitive design, this application addresses common challenges faced during CV preparation and provides users with a reliable tool to present their qualifications effectively.

This report will explore the features, functionality, and benefits of the CV generating web app, demonstrating its value as an indispensable resource for today's job seekers.

1.2 Problem Statement

The problem Statement for the present work can be stated as follows:

As stated above that, in the highly competitive job market, creating a well-structured and visually appealing Curriculum Vitae (CV) is essential for job seekers to effectively showcase their skills, qualifications, and experience. Despite the importance of a professional CV, many individuals struggle with the task due to a lack of design skills, unfamiliarity with industry standards, and the time-consuming nature of the process. This often results in CVs that fail to capture the attention of employers, diminishing the job seekers' chances of securing interviews and job offers.

Figure 1.1 Representation of common CV getting rejected

Traditional methods of CV creation, such as using word processors or relying on outdated templates, do not adequately address these challenges. They often require significant manual effort to format and customize, leading to inconsistencies and errors. Additionally, these methods do not provide real-time feedback or suggestions for improvement, further complicating the task for users who are unsure of best practices in CV writing.[3]

The problem is exacerbated for individuals who are transitioning into new industries, recent graduates, or those re-entering the workforce, as they may be unfamiliar with the specific requirements and expectations of employers in their target fields. Consequently, there is a clear need for a solution that simplifies the CV creation process, ensures adherence to professional standards, and enhances the overall quality and effectiveness of the final document.

This report addresses these issues by presenting a web-based CV generating application designed to provide an intuitive, efficient, and accessible platform for creating professional CVs. The application aims to eliminate the common obstacles faced by job seekers in CV preparation, offering a comprehensive tool that facilitates the creation of high-quality, industry-appropriate CVs with minimal effort and maximum impact. resource.

1.3 Proposed Solution

To address the challenges faced by job seekers in creating professional and effective CVs, we propose the development of a comprehensive web-based CV generating application. This solution leverages modern web technologies to provide an intuitive and efficient platform that simplifies the CV creation process while ensuring high-quality results. The key features of the proposed solution include[4]:

1. User-Friendly Interface: The application will feature a clean, intuitive interface that guides users through the CV creation process step-by-step. This ensures that even users with minimal technical skills can easily navigate and utilize the tool.

2. Customizable Templates: Users will have access to a variety of professionally designed templates tailored to different industries and job roles. These templates will be fully customizable, allowing users to modify layouts, fonts, colors, and sections to suit their individual preferences and needs.

3. Automated Formatting: The application will automatically format the entered information, ensuring consistency and professionalism. This feature eliminates the manual effort required to adjust margins, align text, and maintain uniformity throughout the document.

4. Real-Time Previews: Users will be able to see real-time previews of their CV as they input their information. This feature provides immediate feedback and allows users to make adjustments on the fly, ensuring the final document meets their expectations.

5. Content Suggestions and Tips: The application will include a library of sample phrases, tips, and best practices for CV writing. It will also offer contextual suggestions based on the information provided, helping users enhance the content and structure of their CVs.

6. Multi-Format Export: Users will have the option to export their CVs in multiple formats, including PDF, Word, and plain text. This flexibility ensures that users can easily submit their CVs in the preferred format of potential employers.

7. Secure Data Management: The application will prioritize user privacy and data security, employing robust encryption and secure storage practices to protect personal information.

8. Mobile Compatibility: The web app will be fully responsive and optimized for mobile devices, allowing users to create and edit their CVs on-the-go.

By incorporating these features, the proposed CV generating web application aims to address the common pain points associated with CV creation. It provides a reliable and accessible tool that enhances the quality and effectiveness of CVs, thereby increasing job seekers' chances of securing employment. This solution not only streamlines the CV writing process but also empowers users to present their qualifications in a professional and impactful manner.

Chapter 2 Objectives And Applications

2.1 Objective:

The proposed work objectives are as follows:

- Simplify the CV Creation Process: Provide an intuitive and user-friendly platform that simplifies the task of creating a professional CV, making it accessible to users with varying levels of technical proficiency and design skills.
- Enhance CV Quality: Offer high-quality, customizable templates to ensure that users can create visually appealing and professionally formatted CVs that adhere to industry standards.
- Increase Job Seeker Competitiveness: Equip users with the tools and guidance needed to craft compelling and well-structured CVs that effectively showcase their qualifications, skills, and experience, thereby increasing their chances of securing job interviews and employment offers.
- Ensure Flexibility and Accessibility: Enable users to export their CVs in multiple formats (PDF, Word, plain text) and ensure that the application is mobile-compatible, allowing for seamless use on various devices and operating systems.
- Provide Industry-Specific Guidance: Include a library of sample phrases, tips, and best practices tailored to different industries and job roles, helping users to create CVs that meet the specific expectations of their target employers.

By achieving these objectives, the CV generating web application will serve as a valuable tool for job seekers, helping them to present their credentials in the best possible light and navigate the competitive job market with greater confidence and success.

2.2 Applications:

So now the question arises here that, for whom this CV generator is suitable to use? Mainly the use of CV generator can be done by these:

1. Job Seekers

- Recent Graduates: New graduates entering the job market can use the application to create their first professional CVs, ensuring that their academic achievements, internships, and skills are presented effectively.
- Career Changers: Individuals transitioning to new industries or job roles can leverage industry-specific templates and content suggestions to tailor their CVs to their new career paths.
- Experienced Professionals: Professionals seeking new opportunities can quickly update and enhance their existing CVs, highlighting their experience, accomplishments, and skills.

2. Recruitment Agencies:

- Streamlining CV Preparation: Recruitment consultants can use the application to assist clients in creating polished CVs that meet employer expectations, thus improving the chances of successful job placements.
- Consistency and Quality Control: The application ensures consistency in formatting and presentation, enabling agencies to maintain high standards across all client CVs.

3. Educational Institutions:

- Career Services: Universities and colleges can integrate the application into their career services, providing students with a robust tool to prepare professional CVs for internships and job applications.
- Workshops and Training: Career counselors can use the application during workshops and training sessions to teach students effective CV writing techniques and best practices.

4. Corporate HR Departments:

- Internal Mobility: HR departments can use the application to help employees create CVs for internal job postings and promotions, facilitating career development within the organization.
- Talent Acquisition: HR professionals can assist candidates in creating professional CVs, ensuring that potential hires present their qualifications effectively.

5. Freelancers and Contractors and Multiple recruiters:

- Project Bidding: Freelancers and contractors can create tailored CVs for specific projects or clients, showcasing relevant skills and experience to increase their chances of winning contracts.
- Portfolio Integration: The application can help freelancers integrate their CVs with their portfolios, providing a comprehensive overview of their professional capabilities.
- Volunteer Recruitment: Non-profits can use the application to help volunteers create CVs that highlight their skills and experiences, matching them with suitable roles within the organization.

By catering to these diverse user groups, the CV generating web application not only streamlines the CV creation process but also enhances the overall quality and effectiveness of CVs across various contexts.[5] This comprehensive approach ensures that users can present their qualifications in the best possible light, thereby improving their chances of success in their respective endeavors.

2.3 Literature Survey/Related Work

1. Web based resume generator : In this paper abstract says, In reality, most recruiters look over a résumé for no more than 6 seconds on average. To provide valuable inputs into professionally formatted résumé can stand-out as per desired job requirements. Therefore, the purpose of this system is to generate résumé with content recommendations specifically for the IT job field. Researchers opted Waterfall model has been chosen as the methodology because the project is being developed step by step from requirement gathering and analysis to testing and evaluation. Content-based recommendation algorithm is implemented into the system as it can recommend the best skill content suitable for a job application. **[6]**

2. Web-Application on CV-Building - This academic report presents a research proposal to develop a web-based CV- building platform using ReactJS and the MERN (MongoDB, Express.js, ReactJS, Node.js) stack. Traditional CV creation methods often lack the interactivity and proactive features needed to make an impression on potential employers for a considerable amount of time. The platform's goal is to offer users a versatile and user-friendly interface to create personalized and visually impressive CVs, enhancing their capacity for work. **[7]**

3. Resume Builder Application with Automated Job Prediction- A resume is a formal document used by individuals to present their backgrounds and skill sets. A resume mainly consists of the individual's educational background, technical skills, work experience, social skills, and awards or publications(if any). This is the first document that describes the individual to an employer or a recruiter. It portrays the individual's image before the interview. Hence, it is essential that one should have an apt resume ready before applying for any company or job. We have proposed a system that will create a formal resume for the user and suggest/predict jobs for the user based on his/her skills. The prediction will help the user in applying for jobs that are aligned with his/her interests and skills. The Resume Builder Application will help users build his/her personal resume. **[8]**

4. ResumeCraft: A Machine Learning-powered Web Platform for Resume Building - This paper introduces ResumeCraft, a web-based platform empowering users to build strong resumes and optimize them for ATS compatibility. ResumeCraft leverages Machine Learning (ML) for data analysis and user guidance, while the user interface is built with Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript for a user-friendly experience. The system allows users to input their personal and professional details through a series of form fields, and provides a real-time preview of the resume design as the user inputs their data. The resume generator uses JavaScript to dynamically populate the preview with the user's input, and allows users to select from a range of pre-designed templates and color schemes to customize the look and feel of their resume. **[9]**

The main reason for designing this web app is the fact that in the competitive job market, the need for a web-based CV generator is paramount to simplify and standardize the CV creation process for job seekers of all backgrounds. Traditional methods of crafting CVs can be time-consuming, error-prone, and daunting for those lacking design skills or familiarity with industry standards. A web-based CV generator provides an intuitive, accessible platform that offers professionally designed templates, automated formatting, real-time previews, and expert content suggestions. This ensures that users can effortlessly create polished, high-quality CVs that effectively showcase their qualifications and experiences, enhancing their chances of securing job interviews and employment opportunities. Additionally, its accessibility from any device allows for flexibility and convenience, meeting the demands of modern job seekers who require efficient and effective tools for career advancement.

Chapter 3 System Analysis

To develop and deploy a Web Application for CV Building, we need to establish a set of system requirements that cover various aspects of the project. The successful development and deployment of the CV generating web application require a comprehensive set of system requirements, encompassing hardware, software, network, and security aspects. These requirements ensure the application is robust, user-friendly, secure, and scalable to handle various user demands.

Hardware Requirements

- Processor: Any modern processor (Intel i3 or equivalent)
- Memory: Minimum 4 GB RAM
- Storage: At least 100 MB free space for temporary data storage

Software Requirements

- Web Browser: Latest versions of Chrome, Firefox, Safari, or Edge
- Frameworks and Libraries: React.js or Angular for dynamic user interfaces, Bootstrap for responsive design

Language

- The language(s) used for development can be just simple as HTML, CSS, JS, e.t.c.

Network Requirements

- Bandwidth: Sufficient bandwidth to handle multiple concurrent users

Performance Requirements

- Scalability: Auto-scaling capabilities to handle varying loads
- Response Time: Average response time should be less than 2 seconds
- Availability: 99.9% uptime with minimal downtime for maintenance

Usability Requirements

- User Interface: Intuitive and user-friendly interface with accessibility features
- Cross-Platform Compatibility: Consistent performance across different operating systems and devices
- Mobile Responsiveness: Fully responsive design for seamless mobile use

By adhering to these system requirements, the CV generating web application will be well-equipped to deliver a reliable, efficient, and secure service that meets the needs of job seekers, recruitment agencies, educational institutions, and other stakeholders.

Chapter 4 System Design

Till now we have clear idea of building a Web applicaton for Building CVs. For which we would be using the basic HTML, CSS and JS, along with the basic use of Bootstrap, that provides a collection of HTML, CSS, and JavaScript components and tools that enable developers to build responsive, mobile-first websites with ease. So we rely only on JavaScript for the functionoing of the whole Web Application and not any other Designing tool.

The Reason of using JS is to show the power and ability that JS provides and stills competes with other languages and framework.

Building a CV generating web application using JavaScript involves several key steps, from initial design to deployment. First, the front-end is developed using a framework like React.js to create a dynamic and responsive user interface. This UI allows users to sign up, log in, and access a dashboard where they can create, edit, and manage their CVs. Users can choose from various templates and input personal information through intuitive forms. The state management is handled using Redux to ensure data consistency across the application.

For the back-end, Node.js with Express.js is used to build the server-side logic. This involves setting up RESTful API endpoints for user authentication, CV management, and template handling. Authentication is secured using JSON Web Tokens (JWT) to manage user sessions. The application stores user data and CV content in a MongoDB database, which allows for flexible and scalable data management. The back-end also handles the generation of CVs in different formats like PDF and Word, utilizing libraries such as pdf-lib or docx.

To manage templates and static assets, AWS S3 is used for storage, providing a scalable and reliable solution for file management. The entire application is containerized using Docker, facilitating consistent environments from development to production. Kubernetes is then employed for orchestration, ensuring the application can scale efficiently to handle varying loads.

