"PORTFOLIO WEBSITE"

*Project report (CA3) submitted in fulfilment of the requirements for the Degree of*

# BACHELOR OF TECHNOLOGY

**in**

**COMPUTER SCIENCE AND ENGINEERING**

By

**ARPAN GUPTA**

## ( 12210528 )

Section- K22WD, Roll No.- 18

SUBJECT

**INT219: FRONT END WEB DEVELOPER**



**School of Computer Science and Engineering**

Lovely Professional University Phagwara, Punjab (India) November, April 2024

# TABLE OF CONTENTS

1. [INTRODUCTION 03](#_bookmark0)
   1. [PROBLEM STATEMENT 03](#_bookmark1)
   2. [WHY USE THIS TECHNOLOGY? 04](#_bookmark2)
   3. [MVC ARCHITECTURE 05](#_bookmark3)
2. [TECHNOLOGIES USED 07](#_bookmark4)
   1. [HTML CSS JAVASCRIPT 07](#_TOC_250002)
   2. [JQUERY 08](#_TOC_250001)
   3. [BOOTSTRAP 08](#_TOC_250000)
   4. HBS 08
3. [KEY FEATURES OF THE APP 09](#_bookmark5)
4. [MODULES 10](#_bookmark6)
5. [WEBSITE SNAPSHOTS 11](#_bookmark7)
6. [GITHUB LINK 16](#_bookmark8)
7. [LIST OF REFERENCES 16](#_bookmark9)

# INTRODUCTION

## PROBLEM STATEMENT

I

1. **User Engagement**: Capturing and retaining the attention of visitors is crucial for a successful portfolio website. Achieving this requires a seamless and intuitive user experience that guides visitors through the content, showcasing relevant information in an engaging manner. Failure to do so may result in high bounce rates and lost opportunities to impress potential clients or employers.
2. \*\***Visual Appeal**\*\*: A visually appealing design is paramount for a portfolio website, as it reflects the creator’s aesthetic sensibilities and attention to detail. The website should not only showcase the work but also present it in an visually attractive and cohesive manner. Achieving a balance between aesthetics and functionality can be challenging, particularly when dealing with diverse media types, such as images, videos, and interactive elements.
3. **Technical Implementation**: Developing a portfolio website often involves integrating various technologies and frameworks to achieve the desired functionality and performance. This may include server-side rendering, dynamic content generation, responsiveness across different devices, and optimization for search engines. Choosing the right tech stack and ensuring seamless integration can be a daunting task, especially for those without extensive web development experience.
4. **Content Management**: As professionals continually update their work and achievements, managing and maintaining the website’s content can become cumbersome. Implementing an efficient and user-friendly content management system (CMS) or a custom solution is crucial for streamlining the process of adding, updating, and organizing content while ensuring consistency across the website.
5. **Showcasing Diverse Skills**: For professionals with a wide range of skills and expertise, effectively showcasing their work in a cohesive and organized manner can be challenging. Finding the right balance between depth and breadth of content, while maintaining a clear and intuitive structure, is essential to ensure visitors can easily navigate and appreciate the full scope of the individual’s abilities.
6. **Search Engine Optimization (SEO)**: In today’s digital landscape, optimizing the website for search engines is crucial for ensuring visibility and discoverability. Implementing SEO best practices, such as optimizing meta tags, structuring content for crawlers, and leveraging schema markup, can be a complex undertaking, particularly for those unfamiliar with the nuances of search engine algorithms.

Addressing these challenges requires a strategic approach that combines creative design, technical proficiency, and a deep understanding of user experience principles. By overcoming these obstacles, professionals can create a portfolio website that effectively showcases their skills, captivates their audience, and ultimately contributes to their professional success.

## WHY USE THIS TECHNOLOGIES ?

## HTML, CSS, and JavaScript

## HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript form the core technologies used for developing the front-end of the portfolio website. These technologies work together to create a visually appealing and interactive user experience.

## HTML provides the structural foundation, defining the semantic elements and content hierarchy of the web pages. It acts as the blueprint for the website, organizing the content into sections, headings, paragraphs, and other meaningful elements.

## CSS is responsible for controlling the visual presentation and styling of the HTML elements. It allows developers to define the layout, colors, typography, and overall aesthetic appeal of the website. CSS ensures a consistent look and feel across different pages and devices, enhancing the user experience.

