PROJECT REPORT

On

STUDENT PORTFOLIO WEBSITE

SUBMITTED BY

Aparna Kumari	**19105125059**
Nilesh kishore	**19105125044**
Anubhav Kumar Jha	**19105125039**
Shivam Singh	**19105125060**
Priyadarshani	**19105125017**

Under the Supervision of

Prof. Rohit Kumar

Department of Computer Science & Engineering
Rashtrakavi Ramdhari Singh Dinkar College of Engineering
Begusarai, Bihar - 851134

Session: 2019-23

DECLARATION

We declare that the work presented in this project titled "**Student Portfolio Website**", submitted to the Computer Science & Engineering Department of **Rashtrakavi Ramdhari Singh Dinkar College of Engineering, Begusarai (Bihar)** for the award of the **Bachelor of Technology** degree in **Computer Science & Engineering** from **Bihar Engineering University, Patna** is our original work based on our own experience and observations to the best of our knowledge and understanding. we have not plagiarized or submitted the same work for the award of any other degree. It is realiable document and is of a bonafide nature.

Name of all the students of the group with signature and registration number

Name	Reg no.	Signature
Aparna Kumari	19105125059	
Nilesh Kishore	19105125044	
Anubhav Kumar Jha	19105125039	
Shivam Singh	19105125060	
Priyadarshani	19105125017	

Place: Begusarai

CERTIFICATE

This is to certify that the Project entitled "**Student Portfolio Website**" in partial fulfilment of the requirement for the award of the **Bachelor of Technology** degree in **Computer Science & Engineering**, submitted to **Rashtrakavi Ramdhari Singh Dinkar College of Engineering Begusarai, Bihar** is an authentic record of research work carried out by "**Aparna Kumari, Nilesh Kishore, Anubhav Kumar Jha, Shivam Singh & Priyadarshani**" under my supervision.

Prof. ROHIT KUMAR
Department of CSE
R.R.S.D.C.E, Begusarai

Prof. MD SHAHID AHMAD
H.O.D Department of CSE
R.R.S.D.C.E, Begusarai

External examiner

ACKNOWLEDGEMENT

We would like to acknowledge and express our special feeling and deep gratitude towards **Computer Science & Engineering Department RRSDCE, Begusarai** and all team member who gave me the golden opportunity to take participate in such type of Project.

We would like to thank our guide, **Prof. Rohit Kumar Sir** for his guidance in the whole project and special thanks for support and helped in completing this Project submission and providing us such a great learning experience.

We take profound sense of pride to convey my gratefulness towards **Bihar Engineering University** (Dept. of Science & Technology, Govt. of Bihar) and my Institution **Rashtrakavi Ramdhari Singh Dinkar College of Engineering, Begusarai** for providing me with this great opportunity to such an enlightment minor Project in 7th Semester.

ABSTRACT

Our project is a portfolio website built using HTML, CSS, JavaScript, Sass, PHP, and XAMPP server. It provides students with a platform to create personalized portfolios by filling a form with 32 different attributes. The submitted information is securely stored in a database. Users can then retrieve their portfolio by typing their registration number, accessing a personalized portfolio webpage. This project aims to enhance students' online presence and showcase their achievements in a professional and customizable manner.

In this project, we utilize a combination of front-end technologies such as HTML, CSS, JavaScript, and Sass to design an attractive and responsive user interface. The form enables students to input comprehensive details about themselves, including personal information, educational background, skills, projects, work experience, certifications, and more.

The back-end functionality is implemented using PHP and the XAMPP server, ensuring the secure storage and retrieval of user information. When a student submits the form, their data is saved in a database, providing a reliable and organized storage solution. Subsequently, the registration number acts as a unique identifier to retrieve the stored information and generate a personalized portfolio webpage.

The portfolio webpage showcases the student's information in a visually appealing and structured format. It allows them to present their achievements, skills, and experiences in a personalized way, making a lasting impression on potential employers, collaborators, or educational institutions.

The project not only offers a platform for students to create their portfolios but also enhances their online visibility and professional branding. By providing a personalized portfolio webpage, it enables students to conveniently share their portfolio with others and stand out in competitive environments.