Security is paramount, so HTTPS is enforced for all communications, and sensitive data is encrypted both in transit and at rest. Regular security audits and the implementation of best practices in coding help protect against common vulnerabilities. Continuous integration and deployment (CI/CD) pipelines are set up using GitHub Actions, automating the testing and deployment process to ensure that updates can be rolled out smoothly and efficiently.

Finally, monitoring and logging are implemented using Prometheus and Grafana for performance monitoring, and the ELK stack (Elasticsearch, Logstash, Kibana) for logging. This setup allows for real-time monitoring and quick troubleshooting of any issues that arise. By following this approach, a robust, scalable, and user-friendly CV generating web application can be developed using JavaScript.

4.1 MODULES AND LIBRARIES

The Modules that we would be using in building the model are:

1. **HTML-** HTML, or HyperText Markup Language, is the standard language used to create and design documents on the web. It structures web content by using elements and tags to define the layout and format of text, images, links, and other multimedia. HTML is a cornerstone technology of the World Wide Web, alongside CSS (Cascading Style Sheets) and JavaScript.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. [10]

Key Features of HTML:

- **Elements and Tags:**

  HTML documents are composed of elements, which are defined using tags. Tags are enclosed in angle brackets, and they often come in pairs: an opening tag <tagname> and a closing tag </tagname>. Some tags, like <img> and <br>, are self-closing.

- **Document Structure:**

  An HTML document typically starts with a <!DOCTYPE html> declaration, followed by the <html> tag that encloses the entire document. The document is divided into two main sections:

  - Head (<head>): Contains meta-information about the document, such as the title (<title>), character set (<meta>), styles (<style> or links to CSS files), and scripts (<script>).
  - Body (<body>): Contains the actual content of the document, including text, images, links, and other media.

- **Text Formatting:**

  HTML provides a variety of tags to format text, such as <h1> to <h6> for headings, <p> for paragraphs, <a> for links, <strong> and <em> for bold and italic text, respectively.

- **Lists:**

  HTML supports ordered (<ol>) and unordered (<ul>) lists, with list items defined using the <li> tag.

- **Tables:**

  Tables are created using the <table> tag, with rows defined by <tr>, headers by <th>, and data cells by <td>.

- **Forms:**

  HTML forms (<form>) are used to collect user input. They include various input elements like <input>, <textarea>, <select>, and <button>.

- **Media:**

  HTML allows the embedding of images (<img>), audio (<audio>), and video (<video>), as well as other media types.

- Semantic Elements:

  Modern HTML includes semantic elements that provide meaning to the structure of the document, such as <header>, <footer>, <article>, <section>, and <nav>. These elements enhance accessibility and SEO.

- **Links:**

  Hyperlinks are created using the <a> tag, allowing navigation between different web pages and resources.

- **Attributes:**

  HTML tags can have attributes that provide additional information about the elements, such as id, class, src (for images), href (for links), and style (for inline CSS)OpenCV is written natively in C++ and has a template interface that works seamlessly with STL containers.

2. **CSS-** CSS, or Cascading Style Sheets, is a stylesheet language used to describe the presentation and design of HTML documents. It controls the layout, colors, fonts, and overall visual appearance of web pages, allowing developers to separate content (HTML) from presentation (CSS)

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility;[further explanation needed] provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. [11]

Key Features of CSS:

- **Selectors:**

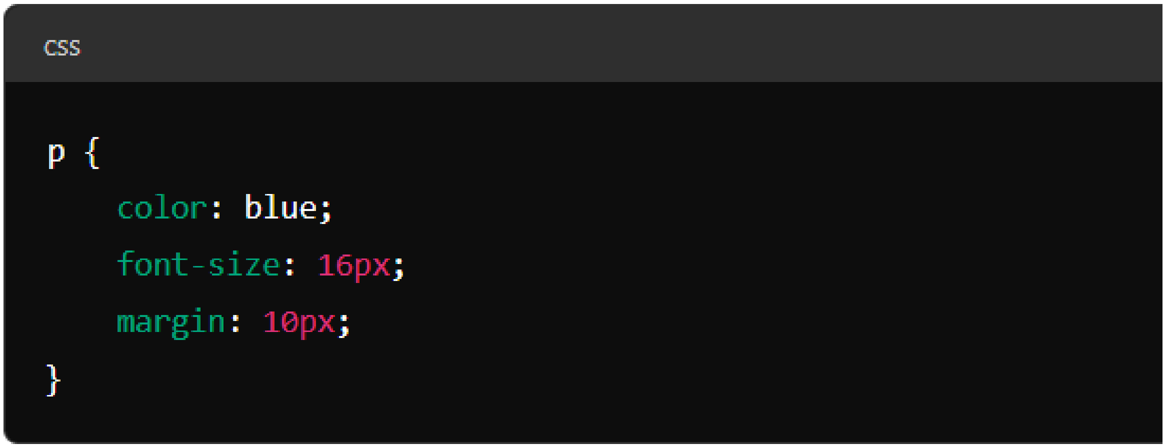
  CSS selectors are used to target HTML elements that you want to style. Common types of selectors include:

  - Element Selector: Targets elements by their tag name (e.g., p { }).
  - Class Selector: Targets elements by their class attribute (e.g., .classname { }).
  - ID Selector: Targets elements by their id attribute (e.g., #idname { }).

- ➢ Attribute Selector: Targets elements by their attributes (e.g., [type="text"] { }).

- **Properties and Values:**

  CSS uses properties and values to define the styles for elements. For example:



```css
p {
    color: blue;
    font-size: 16px;
    margin: 10px;
}
```

Figure 4.1 Describing CSS property

In this example, color, font-size, and margin are properties, and blue, 16px, and 10px are their respective values.

- **Cascading and Specificity:**

The term "cascading" in CSS means that styles can cascade from one stylesheet to another, allowing multiple stylesheets to be combined. CSS follows a specific order of precedence:

- ➢ Inline styles (inside an HTML element) have the highest precedence.
- ➢ Internal styles (within a <style> tag in the HTML document) have medium precedence.
- ➢ External styles (linked from an external CSS file) have the lowest precedence.
- ➢ Specificity determines which styles are applied when there are conflicting rules. Inline styles have the highest specificity, followed by IDs, classes, and finally element selectors.

- **Box Model:**

  The CSS box model describes the rectangular boxes that are generated for elements in the document tree. It consists of:

    - Content: The actual content of the element (text, image, etc.).
    - Padding: The space between the content and the border.
    - Border: The border that surrounds the padding (if any).
    - Margin: The space outside the border.

- **Layouts:**

  CSS provides various layout techniques:

  - Floats: Used to float elements to the left or right of a container.
  - Flexbox: A layout model for creating flexible and responsive layouts.
  - Grid: A powerful 2-dimensional system for creating complex layouts.

- **Responsive Design:**

  CSS enables responsive design to ensure web pages look good on all devices (desktops, tablets, and mobile phones). Media queries are used to apply different styles for different screen sizes:

3. **JavaScript-** JavaScript is a versatile, high-level programming language that is widely used for creating interactive and dynamic web pages. Initially developed by Netscape in 1995, JavaScript has evolved into one of the core technologies of web development, alongside HTML and CSS. It enables developers to enhance the user experience by allowing web pages to respond to user actions, manipulate the DOM (Document Object Model), and communicate asynchronously with servers.

JavaScript (/ˈdʒɑːvəskrɪpt/), often abbreviated as JS, is a programming language and core technology of the Web, alongside HTML and CSS. 99% of websites use JavaScript on the client side for webpage behavior. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[12]

It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).[13]

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

Key Features of JavaScript:

- **Dynamic Typing:**

  JavaScript is dynamically typed, meaning variables do not require a predefined type and can hold any type of data at any time. This provides flexibility but also requires careful handling to avoid type-related errors.

- **Object-Oriented:**

  JavaScript supports object-oriented programming, allowing the creation of objects to model real-world entities. It uses prototypes rather than classical inheritance, making it unique compared to traditional OOP languages like Java or C++.

- **First-Class Functions:**

  Functions in JavaScript are first-class citizens, meaning they can be assigned to variables, passed as arguments, and returned from other functions. This feature is fundamental for functional programming and higher-order functions.

- **Event-Driven:**

  JavaScript can respond to user interactions such as clicks, form submissions, and keyboard events. Event handling is crucial for creating interactive web applications.

- **Asynchronous Programming:**

  JavaScript supports asynchronous operations using callbacks, promises, and the async/await syntax. This is essential for tasks like fetching data from APIs without blocking the main execution thread.

- **Cross-Platform:**

  JavaScript runs in web browsers, making it inherently cross-platform. Additionally, with the advent of environments like Node.js, JavaScript can also be used for server-side development.

- **Wide Ecosystem:**

  JavaScript has a vast ecosystem of libraries and frameworks, such as React, Angular, and Vue.js for front-end development, and Node.js for back-end development. Tools like npm (Node Package Manager) facilitate easy management of dependencies.

Key Concepts:

- **Variables and Data Types:**

  JavaScript supports various data types including numbers, strings, booleans, objects, arrays, and special values like null and undefined. Variables can be declared using var, let, or const, with let and const being the preferred options due to block scoping.

- **Functions:**

  Functions are defined using the function keyword, and ES6 introduced arrow functions for a more concise syntax. Functions can be anonymous, and higher-order functions can take other functions as arguments or return them.

- **Objects and Arrays:**

  Objects are collections of key-value pairs, and arrays are ordered collections of values. Both are essential for managing data structures in JavaScript.

- **Control Structures:**

  JavaScript includes traditional control structures such as if statements, for loops, while loops, and switch statements, allowing for complex decision-making and iteration.

- **DOM Manipulation:**

  JavaScript interacts with HTML through the DOM, enabling dynamic updates to the content and structure of web pages. Methods like getElementById, querySelector, and createElement are commonly used for DOM manipulation.

- **Event Handling:**

JavaScript handles user interactions through events. Event listeners can be added to elements to execute code in response to user actions like clicks, key presses, and form submissions.

- **Asynchronous Programming:**
  Asynchronous operations are crucial for web applications, allowing tasks like data fetching, timers, and I/O operations to run without blocking the main thread. Promises and async/await syntax simplify handling asynchronous code.

- **Modules:**
  ES6 introduced modules, allowing code to be split into reusable pieces. The import and export keywords enable the use of modules, promoting better organization and maintainability of code.

4. **BootsStrap** - Bootstrap is a popular open-source front-end framework that simplifies the development of responsive and visually appealing web applications. Developed by Twitter and released in 2011, Bootstrap provides a collection of CSS and JavaScript tools designed to make the process of creating web pages faster and more efficient. It is widely used for its ease of use, flexibility, and ability to create consistent layouts across different browsers and devices.

Bootstrap (formerly Twitter Bootstrap) is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

As of May 2023, Bootstrap is the 17th most starred project (4th most starred library) on GitHub, with over 164,000 stars.[14] According to W3Techs, Bootstrap is used by 19.2% of all websites.

Bootstrap is an HTML, CSS and JS library that focuses on simplifying the development of informative web pages (as opposed to web applications). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking.

Key Features of Bootstrap:

- **Responsive Design:**

  Bootstrap includes a responsive grid system that allows developers to create fluid layouts that adapt to different screen sizes. This grid system is based on a 12-column layout and includes classes for different breakpoints to handle various devices such as desktops, tablets, and mobile phones.

- **Pre-styled Components:**

  Bootstrap comes with a variety of pre-styled components such as buttons, forms, tables, navigation bars, alerts, modals, and carousels. These components are designed to be easily customized and integrated into web pages, saving time on basic styling.

- **CSS Classes:**

  Bootstrap provides a rich set of CSS classes that simplify styling elements. These classes cover a wide range of functionalities, including typography, spacing, alignment, and visibility.

- **JavaScript Plugins:**

  Bootstrap includes a set of JavaScript plugins that enhance the functionality of web pages. These plugins, based on jQuery, enable features such as modals, tooltips, popovers, dropdowns, and carousels. They can be easily activated using data attributes or JavaScript.

- **Customizable:**

  Bootstrap is highly customizable, allowing developers to override default styles and create custom themes. The framework uses Sass (Syntactically Awesome Style Sheets) for its styles, enabling the use of variables, nested rules, and mixins to create custom designs.

- **Consistent Design:**
  By using Bootstrap, developers can ensure a consistent design across different parts of their web application. This consistency is achieved through a unified set of styles and components.

Key Concepts:

- **Grid System:**
  The grid system in Bootstrap is a flexible and responsive layout structure. It uses a series of containers, rows, and columns to layout and align content. Columns can be combined to create wider columns, and they will stack on smaller devices.

- **Responsive Utilities:**
  Bootstrap includes utility classes for showing, hiding, or altering the appearance of elements based on the viewport size. These classes include .d-none, .d-block, .d-md-none, and others.

- **Predefined Classes:**
  Bootstrap provides a wide range of predefined classes for typography, forms, buttons, images, and more. These classes make it easy to apply consistent styling across different elements.