## JavaScript, on the other hand, introduces interactivity and dynamic behavior to the website. It enables features such as animations, form validations, pop-ups, carousels, and other interactive components. JavaScript also facilitates communication with servers, enabling the retrieval and manipulation of data in real-time, enhancing the website's functionality and responsiveness.

## jQuery

## jQuery is a popular JavaScript library that simplifies client-side scripting by abstracting common tasks such as DOM (Document Object Model) manipulation, event handling, and Ajax requests. Its concise syntax and extensive plugin ecosystem enhance productivity and cross-browser compatibility, making it easier to develop complex web applications.

## By leveraging jQuery, developers can write less code and achieve more functionality, reducing development time and increasing efficiency. jQuery's powerful selectors and manipulation methods allow for easy traversal and modification of the DOM, while its event handling capabilities streamline the management of user interactions.

## Bootstrap

## Bootstrap is a widely-used front-end framework that streamlines responsive web design with its grid system, pre-styled components, and utilities. By leveraging Bootstrap's CSS and JavaScript components, the portfolio website achieves consistency, responsiveness, and mobile-friendliness across devices.

## Bootstrap's grid system provides a flexible and responsive layout structure, ensuring that the website's content adapts seamlessly to different screen sizes and resolutions. Its extensive library of pre-built components, such as navigation bars, buttons, forms, and modals, accelerates development and promotes a consistent user interface across the entire website.

## Handlebars.js (hbs

## Handlebars.js (hbs) is a popular templating engine used for server-side rendering of dynamic content. It allows developers to embed dynamic data and logic directly into HTML templates using a concise syntax. With Handlebars.js, the portfolio website can generate HTML pages on the server-side, utilizing data from various sources, such as databases or APIs. This approach enhances flexibility and maintainability by separating the presentation logic from the application logic.

## Handlebars.js templates consist of HTML markup with placeholders for dynamic data, known as expressions. These expressions are wrapped in double curly braces, like `{{expression}}`. The expressions can be as simple as displaying a variable or as complex as executing conditional statements or loops. Handlebars.js also supports partials, which are reusable template snippets that can be included in other templates, promoting code reuse and modular design.

## By employing Handlebars.js in the portfolio website, developers can leverage server-side rendering to generate dynamic HTML pages based on data from various sources, ensuring a seamless and personalized user experience. This approach offers benefits such as improved performance, better search engine optimization (SEO), separation of concerns, reusability, and flexibility to adapt to changing requirements, ensuring a compelling and engaging experience for visitors.

## MVC ARCHITECTURE

The portfolio website follows the Model-View-Controller (MVC) architectural pattern to ensure a well-structured and maintainable codebase. This pattern divides the application into three interconnected components:

**Model**: The Model component represents the data and database interactions in the portfolio website. It is responsible for managing data related to projects, skills, and other relevant information, ensuring that data is stored, retrieved, and manipulated correctly.

**View**: The View component deals with the presentation and user interface of the portfolio website. It ensures that the content, such as project details, skill descriptions, and other information, is displayed appropriately to visitors, providing an interactive and visually appealing experience.

**Controller**: The Controller component acts as an intermediary between the Model and View in the portfolio website. It processes user interactions, such as navigating to different sections or filtering projects, and updates the Model accordingly. It then selects the appropriate View to display the updated information to the visitor.

By adhering to the MVC pattern, the portfolio website achieves a clear separation of concerns, making it easier to maintain and extend the codebase. The Model handles data management, the View handles presentation logic, and the Controller manages the flow of data between the two. This modular approach promotes code reusability, testability, and scalability, ultimately enhancing the overall development and maintenance process of the portfolio website.