Future possibilities for this project include adding additional customization options to allow students to further personalize their portfolio webpage, integrating social media features for enhanced networking, and incorporating interactive elements to engage visitors.

CONTENTS

1. INTRODUCTION

Welcome to our innovative project, a comprehensive portfolio website designed to empower students and showcase their talents and achievements in a personalized and professional manner. Our project brings together the power of HTML, CSS, JavaScript, Sass, PHP, and the XAMPP server to create a dynamic platform where students can build their own customized portfolios.

In today's competitive world, having a standout portfolio is crucial for students seeking opportunities in academia, employment, or entrepreneurship. Our project addresses this need by providing a user-friendly interface where students can fill a detailed form with 32 different attributes. These attributes encompass various aspects of their personal information, educational background, skills, projects, work experience, certifications, and more.

With the submission of the form, the student's data is securely stored in a robust database, ensuring the privacy and integrity of their information. The registration number assigned to each student serves as a unique identifier, allowing for easy retrieval of their portfolio data. By simply entering their registration number, students gain access to a personalized portfolio webpage tailored to showcase their accomplishments.

The portfolio webpage presents the student's information in a visually captivating and organized manner. With the power of HTML, CSS, and JavaScript, we create a responsive and attractive layout that highlights the student's skills, achievements, and experiences. The result is a professionally designed portfolio that leaves a lasting impression on potential employers, collaborators, or educational institutions.

Our project not only offers a platform for students to build their portfolios but also enhances their online presence and personal branding. By providing a dedicated webpage, students can easily share their portfolios with others, including potential employers or admissions committees, effectively conveying their capabilities and aspirations.

Looking ahead, our project opens up exciting possibilities for further development. We envision expanding customization options, enabling students to personalize their portfolio webpages even more. Integration with social media features can facilitate networking and promote wider reach. Interactive elements can be incorporated to engage visitors and provide a richer user experience.

2. SOFTWARE USED

2.1 Visual Studio Code

Version Used 1.78

Visual Studio Code, is a free source-code editor developed by Microsoft. It is widely used by developers for writing and editing code across different programming languages. VS Code is known for its lightweight and customizable nature, making it a popular choice among developers

Fig 1. Logo of VS Code

2.2 XAMPP Control Panel

Version Used 3.3.0

XAMPP Control Panel is a graphical user interface (GUI) application that comes with XAMPP, which is a software package containing Apache HTTP Server, MySQL database, PHP, and Perl. XAMPP is commonly used for web development and creating a local server environment on your computer.

The XAMPP Control Panel provides an interface to manage and control the various components of XAMPP. It allows you to start, stop, and configure the Apache web server, MySQL database server, and other related services.

Fig 2. Logo of XAMMP Control Panel

3. SYSTEM REQUIREMENT

The system requirements provided outline the minimum specifications necessary to run XAMPP on different operating systems. Here's an explanation of each requirement.

3.1 Windows requirements:

The system requirements provided outline the minimum specifications necessary to run XAMPP on different operating systems. Here's an explanation of each requirement.

Operating system: XAMPP is compatible with Windows 8 or later versions.

Processor: An Intel Pentium 4 or later processor is required.

Memory: The minimum required memory is 2 GB, but it is recommended to have 4 GB or more for optimal performance.

Screen resolution: A screen resolution of 1280x1024 or larger is recommended.

Application window size: The application window should have a size of 1024x680 or larger to ensure proper display of the XAMPP Control Panel.

Internet connection: An internet connection is required to download and update XAMPP components as needed.

3.2 Mac requirements:

Operating system: XAMPP is compatible with macOS High Sierra 10.13 or later versions.

Processor: Any Intel processor is suitable for running XAMPP on a Mac.

Memory: The required memory specifications for Mac are not explicitly mentioned, but it is recommended to have a sufficient amount of RAM for smooth operation.

Screen resolution: The screen resolution requirements are not specified, but it is recommended to have a screen resolution that provides a comfortable viewing experience.

Application window size: Similar to the screen resolution, specific requirements for the application window size are not provided.

Internet connection: An internet connection is necessary for downloading and

updating XAMPP components.

3.3 Linux requirements:

Operating system: XAMPP is compatible with 64-bit Linux distributions, including Ubuntu 14.04 or later, Debian 8 or later, openSUSE 13.3 or later, and Fedora Linux 24 or later.