- **JavaScript Components:**
  Bootstrap's JavaScript components enhance interactivity. These components are activated using data attributes or JavaScript code.
- **Customization:**
  Developers can customize Bootstrap by modifying the Sass variables and recompiling the CSS. This allows for a tailored look and feel that matches the branding and design requirements of a project.

Bootstrap is a powerful tool for web development, offering a comprehensive set of styles and components that streamline the creation of responsive and modern web pages. Its ease of use, flexibility, and extensive documentation make it a popular choice among developers for both small projects and large-scale web applications.

4.2 Model Architecture:

The Architecture and proposed flowchart for building the Web Application is shown in form of flowchart diagram below.

This Basic circuit Diagram shows the flow data and exact execution of Web Application:

The basic blocks show the modules and data used and arrow reflect the flow of process →

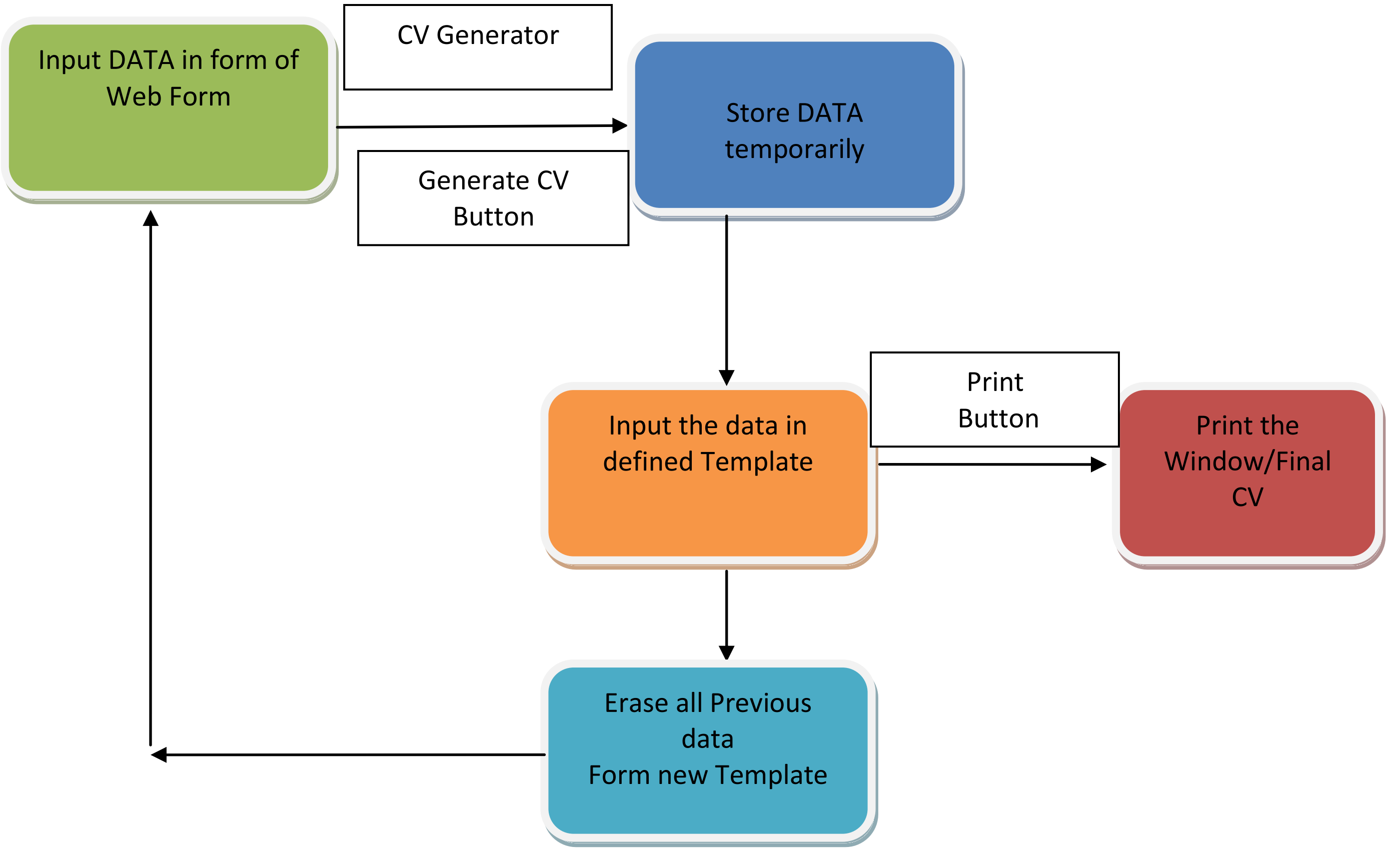


Figure 4.2.1 Flow Diagram for CV Generator

4.3 CV v/s Resume

When applying for jobs, you might come across the terms "resume" and "CV" (Curriculum Vitae). Although they are often used interchangeably, they serve different purposes and have distinct formats. Understanding the differences between a resume and a CV is crucial for job seekers to present their qualifications effectively.

4.3.1 What is CV?

CV is a comprehensive document that provides an in-depth overview of your academic and professional history. It includes detailed information about your education, work experience, research, publications, presentations, awards, and other achievements.

The CV is often used in academic, research, medical, and scientific fields.

CVs can vary in length, ranging from two pages to several pages, depending on your experience and accomplishments.

CVs are commonly used in Europe, the Middle East, Africa, and Asia for job applications, as well as in academic and research settings worldwide.

4.3.2 What is Resume?

A resume is a concise document, typically one to two pages long, that summarizes your education, work experience, skills, and achievements.

Its primary purpose is to provide potential employers with a quick overview of your qualifications tailored to a specific job or industry.

Resumes are commonly used in the United States and Canada for most job applications.

4.3.3 Key Differnces

- **Length:**

➢ **Resume:** Typically one to two pages, focusing on brevity and relevance to the job applied for.

➢ **CV:** Length can vary, often running several pages to comprehensively cover academic and professional achievements.

- **Content:**

➢ **Resume:** Includes sections such as contact information, professional summary or objective, work experience, education, skills, and sometimes additional sections like certifications, volunteer work, or interests.

➢ **CV:** Includes detailed sections such as contact information, personal statement, detailed education history, work experience, research experience, publications, presentations, awards, honors, professional affiliations, and other relevant activities.

- **Focus:**

➢ **Resume:** Tailored to the specific job application, emphasizing skills and experiences relevant to the position.

➢ **CV:** Provides a thorough and detailed account of one's entire academic and professional history, often not tailored to a specific position.

- **Usage:**

➢ **Resume:** Used for most job applications outside of academia, particularly in North America.

➢ **CV:** Used for academic, research, and medical positions, and in regions where CVs are the norm for job applications.

4.3.4 Structural Differences

Resume Structure:

- Contact Information: Name, phone number, email address, LinkedIn profile (optional).
- Professional Summary or Objective: A brief statement highlighting your career goals and key qualifications.
- Work Experience: List of jobs held, with bullet points outlining key responsibilities and achievements. Typically, this section is in reverse chronological order.
- Education: Degrees obtained, institutions attended, and graduation dates.
- Skills: A list of relevant skills, both technical and soft skills.
- Additional Sections (Optional): Certifications, volunteer work, languages, projects, interests, etc.

CV Structure:

- Contact Information: Name, phone number, email address, LinkedIn profile (optional).
- Personal Statement: A detailed overview of your academic interests, career goals, and professional aspirations.
- Education: Detailed information on all degrees obtained, institutions attended, graduation dates, thesis titles, and relevant coursework.
- Work Experience: Comprehensive list of all relevant professional positions held, with detailed descriptions of responsibilities and accomplishments.
- Research Experience: Detailed account of research projects undertaken, methodologies used, and results obtained.
- Publications: A comprehensive list of all published works, including articles, papers, books, and other publications.
- Presentations: A list of conferences, seminars, and other events where you have presented your work.
- Awards and Honors: Detailed account of all academic and professional awards and recognitions.
- Professional Affiliations: Membership in professional organizations and associations.

- Additional Sections: Grants, fellowships, certifications, skills, languages, and any other relevant information.

The Resumes are concise and targeted, suitable for most job applications, focusing on relevant skills and experiences. CVs, on the other hand, are detailed and comprehensive, ideal for academic, research, and scientific positions, providing an in-depth look at your entire career. Understanding these differences ensures that you present your qualifications in the best possible light, depending on the job or position you are applying for.

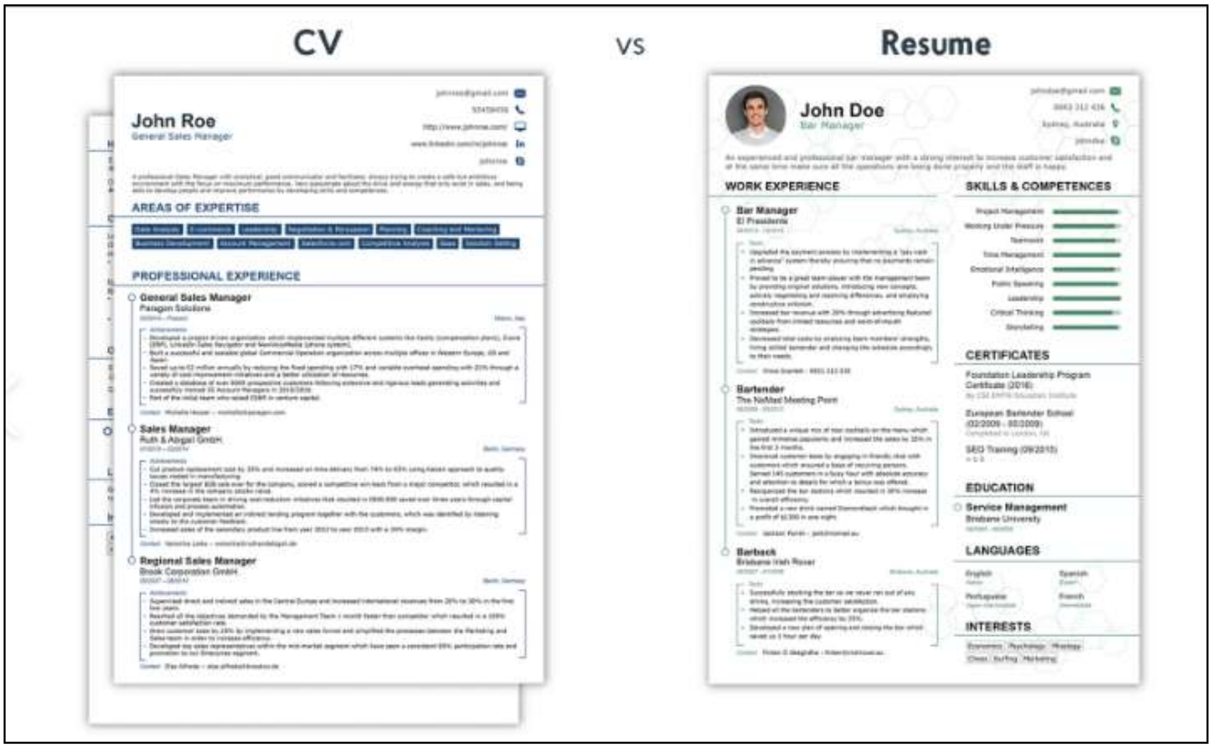


Figure4.3.4 Visual Difference between CV and Resume

4.4 Advantages of CV over Resume:

A CV (Curriculum Vitae) provides a comprehensive and detailed overview of an individual's academic and professional history, making it ideal for positions in academia, research, and fields requiring extensive experience. This detailed document showcases a candidate's full breadth of qualifications and accomplishments, which can be advantageous in fields that value a thorough understanding of one's background.

Compared to a resume, a CV allows for the inclusion of detailed sections such as publications, presentations, research experience, and professional affiliations. This level of detail can highlight a candidate's expertise and contributions to their field, providing a richer picture of their qualifications and achievements.

Additionally, a CV's flexibility in length means that candidates can provide a complete record of their career, without the constraints of brevity that a resume imposes. This can be particularly beneficial for seasoned professionals with extensive experience, ensuring that all relevant information is included and nothing important is omitted.

4.5 IMPLEMENTATION AND WORKING

4.5.1 Implementation:

Building a web-based CV builder involves implementing several key components that work together to provide a seamless user experience for creating, customizing, and generating professional CVs. The development process begins with the front-end interface, which is designed using HTML for structure, CSS for styling, and JavaScript for interactivity. HTML forms the backbone of the application, creating the structure for the input fields where users can enter their personal information, education, work experience, skills, and other relevant details. CSS, along with frameworks like Bootstrap, ensures the interface is visually appealing and responsive across different devices, enhancing usability.

JavaScript plays a crucial role in adding dynamic functionality to the application. It manages user inputs, handles form submissions, and updates the CV preview in real-time as users fill out their details. React.js, a popular JavaScript library, is often used to create reusable components for the form fields and CV preview, facilitating a modular and maintainable codebase, ensuring the application remains responsive to user interactions.