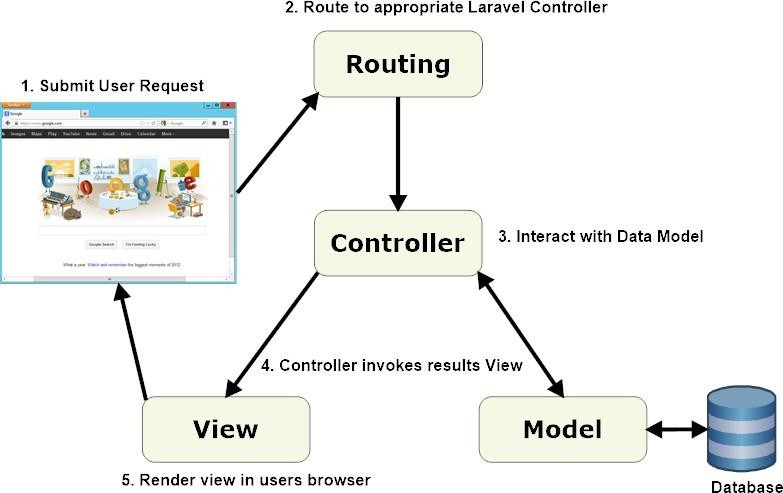


Figure 1: MVC architecture of the application

The Model-View-Controller (MVC) architecture, widely utilized in software development and particularly prevalent in web applications, divides application responsibilities into three interconnected elements: Model, View, and Controller. Each component plays a vital role in managing and delivering the application's functionality effectively.

## Characteristics of Model:

* + - Contains the application's business logic, including validation rules and data transformation.
    - Provides an interface for interacting with the database.
    - Isolated from the user interface and presentation concerns.

## Characteristics of View

* + - Receives user input, such as form submissions or user interactions.
    - Often includes HTML templates, CSS, and other elements that control the visual representation of the application.
    - Isolated from the application's business logic and data processing.

## Characteristics of Controller:

* + - Interacts with the Model to retrieve or update data based on user input.
    - Communicates with the View to ensure the appropriate presentation of data to the user.
    - Orchestrates the flow of data and user interactions, ensuring that the Model and View remain isolated from each other.

## Advantages of MVC:

* + - **Separation of Concerns:** Promotes a clear separation of responsibilities, enabling developers to work on individual components independently, enhancing modularity, maintainability, and extensibility of code.
    - **Code Reusability:** Components within MVC can be reused in other parts of the application or in different applications, reducing redundancy and development time.
    - **Scalability:** The modular structure of MVC makes it easier to scale and enhance specific components without impacting the entire application.
    - **Parallel Development:** Developers can work in parallel on different components of the architecture, streamlining the development process.

# TECHNOLOGIES USED

## HTML CSS JAVASCRIPT

HTML, CSS, and JavaScript collectively shape the backbone of web development, offering distinct functionalities that converge to craft immersive and interactive digital experiences:

## HTML (Hypertext Markup Language):

* + - **Structural Foundation**: Serves as the structural backbone for web content, defining elements and organizing information for seamless display.
    - **Semantic Markup**: Encodes content meaningfully, aiding accessibility and search engine optimization efforts.
    - **Linking and Embedding**: Facilitates the integration of multimedia elements, hyperlinks, and resources within web pages.

## CSS (Cascading Style Sheets):

* + - **Visual Styling:** Controls the aesthetic presentation of HTML elements, encompassing layout, colors, typography, and design elements.
    - **Responsive Design:** Enables the creation of adaptive layouts that adjust gracefully across various devices and screen sizes.
    - **Animations and Transitions:** Enhances user engagement through animated effects, transitions, and visual enhancements.

## JavaScript:

* + - **Dynamic Interactivity**: Empowers the creation of interactive web elements, enabling real-time updates, user input processing, and dynamic content rendering.
    - **Event Handling**: Facilitates the responsiveness of web pages to user actions, triggering events like clicks, scrolls, and keyboard inputs.
    - **Client-Side Functionality**: Executes logic within users' browsers, reducing server load and enhancing user experience through dynamic behavior and personalized features.

## JQUERY

jQuery is a popular JavaScript library known for simplifying client-side scripting and DOM manipulation tasks, offering numerous advantages:

* + - **Ease of Use**: jQuery simplifies complex tasks, such as event handling, animation, and AJAX calls, with concise syntax.
    - **Cross-Browser Compatibility**: Ensures consistent behavior across different browsers, streamlining development and enhancing user experience.
    - **Extensive Plugin Ecosystem**: Access to a wide range of plugins for additional functionality, accelerating development and expanding capabilities.

## BOOTSTRAP

Bootstrap is a front-end framework that streamlines web development by providing pre- designed components and styles, offering various benefits:

* + - **Responsive Design**: Ensures websites are mobile-friendly and adapt seamlessly to different screen sizes.
    - **Consistent UI**: Provides a unified and visually appealing user interface across web pages, enhancing user experience.
    - **Customization**: Allows for easy customization through themes, templates, and CSS variables, facilitating design flexibility.

## EJS(Embedded JavaScript):

Handlebars.js (hbs) is a popular templating engine used for server-side rendering of dynamic content. It allows developers to embed dynamic data and logic directly into HTML templates using a concise syntax. With Handlebars.js, the portfolio website can generate HTML pages on the server-side, utilizing data from various sources, such as databases or APIs. This approach enhances flexibility and maintainability by separating the presentation logic from the application logic.

Handlebars.js templates consist of HTML markup with placeholders for dynamic data, known as expressions. These expressions are wrapped in double curly braces, like {{expression}}. The expressions can be as simple as displaying a variable or as complex as executing conditional statements or loops. Handlebars.js also supports partials, which are reusable template snippets that can be included in other templates, promoting code reuse and modular design.

By employing Handlebars.js in the portfolio website, developers can leverage server-side rendering to generate dynamic HTML pages based on data from various sources, ensuring a seamless and personalized user experience. This approach offers benefits such as improved performance, better search engine optimization (SEO), separation of concerns, reusability, and flexibility to adapt to changing requirements, ensuring a compelling and engaging experience for visitors.

# KEY FEATURES OF THE APP

The Byte Tabs website offers a host of key features aimed at enhancing user experience, fostering collaboration, and ensuring efficient communication between freelancers and clients. Here are the core features:

Here are the core features:

* + - **Responsive Design**: Byte Tabs is designed with a responsive layout, ensuring optimal viewing and interaction across a wide range of devices, including desktops, tablets, and smartphones.
    - **User-Friendly Interface**: The website boasts a user-friendly interface that makes it easy for freelancers to navigate and apply for jobs, and for clients to find the right freelancers for their projects.
    - **Freelancer Job Application**: Freelancers can apply for jobs directly through the website, providing them with a convenient platform to showcase their skills and expertise to potential clients.
    - **Client Contact Form**: Clients can easily contact Byte Tabs using a contact form, enabling them to discuss their project requirements and get the necessary assistance.
    - **Collaborative Environment**: My Portfolio fosters a collaborative environment where freelancers and clients can interact, discuss project details, and ensure smooth project execution.
    - **Efficient Communication:** facilitates efficient communication between freelancers and clients, enabling them to discuss project details, deadlines, and other important information.
    - **Feedback and Ratings:** Freelancers and clients can provide feedback and ratings based on their experience, helping to build a trusted community of users.

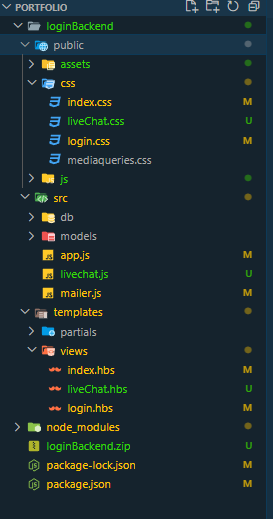
# MODULES

This section will outline key modules of our freelance website ByteTabs, accompanied by code snippets and explanations.

## Project Structure:

* + - **Inside the ByteTabs folder:**

1. **views Folder:** Contains all HBS files.
2. **public Folder:** Contains three subfolders:
   * images: Stores all images used in website.
   * javascripts: Contains all JavaScript files.
   * stylesheets: Contains all CSS files.
3. **app.js File:** Contains code that demonstrates a basic setup for a Node.js and Express application with MongoDB for database storage, including routes for different pages and form handling.