While a database is not required for this implementation, data can be temporarily stored in-memory or using file-based storage to facilitate the CV generation process. For CV generation we would be directly printing the designed CV using the print function of JS to print the final window form of CV image on the Screen window, which are further integrated to convert HTML content into PDF files. This allows users to download their completed CVs in a professional format.

This comprehensive approach to building a web-based CV builder results in a robust, user-friendly application that enables users to effortlessly create and customize their CVs.

4.5.2. Working:

Working Theory of a Web-Based CV Builder

1. User Interface (UI) Design

- ➢ Form Inputs: Provides fields for user to input personal, educational, and professional information.
- ➢ Template Selection: Allows users to choose from various CV templates.

2. Client-Side Development

- ➢ HTML/CSS: Structures the form and styles the UI.
- ➢ JavaScript (JS): Manages form interactions, data validation, and dynamic content updates.

3. State Management

- ➢ Data Storage: Uses JavaScript objects to store user input data temporarily.
- ➢ Live Preview: Updates the CV preview in real-time as users fill out the form.

4. Form Handling

- ➢ Input Validation: Validates user inputs for correct data formats (e.g., email, phone number).
- ➢ Data Binding: Binds form inputs to state variables to reflect changes instantly.

6. PDF Generation

- ➢ HTML to PDF Conversion: Uses libraries like `pdf-lib` or `html2pdf.js` to convert HTML templates to PDF.
- ➢ Dynamic Data Insertion: Inserts user-provided data into predefined CV templates before PDF conversion.

By following these steps, a web-based CV builder can effectively collect user data, dynamically generate CVs, and provide a user-friendly experience, ensuring that users can create professional CVs efficiently.

4.6 Features

Features of Web Application Based CV Generator include:

1. User Registration and Login:

- Secure user authentication and authorization.
- Social media login options (e.g., Google, LinkedIn).

2. Profile Management:

- User profile customization and management.
- Option to upload a profile picture.

3. Template Selection:

- Multiple professionally designed CV templates.
- Option to preview templates before selection.

4. Easy-to-Use Interface:

- Intuitive and user-friendly design.
- Drag-and-drop functionality for rearranging sections.

5. Form Input Fields:

- Sections for personal information, education, work experience, skills, and more.
- Customizable sections to add additional information (e.g., certifications, awards).

6. Real-Time Preview:

- Live preview of the CV as data is entered.
- Instant updates reflecting changes made by the user.

7. Mobile Compatibility:

- Responsive design for mobile and tablet use.
- Mobile app version for on-the-go CV creation.

8. Formating Features:

- Customizable number of fields.
- Ability to adjust the number of inputs for few Fields.

9. Multiple CV Versions:

- Option to create and manage multiple CV versions.
- Easy duplication of existing CVs for modification.

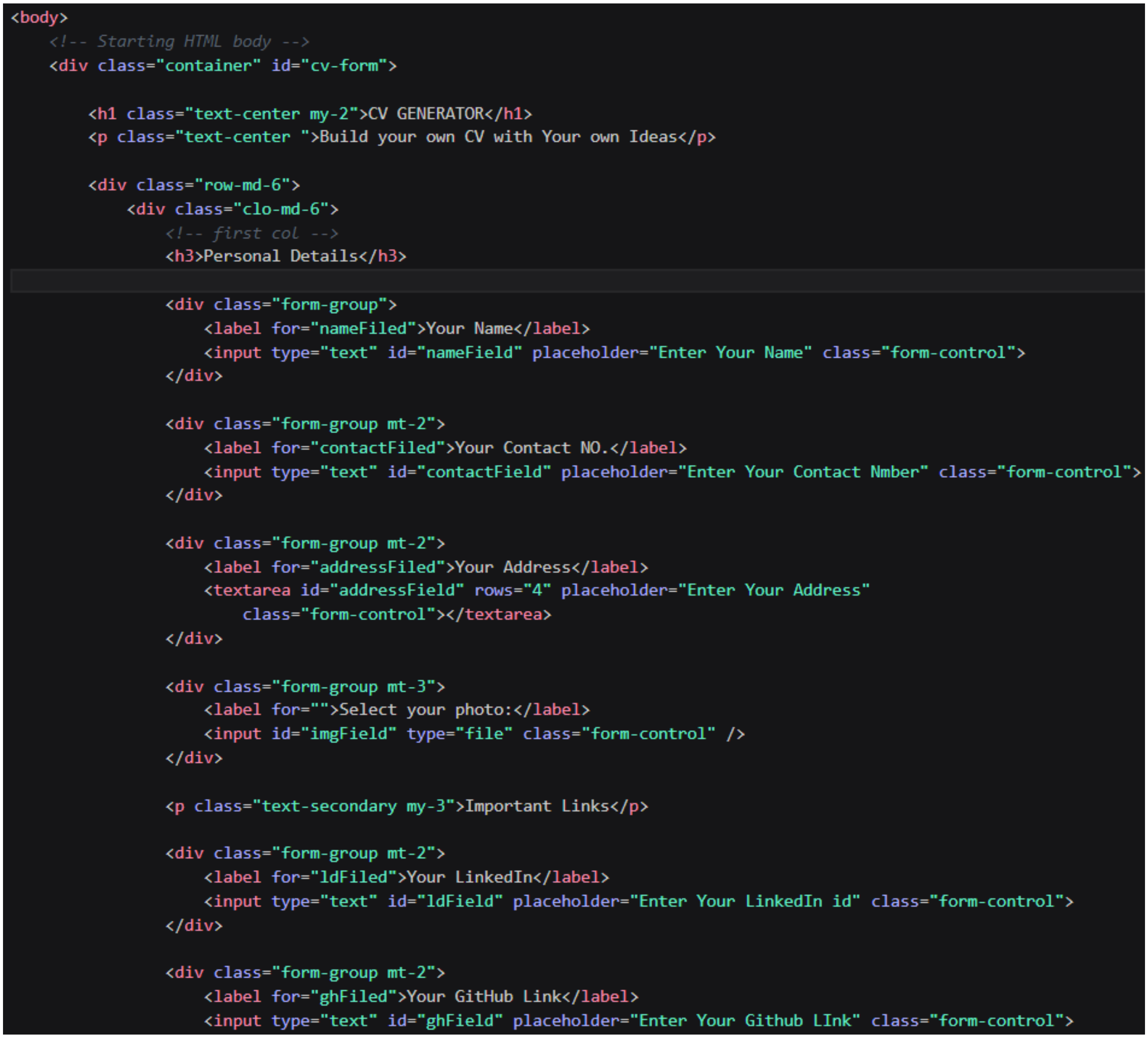
10. Export and Download:

- Download CVs in various formats (PDF, DOCX).
- Print-ready CV versions.

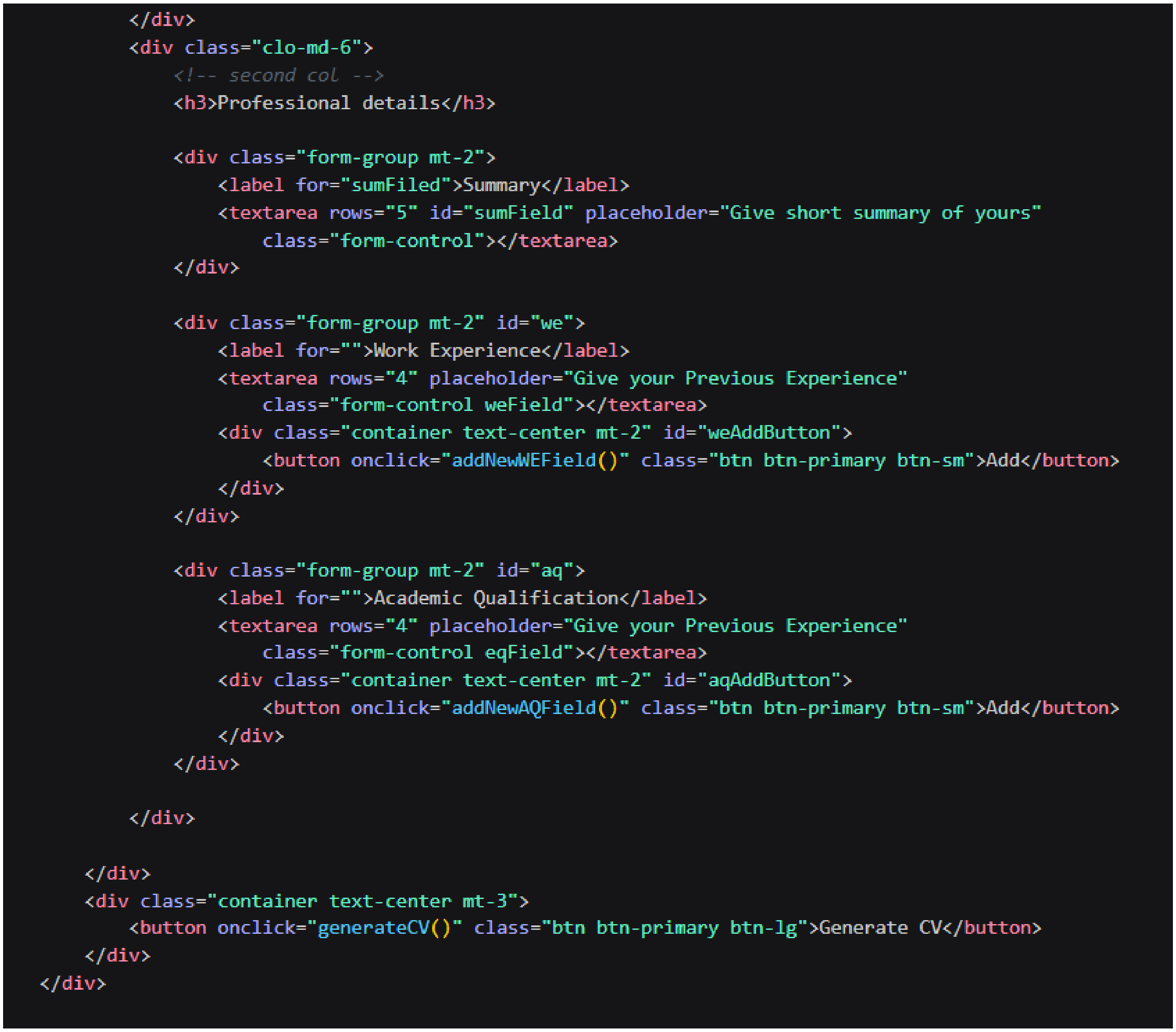
These features collectively contribute to the development of effective and efficient object detection systems using Python-based image processing and computer vision techniques.

Chapter 5 Coding:

A. Building a simple HTML form that will take input from user:



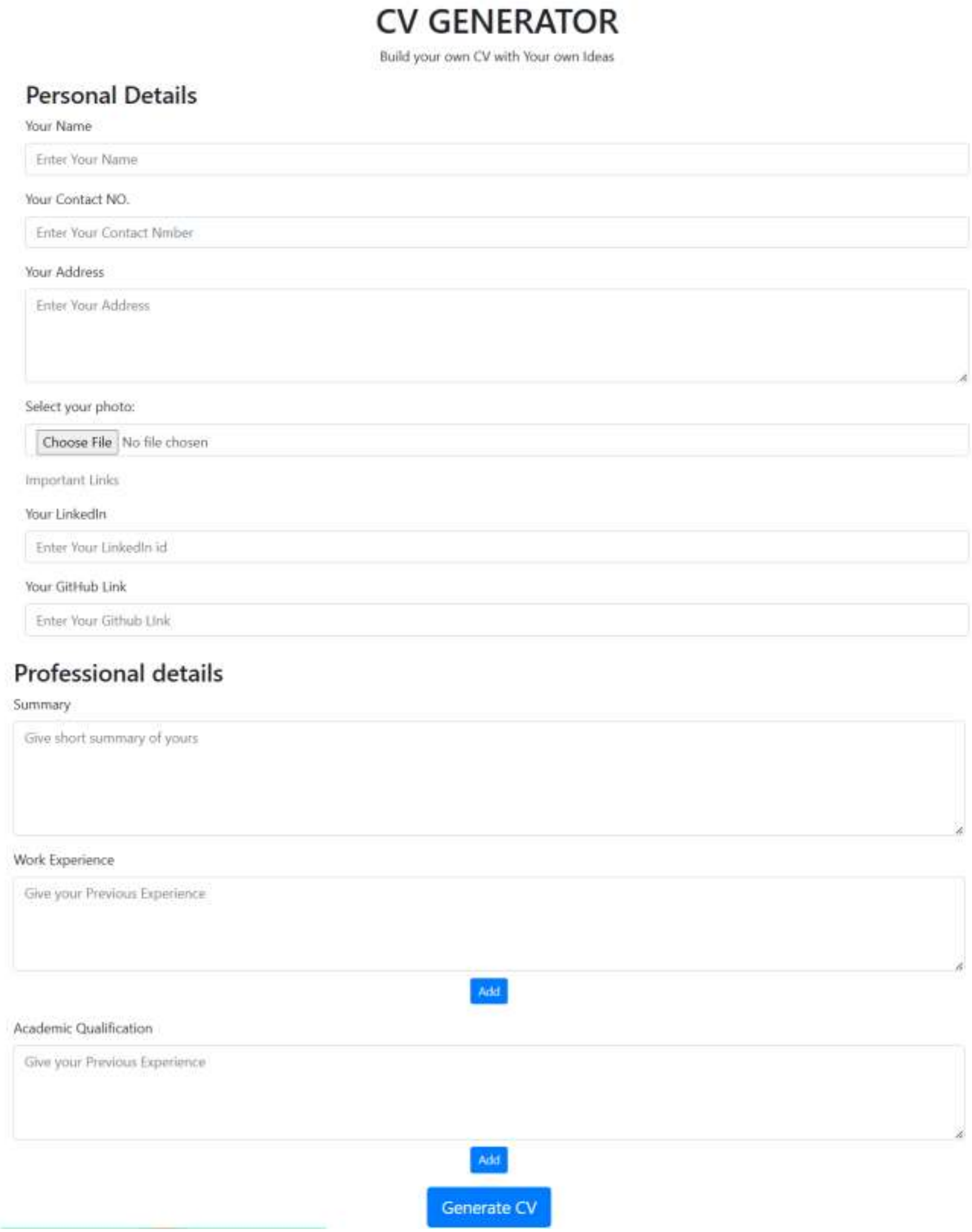
```
<body>
    <!-- Starting HTML body -->
    <div class="container" id="cv-form">

        <h1 class="text-center my-2">CV GENERATOR</h1>
        <p class="text-center ">Build your own CV with Your own Ideas</p>

        <div class="row-md-6">
            <div class="clo-md-6">
                <!-- first col -->
                <h3>Personal Details</h3>

                <div class="form-group">
                    <label for="nameFiled">Your Name</label>
                    <input type="text" id="nameField" placeholder="Enter Your Name" class="form-control">
                </div>

                <div class="form-group mt-2">
                    <label for="contactFiled">Your Contact NO.</label>
                    <input type="text" id="contactField" placeholder="Enter Your Contact Nmber" class="form-control">
                </div>

                <div class="form-group mt-2">
                    <label for="addressFiled">Your Address</label>
                    <textarea id="addressField" rows="4" placeholder="Enter Your Address"
                        class="form-control"></textarea>
                </div>

                <div class="form-group mt-3">
                    <label for="">Select your photo:</label>
                    <input id="imgField" type="file" class="form-control" />
                </div>

                <p class="text-secondary my-3">Important Links</p>

                <div class="form-group mt-2">
                    <label for="ldFiled">Your LinkedIn</label>
                    <input type="text" id="ldField" placeholder="Enter Your LinkedIn id" class="form-control">
                </div>

                <div class="form-group mt-2">
                    <label for="ghFiled">Your GitHub Link</label>
                    <input type="text" id="ghField" placeholder="Enter Your Github LInk" class="form-control">
```



```
        </div>
        <div class="clo-md-6">
            <!-- second col -->
            <h3>Professional details</h3>

            <div class="form-group mt-2">
                <label for="sumFiled">Summary</label>
                <textarea rows="5" id="sumField" placeholder="Give short summary of yours"
                    class="form-control"></textarea>
            </div>

            <div class="form-group mt-2" id="we">
                <label for="">Work Experience</label>
                <textarea rows="4" placeholder="Give your Previous Experience"
                    class="form-control weField"></textarea>
                <div class="container text-center mt-2" id="weAddButton">
                    <button onclick="addNewWEField()" class="btn btn-primary btn-sm">Add</button>
                </div>
            </div>

            <div class="form-group mt-2" id="aq">
                <label for="">Academic Qualification</label>
                <textarea rows="4" placeholder="Give your Previous Experience"
                    class="form-control eqField"></textarea>
                <div class="container text-center mt-2" id="aqAddButton">
                    <button onclick="addNewAQField()" class="btn btn-primary btn-sm">Add</button>
                </div>
            </div>

        </div>

    </div>
    <div class="container text-center mt-3">
        <button onclick="generateCV()" class="btn btn-primary btn-lg">Generate CV</button>
    </div>
</div>
```

Figure 5.1 HTML form for input

OUTPUT:



CV GENERATOR

Build your own CV with Your own Ideas

Personal Details

Your Name

Enter Your Name

Your Contact NO.

Enter Your Contact Nmber

Your Address

Enter Your Address

Select your photo:

Choose File No file chosen

Important Links

Your LinkedIn

Enter Your LinkedIn id

Your GitHub Link

Enter Your Github Link

Professional details

Summary

Give short summary of yours

Work Experience

Give your Previous Experience

Add

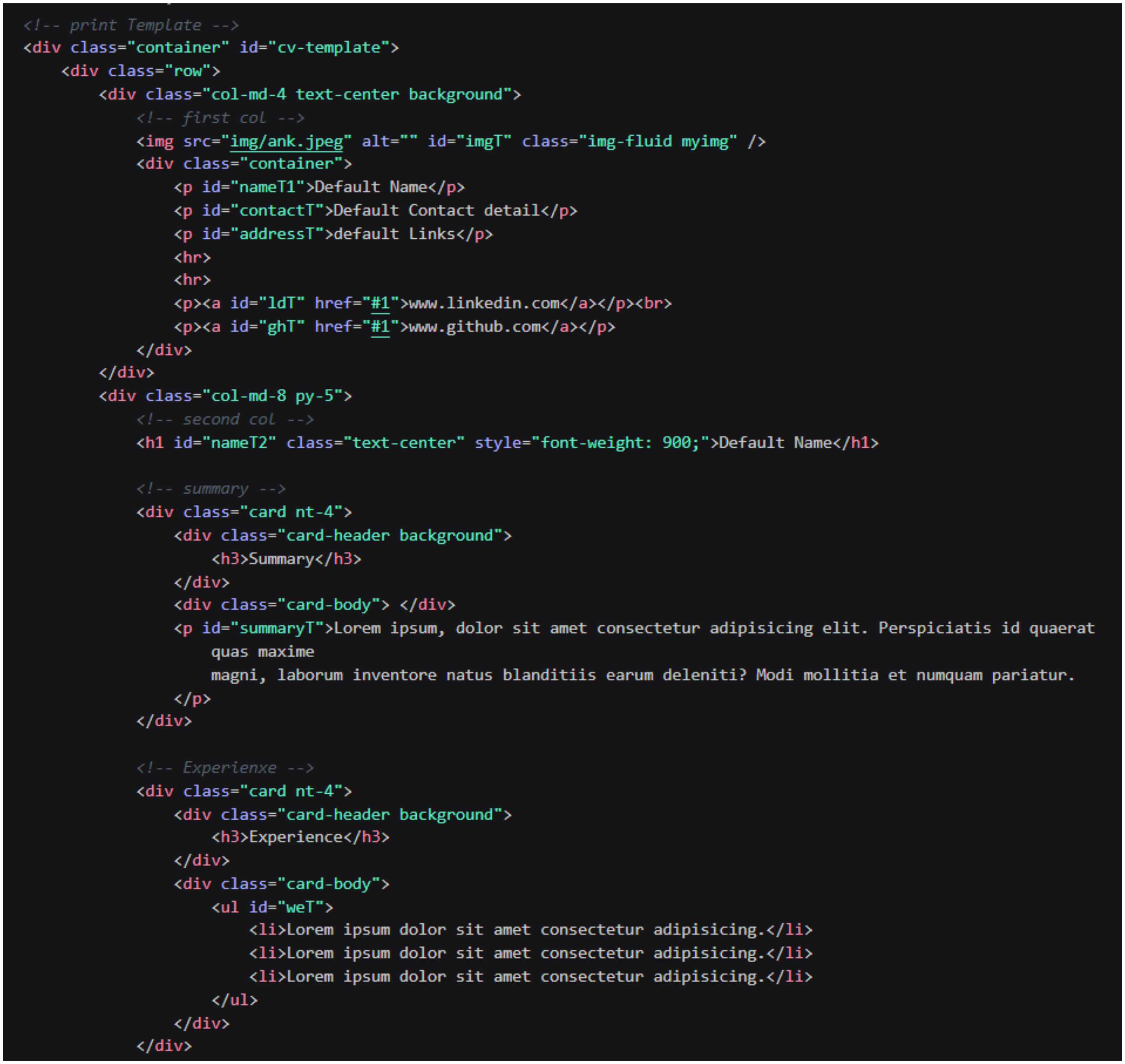
Academic Qualification

Give your Previous Experience

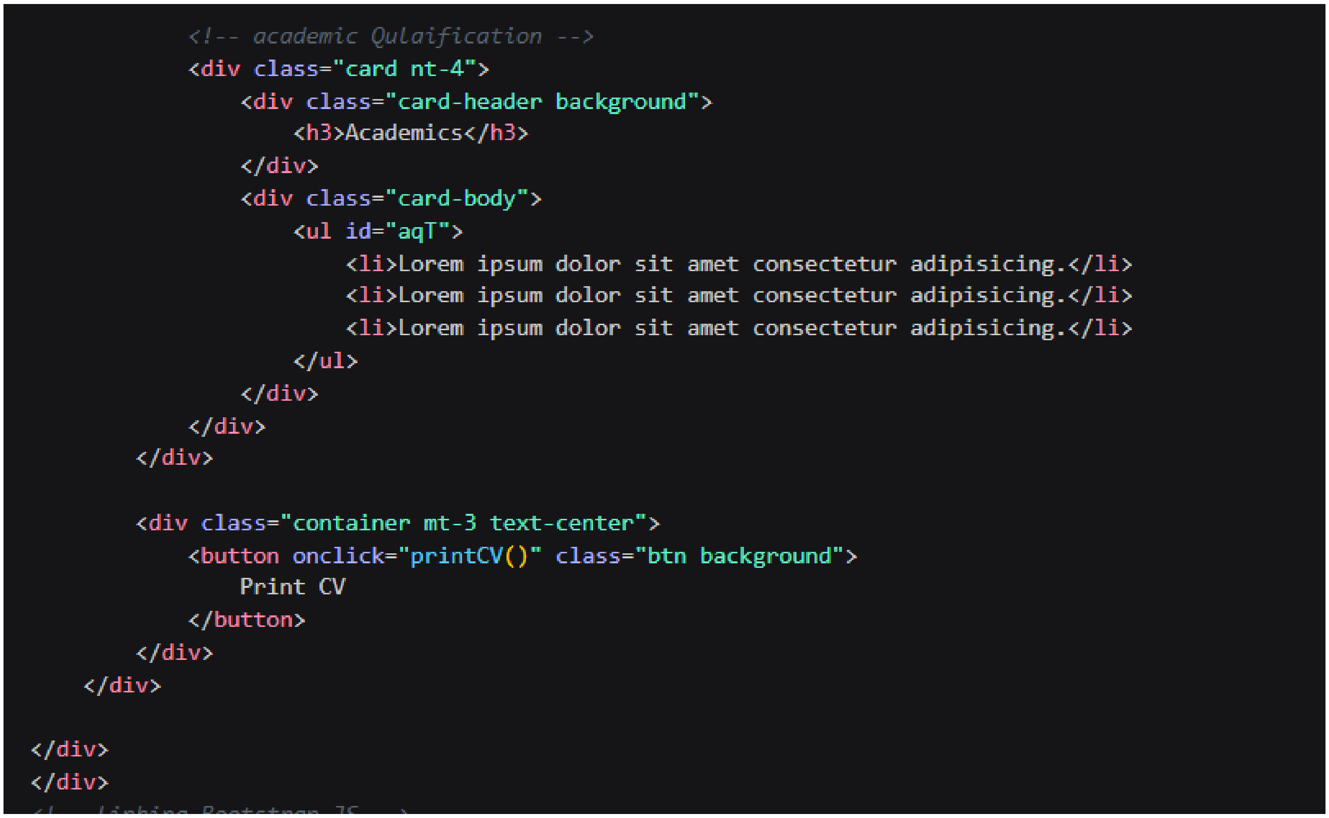
Add

Generate CV

B.HTML scripting for CV template:



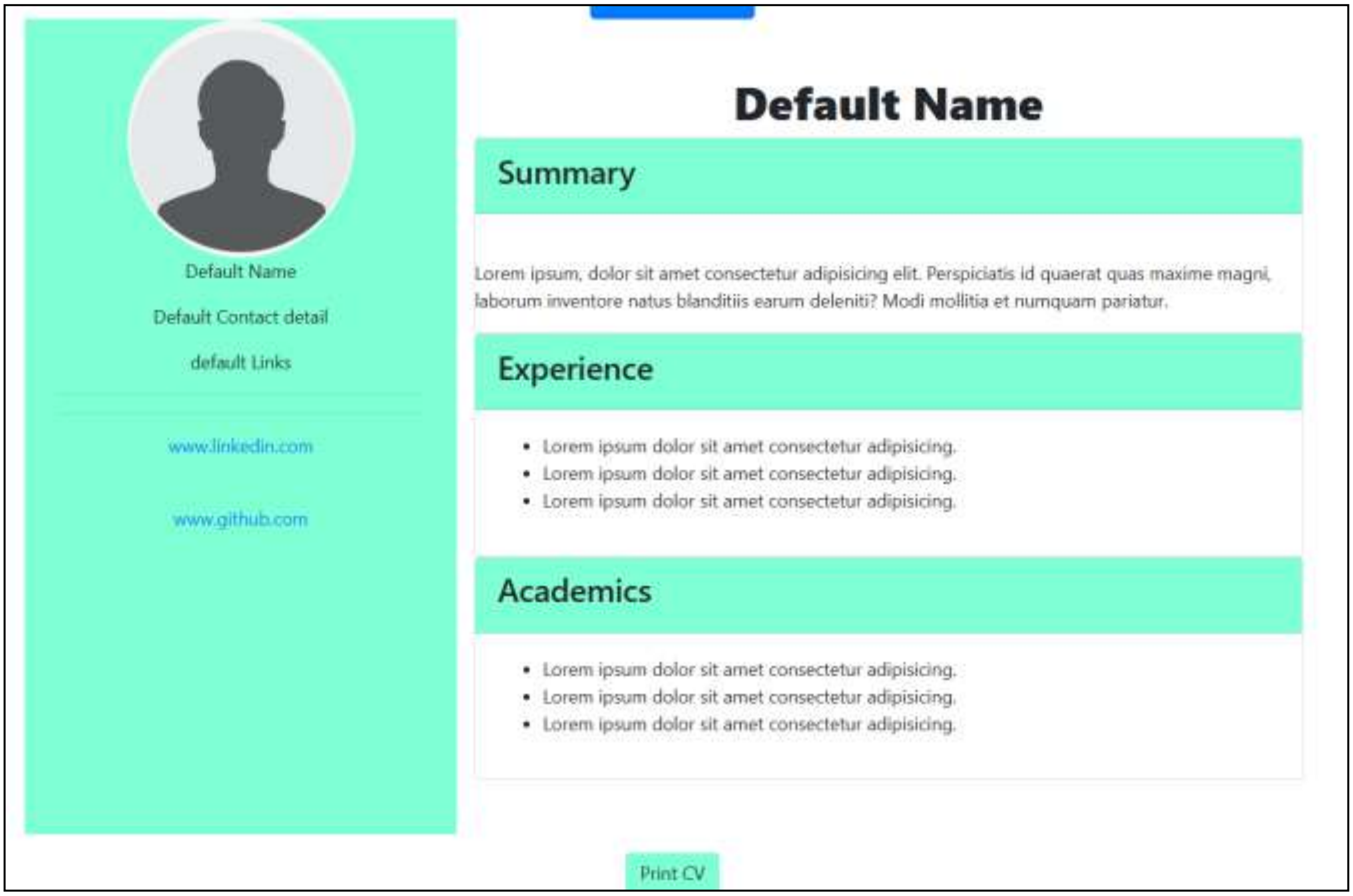
```html
<!-- print Template -->
<div class="container" id="cv-template">
    <div class="row">
        <div class="col-md-4 text-center background">
            <!-- first col -->
            <img src="img/ank.jpeg" alt="" id="imgT" class="img-fluid myimg" />
            <div class="container">
                <p id="nameT1">Default Name</p>
                <p id="contactT">Default Contact detail</p>
                <p id="addressT">default Links</p>
                <hr>
                <hr>
                <p><a id="ldT" href="#1">www.linkedin.com</a></p><br>
                <p><a id="ghT" href="#1">www.github.com</a></p>
            </div>
        </div>
        <div class="col-md-8 py-5">
            <!-- second col -->
            <h1 id="nameT2" class="text-center" style="font-weight: 900;">Default Name</h1>

            <!-- summary -->
            <div class="card nt-4">
                <div class="card-header background">
                    <h3>Summary</h3>
                </div>
                <div class="card-body"> </div>
                <p id="summaryT">Lorem ipsum, dolor sit amet consectetur adipisicing elit. Perspiciatis id quaerat
                    quas maxime
                    magni, laborum inventore natus blanditiis earum deleniti? Modi mollitia et numquam pariatur.
                </p>
            </div>

            <!-- Experienxe -->
            <div class="card nt-4">
                <div class="card-header background">
                    <h3>Experience</h3>
                </div>
                <div class="card-body">
                    <ul id="weT">
                        <li>Lorem ipsum dolor sit amet consectetur adipisicing.</li>
                        <li>Lorem ipsum dolor sit amet consectetur adipisicing.</li>
                        <li>Lorem ipsum dolor sit amet consectetur adipisicing.</li>
                    </ul>
                </div>
            </div>
```



```
            <!-- academic Qulaification -->
            <div class="card nt-4">
                <div class="card-header background">
                    <h3>Academics</h3>
                </div>
                <div class="card-body">
                    <ul id="aqT">
                        <li>Lorem ipsum dolor sit amet consectetur adipisicing.</li>
                        <li>Lorem ipsum dolor sit amet consectetur adipisicing.</li>
                        <li>Lorem ipsum dolor sit amet consectetur adipisicing.</li>
                    </ul>
                </div>
            </div>
        </div>

        <div class="container mt-3 text-center">
            <button onclick="printCV()" class="btn background">
                Print CV
            </button>
        </div>
    </div>

</div>
</div>
<!-- Linking Bootstrap JS -->
```

Figure 5.2 HTML for CV template

OUTPUT:



Default Name

Default Name

Default Contact detail

default Links

www.linkedin.com

www.github.com

Summary

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Perspiciatis id quaerat quas maxime magni, laborum inventore natus blanditiis earum deleniti? Modi mollitia et numquam pariatur.

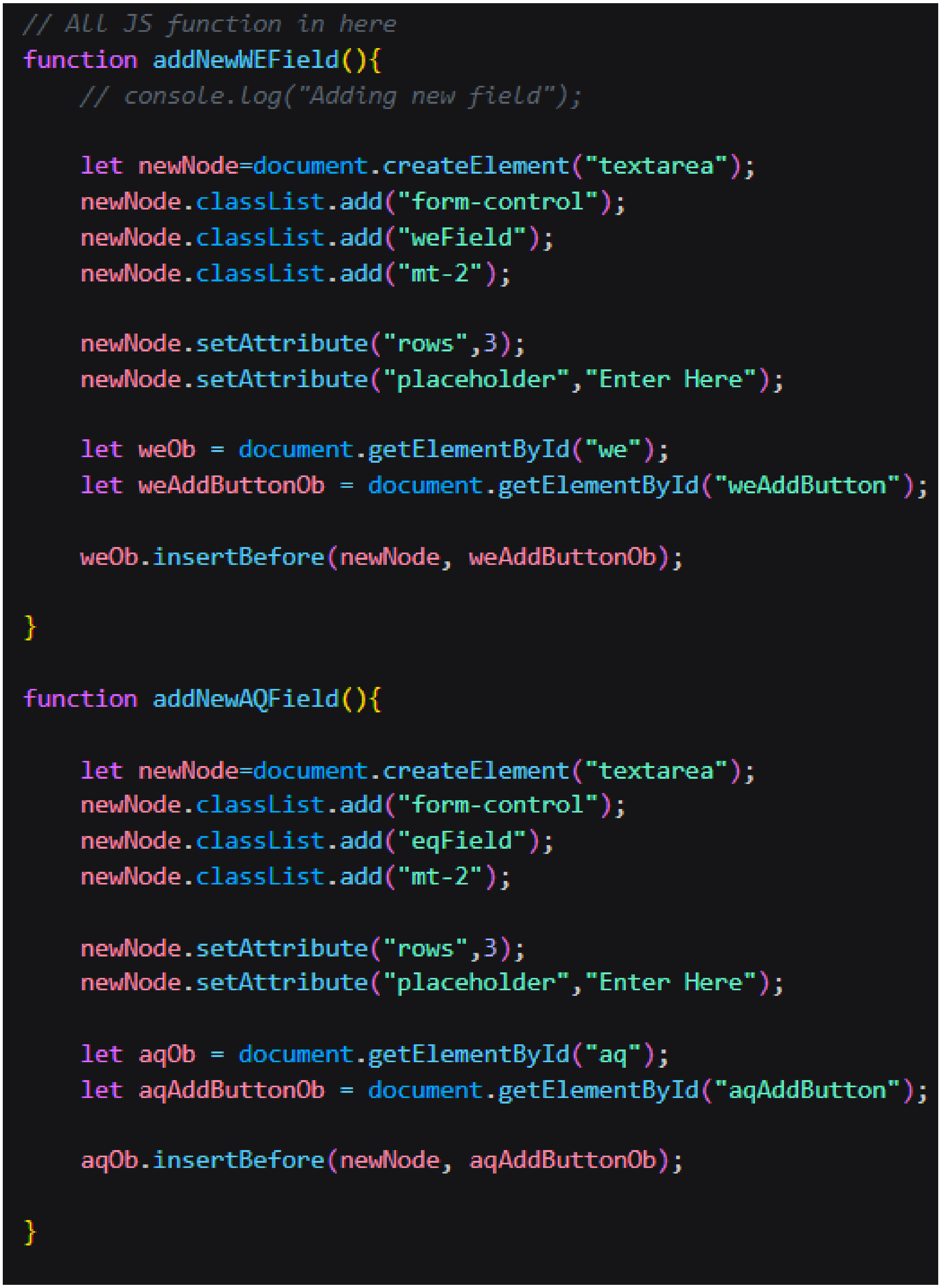
Experience

- Lorem ipsum dolor sit amet consectetur adipisicing.
- Lorem ipsum dolor sit amet consectetur adipisicing.
- Lorem ipsum dolor sit amet consectetur adipisicing.

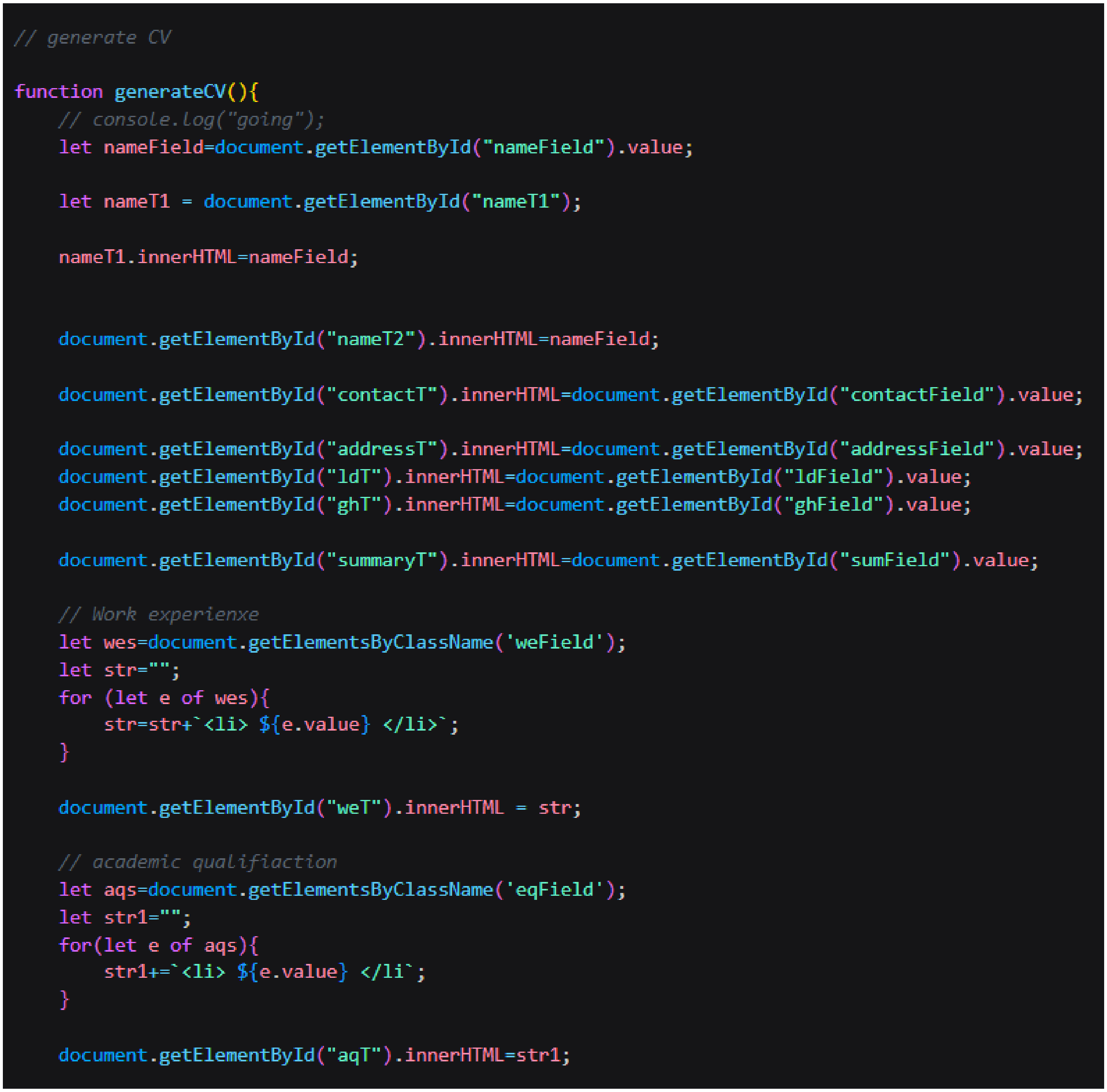
Academics

- Lorem ipsum dolor sit amet consectetur adipisicing.
- Lorem ipsum dolor sit amet consectetur adipisicing.
- Lorem ipsum dolor sit amet consectetur adipisicing.