# WEBSITE SNAPSHOTS

A screenshot of a web page

Description automatically generated

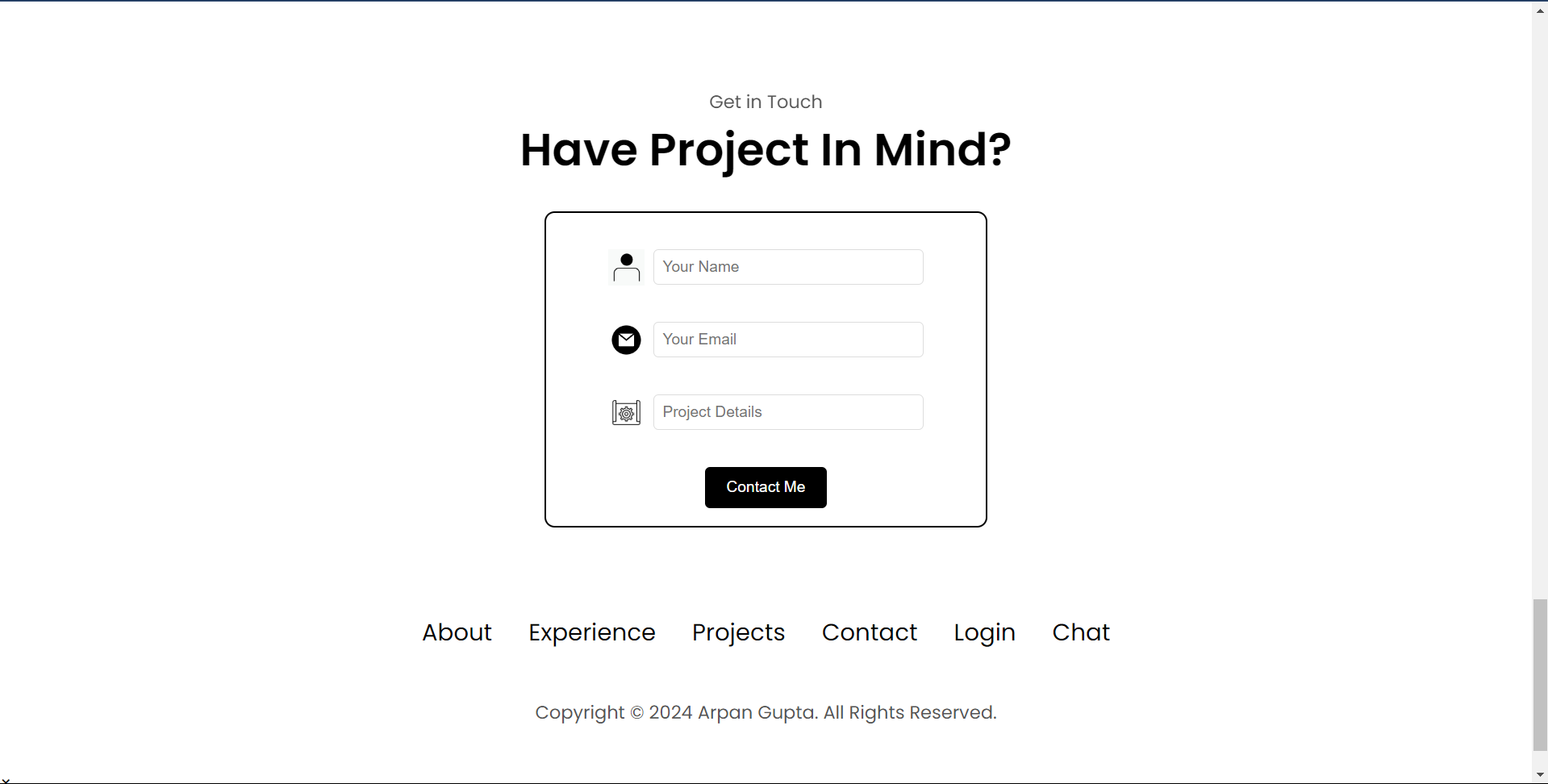
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a group of people

Description automatically generated



14

# GITHUB LINK

* + - The link to the GitHub repository is:

<https://github.com/arpanhub/portfolio>

# LIST OF REFERENCES

* + - HTML Documentation: <https://developer.mozilla.org/en-US/docs/Web/HTML>
    - CSS Documentation: <https://developer.mozilla.org/en-US/docs/Web/CSS>
    - JavaScript Documentation: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
    - Bootstrap 5 Documentation: [https://getbootstrap.com/docs/5.0/getting-](https://getbootstrap.com/docs/5.0/getting-started/introduction/) [started/introduction/](https://getbootstrap.com/docs/5.0/getting-started/introduction/)
    - JQuery Documentaation: <https://api.jquery.com/>