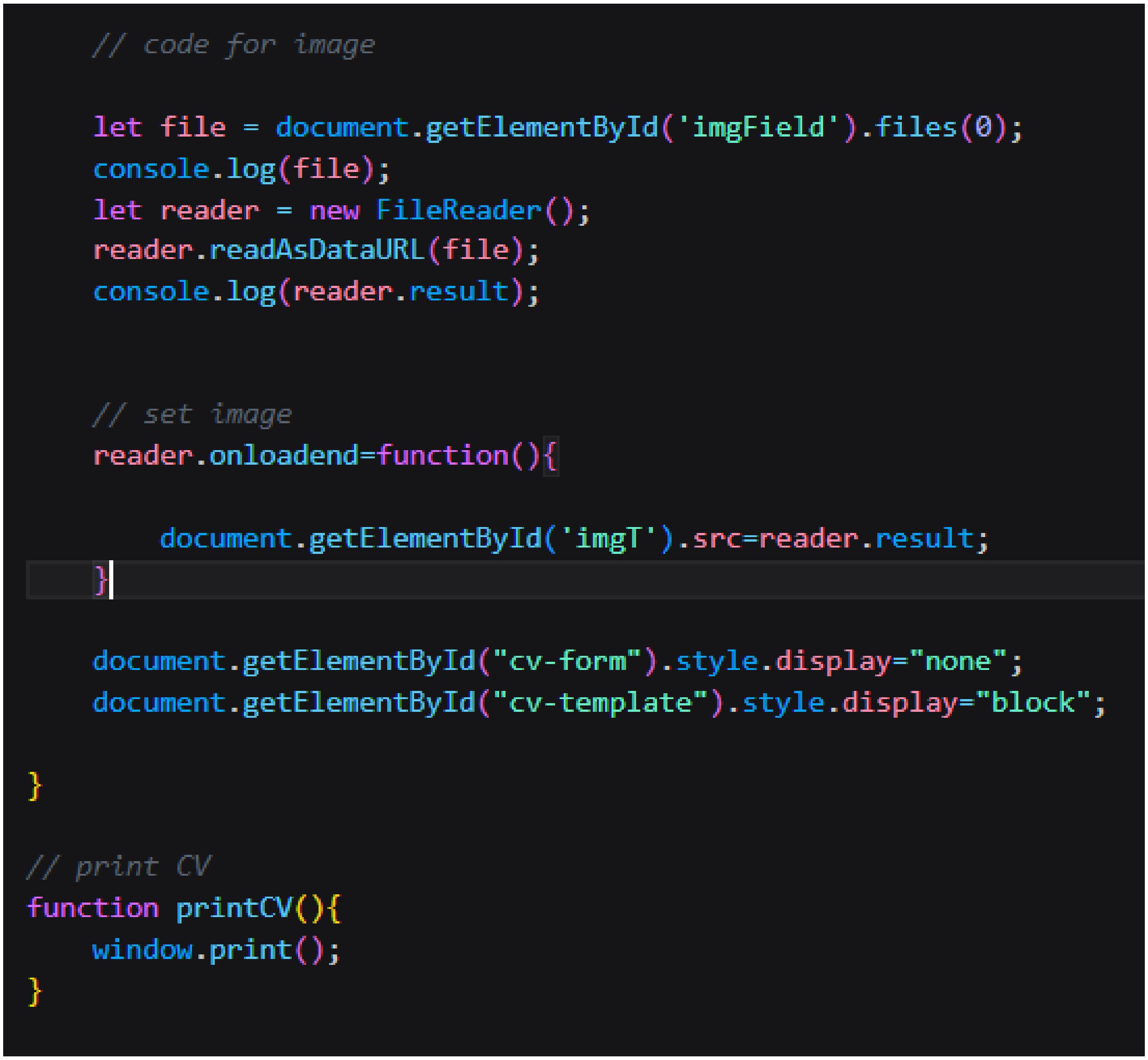
C. Java Script code:



```
// All JS function in here
function addNewWEField(){
    // console.log("Adding new field");

    let newNode=document.createElement("textarea");
    newNode.classList.add("form-control");
    newNode.classList.add("weField");
    newNode.classList.add("mt-2");

    newNode.setAttribute("rows",3);
    newNode.setAttribute("placeholder","Enter Here");

    let weOb = document.getElementById("we");
    let weAddButtonOb = document.getElementById("weAddButton");

    weOb.insertBefore(newNode, weAddButtonOb);

}

function addNewAQField(){

    let newNode=document.createElement("textarea");
    newNode.classList.add("form-control");
    newNode.classList.add("eqField");
    newNode.classList.add("mt-2");

    newNode.setAttribute("rows",3);
    newNode.setAttribute("placeholder","Enter Here");

    let aqOb = document.getElementById("aq");
    let aqAddButtonOb = document.getElementById("aqAddButton");

    aqOb.insertBefore(newNode, aqAddButtonOb);

}
```



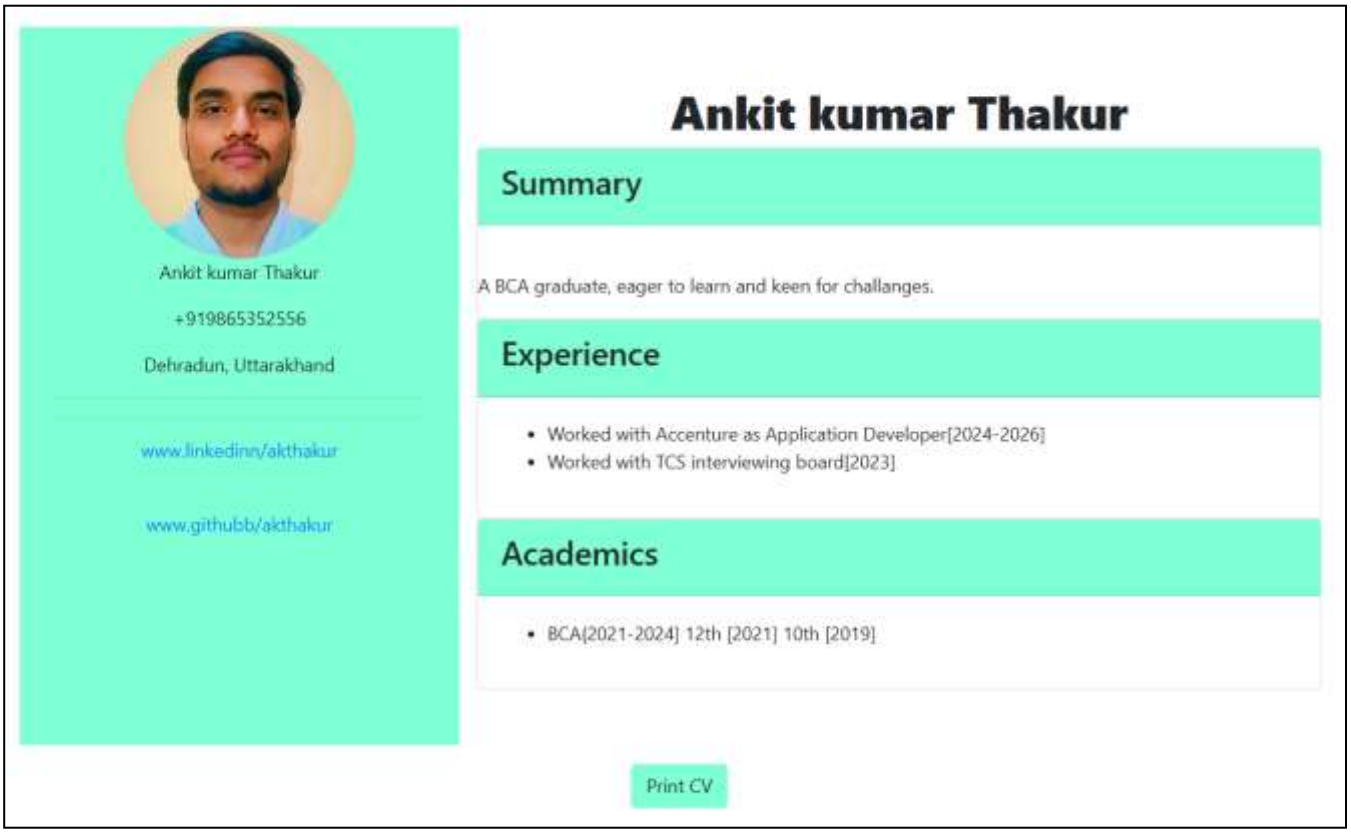
```
// generate CV

function generateCV(){
    // console.log("going");
    let nameField=document.getElementById("nameField").value;

    let nameT1 = document.getElementById("nameT1");

    nameT1.innerHTML=nameField;


    document.getElementById("nameT2").innerHTML=nameField;

    document.getElementById("contactT").innerHTML=document.getElementById("contactField").value;

    document.getElementById("addressT").innerHTML=document.getElementById("addressField").value;
    document.getElementById("ldT").innerHTML=document.getElementById("ldField").value;
    document.getElementById("ghT").innerHTML=document.getElementById("ghField").value;

    document.getElementById("summaryT").innerHTML=document.getElementById("sumField").value;

    // Work experienxe
    let wes=document.getElementsByClassName('weField');
    let str="";
    for (let e of wes){
        str=str+`<li> ${e.value} </li>`;
    }

    document.getElementById("weT").innerHTML = str;

    // academic qualifiaction
    let aqs=document.getElementsByClassName('eqField');
    let str1="";
    for(let e of aqs){
        str1+=`<li> ${e.value} </li`;
    }

    document.getElementById("aqT").innerHTML=str1;
```



```
    // code for image

    let file = document.getElementById('imgField').files(0);
    console.log(file);
    let reader = new FileReader();
    reader.readAsDataURL(file);
    console.log(reader.result);


    // set image
    reader.onloadend=function(){

        document.getElementById('imgT').src=reader.result;
    }

    document.getElementById("cv-form").style.display="none";
    document.getElementById("cv-template").style.display="block";

}

// print CV
function printCV(){
    window.print();
}
```

Figure 5.3 JavaScript code for functioning

Chapter 6 Result and Output:



Ankit kumar Thakur

+919865352556

Dehradun, Uttarakhand

www.linkedinn/akthakur

www.githubb/akthakur

Ankit kumar Thakur

Summary

A BCA graduate, eager to learn and keen for challanges.

Experience

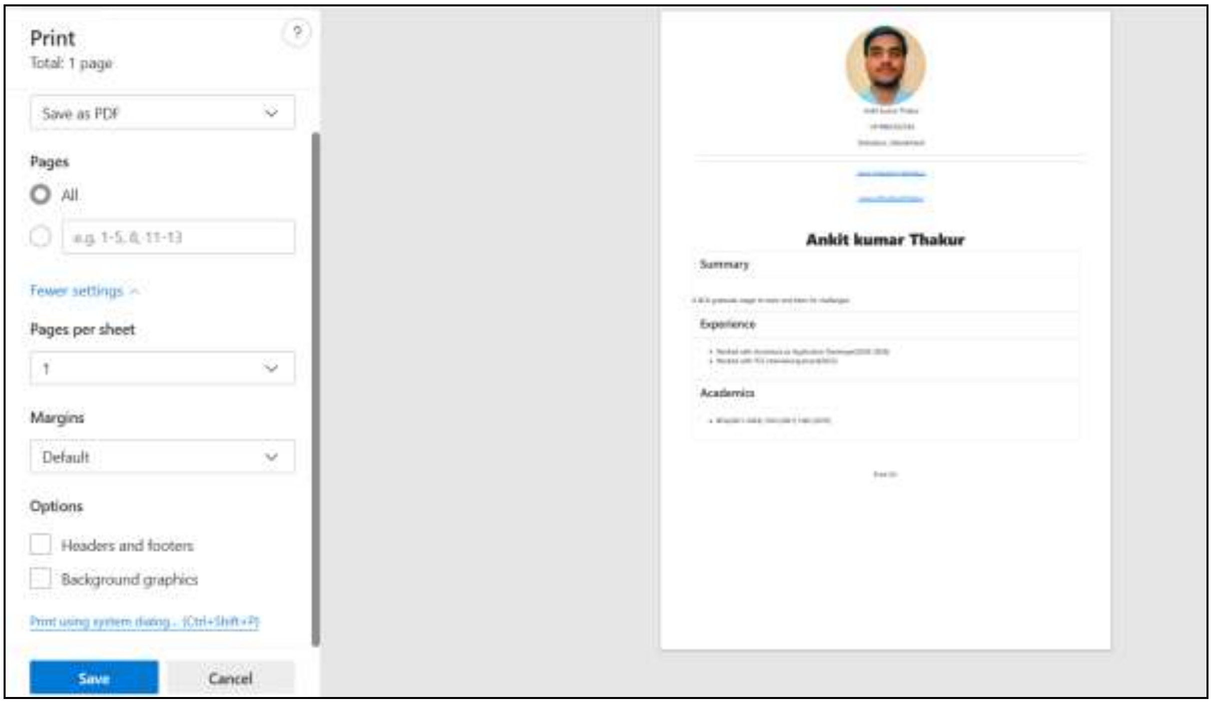
- Worked with Accenture as Application Developer[2024-2026]
- Worked with TCS interviewing board[2023]

Academics

- BCA[2021-2024] 12th [2021] 10th [2019]

Figure 6 Test case 1

After pressing PrintCV button the following CV is printed/ saved in pdf format



Print

Total: 1 page

Save as PDF

Pages

● All

○ e.g. 1-5, 8, 11-13

Fewer settings

Pages per sheet: 1

Margins: Default

Options

□ Headers and footers

□ Background graphics

Print using system dialog... (Ctrl+Shift+P)

Save   Cancel

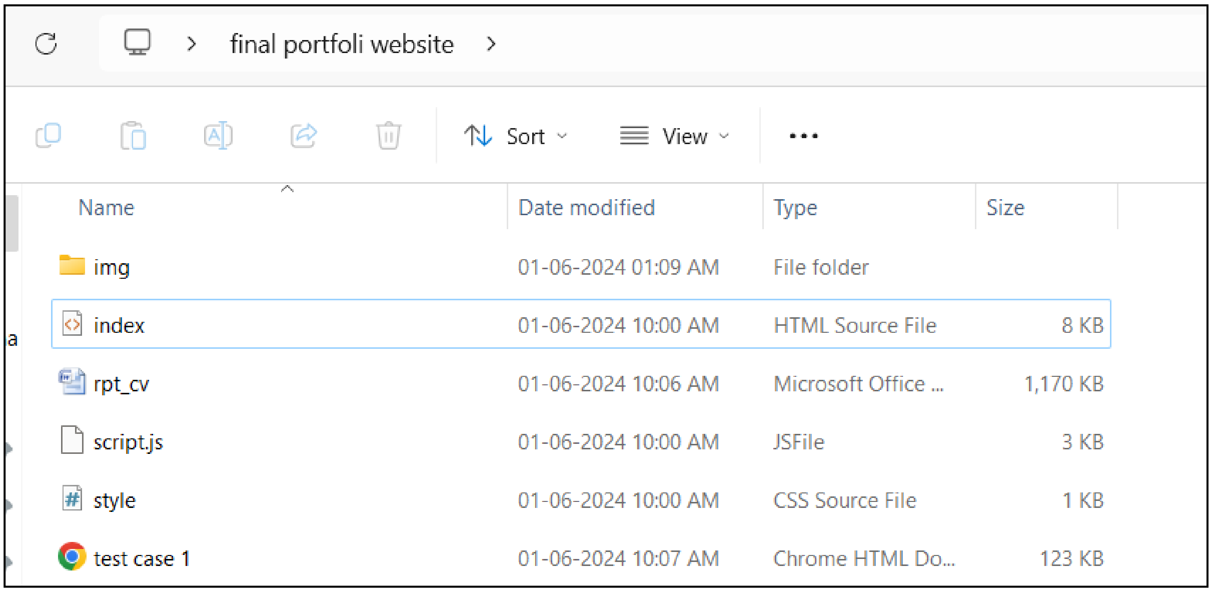
Ankit kumar Thakur

Summary

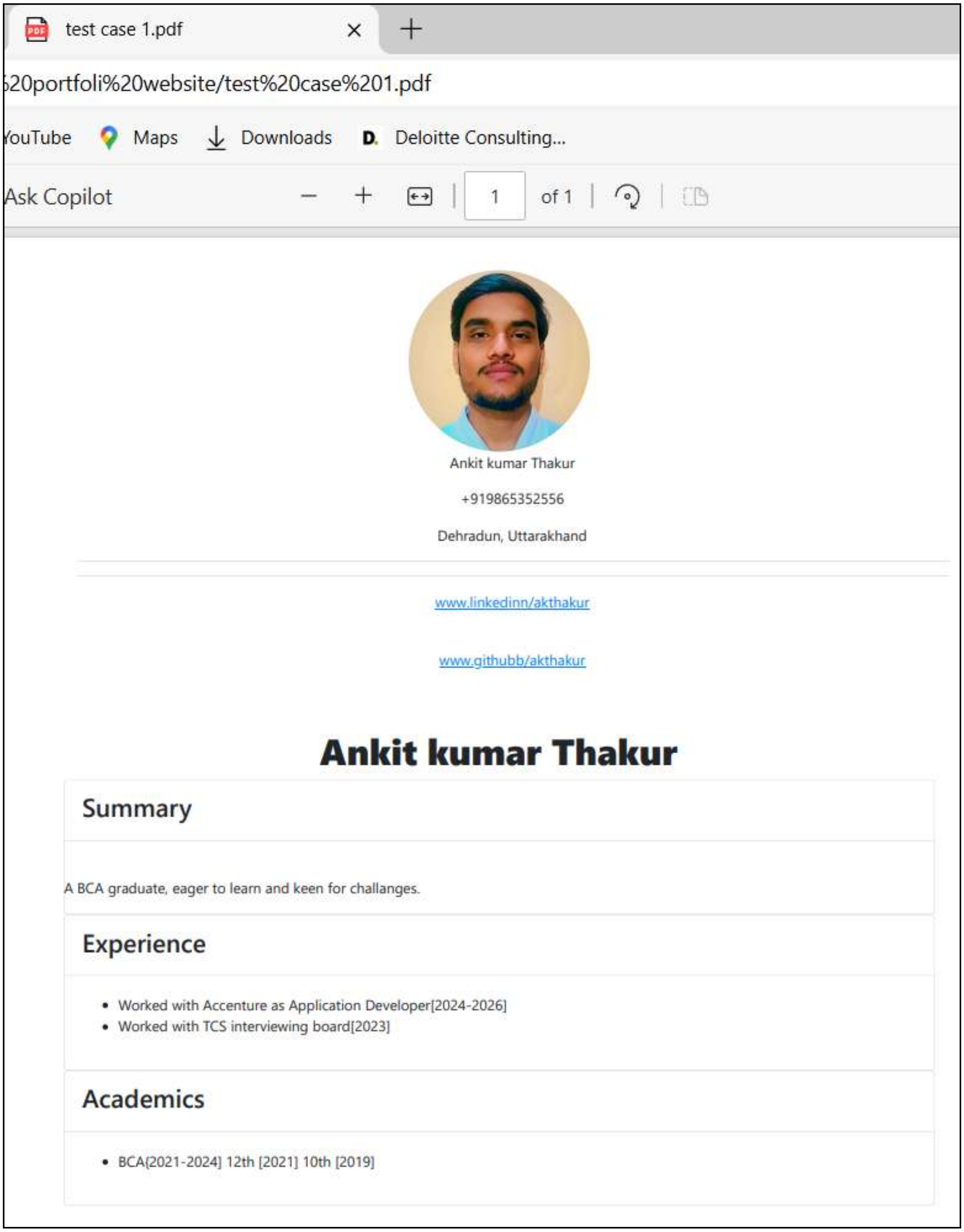
Experience

Academics

CV is saved in same folder with source code in pdf format.



Name	Date modified	Type	Size
img	01-06-2024 01:09 AM	File folder	
index	01-06-2024 10:00 AM	HTML Source File	8 KB
rpt_cv	01-06-2024 10:06 AM	Microsoft Office ...	1,170 KB
script.js	01-06-2024 10:00 AM	JSFile	3 KB
style	01-06-2024 10:00 AM	CSS Source File	1 KB
test case 1	01-06-2024 10:07 AM	Chrome HTML Do...	123 KB



Ankit kumar Thakur

+919865352556

Dehradun, Uttarakhand

www.linkedinn/akthakur

www.githubb/akthakur

Ankit kumar Thakur

Summary

A BCA graduate, eager to learn and keen for challanges.

Experience

- Worked with Accenture as Application Developer[2024-2026]
- Worked with TCS interviewing board[2023]

Academics

- BCA[2021-2024] 12th [2021] 10th [2019]

Chapter 7 Conclusion:

In conclusion, a web-based CV generator offers a powerful and user-friendly solution for individuals seeking to create professional and polished CVs with ease. By leveraging modern web technologies, such as React.js for the front-end and Node.js for the back-end, this application streamlines the process of CV creation, allowing users to focus on content rather than formatting. The inclusion of features like multiple template selections, real-time previews, and secure user authentication enhances the user experience, making the tool accessible and efficient for a wide audience.

The benefits of using a web-based CV generator are manifold. Users can quickly input their information, customize their CVs to suit specific job applications, and export them in various formats. This efficiency not only saves time but also ensures that the CVs produced are of high quality and tailored to meet the requirements of potential employers. Furthermore, the application's ability to store and manage multiple CV versions adds a layer of convenience for users applying to various roles.

Overall, the web-based CV generator stands as a testament to the capabilities of web technologies in simplifying complex tasks. By providing an intuitive interface, robust features, and secure handling of personal data, this application addresses the needs of modern job seekers. As technology continues to evolve, further enhancements can be integrated, such as advanced analytics and AI-driven content suggestions, to continually improve the user experience and effectiveness of CV generation.

Chapter 8 Future Work Plan

The report of project work allocated by the supervisor is as follows:

Sl. No.	Work Description	Duration
1.	Advancing the CV generator to work for multiple Templates and Provide advance suggestions for Building both CV and Resume	3 weeks

Chapter 9 Weekly Tasks

The report of project work allocated by the supervisor is as follows:

Week No.	Date: From- To	Work allocated	Work Completed (Yes/No)	Remarks	Guide Signature
1.	09/02/24 16/02/24	Information collection of previously Models	Yes		
2.	19/02/24 06/03/24	Understanding andlearning about HTML, CSS, JS along with Bootstrap other modules	Yes		
3.	18/03/24 02/04/24	Building and Defining the builder to work accordingly	Yes		
4.	06/05/24 17/05/24	Final Linking and testing of Generator	Yes		

Bibliography & References

[1] : Bye Bye, Bullets: The Stack Overflow Developer Story is the New Technical Resume - Stack Overflow

[2]: How to Make a Resume That Stands Out: Examples & Tips (zety.com)

[3]:https://www.bing.com/search?q=traditional+cv+vs+online+cv&cvid=9cc0fc1f7d15409e905e8a6320a098b9&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIGCAEQABhAMgYIAhAAGEAyBggDEAAYQDIGCAQQABhAMgYIBRAAGEDSAQkxMjE0OGowajmoAgSwAgE&FORM=ANAB01&PC=DCTS

[4]: 10 Resume Writing Tips To Help You Land a Position | Indeed.com

[5] : Academic CV templates for applications or to send to professors? (MS CS) : r/gradadmissions (reddit.com)

[6] : Mohd Khairuddin, Nurul Syahmina, et al. "Re: Gen-web-based resume generator with content recommender for it job field." (2022):

[7]: Khaled, Tahsin Bin. "Web-Application on CV-Building." (2023).

[8]: Mulla, Aslaan, et al. "Resume Builder Application with Automated Job Prediction." NeuroQuantology 21.1 (2023): 128.

[9]: Shivhare, Kratika, Sonam Shakya, and Aashi Singh Bhadouria. "ResumeCraft: A Machine Learning-powered Web Platform for Resume Building".

[10]: https://en.wikipedia.org/wiki/HTML

[11] : https://en.wikipedia.org/wiki/CSS.

[12]: "ECMAScript 2020 Language Specification". Archived from the original on 8 May 2020. Retrieved 8 May 2020.

[13]: https://en.wikipedia.org/wiki/JavaScript#cite_note-tc39-11

[14] "Search · stars:>100000". GitHub. Retrieved December 4, 2022.