Processor: Similar to Windows, an Intel Pentium 4 or later processor is required for Linux systems.

Memory: The memory requirements for Linux are not specified, but it is recommended to have sufficient RAM for smooth operation.

Screen resolution: Specific screen resolution requirements are not mentioned, but it is advisable to have a suitable resolution for comfortable usage.

Application window size: The application window should have a size that allows for the proper display of the XAMPP Control Panel.

Internet connection: An internet connection is necessary for downloading and updating XAMPP components on Linux.

	Windows requirements	**Mac requirements**	**Linux requirements**
Operating system	Windows 8 or later	macOS High Sierra 10.13 or later	64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Memory	2 GB minimum, 4 GB recommended		
Screen resolution	1280x1024 or larger		
Application window size	1024x680 or larger		
Internet connection	Required		

Table 1. System Requirement table

4. Langauge Used

4.1 HTML

HTML, which stands for Hyper Text Markup Language, is the standard markup language used for creating web pages and applications. It provides a set of tags or elements that define the structure and content of a web page. HTML is the backbone of every web page and is interpreted by web browsers to render the visual representation of the page.

4.2 CSS

CSS, which stands for Cascading Style Sheets, is a styling language used to describe the presentation and layout of HTML documents. It provides a set of rules and properties that control the visual appearance of web pages. CSS separates the content of a web page from its presentation, allowing developers to easily apply consistent styles across multiple pages.

4.3 SASS

Sass (Syntactically Awesome Style Sheets) is a preprocessor scripting language that extends the capabilities of CSS. It introduces features and enhancements to CSS, making it more powerful and efficient to write and maintain stylesheets. Sass files are processed and compiled into regular CSS files that can be used in web development.

4.4 PHP

PHP (Hypertext Preprocessor) is a widely used server-side scripting language designed for web development. It is embedded within HTML code and executed on the server before being sent to the client's web browser. PHP enables dynamic web page generation, interaction with databases, handling form data, file manipulation, and much more.

4.5 JAVASCRIPT

JavaScript is a high-level, interpreted programming language primarily used for front-end web development. It allows you to add interactivity and dynamic behavior to web pages. JavaScript is supported by all modern web browsers and can also be used on the server side (e.g., with Node.js) for back-end development

5. SYSTEM DESIGN

5.1 TABLE DESIGN

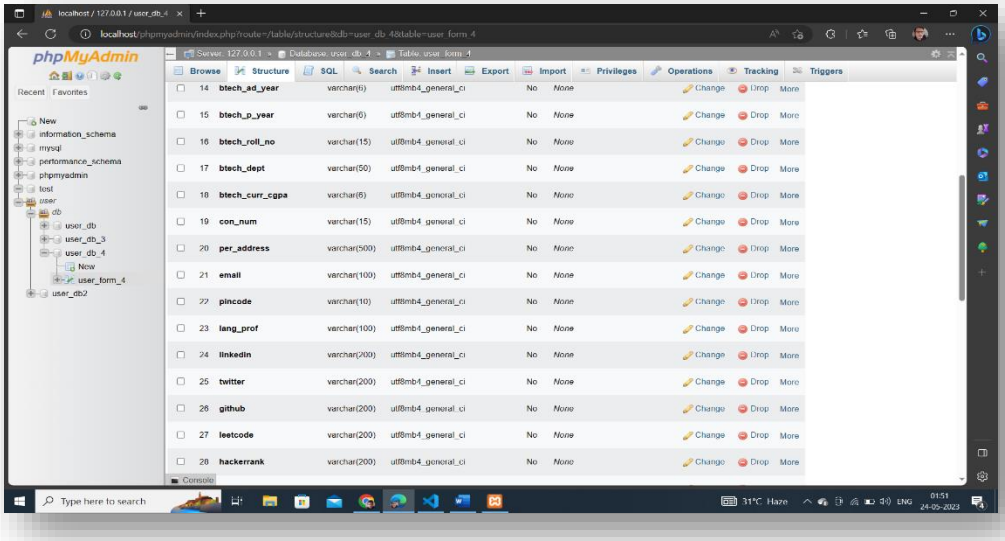

Fig 3 Structure of Database Table

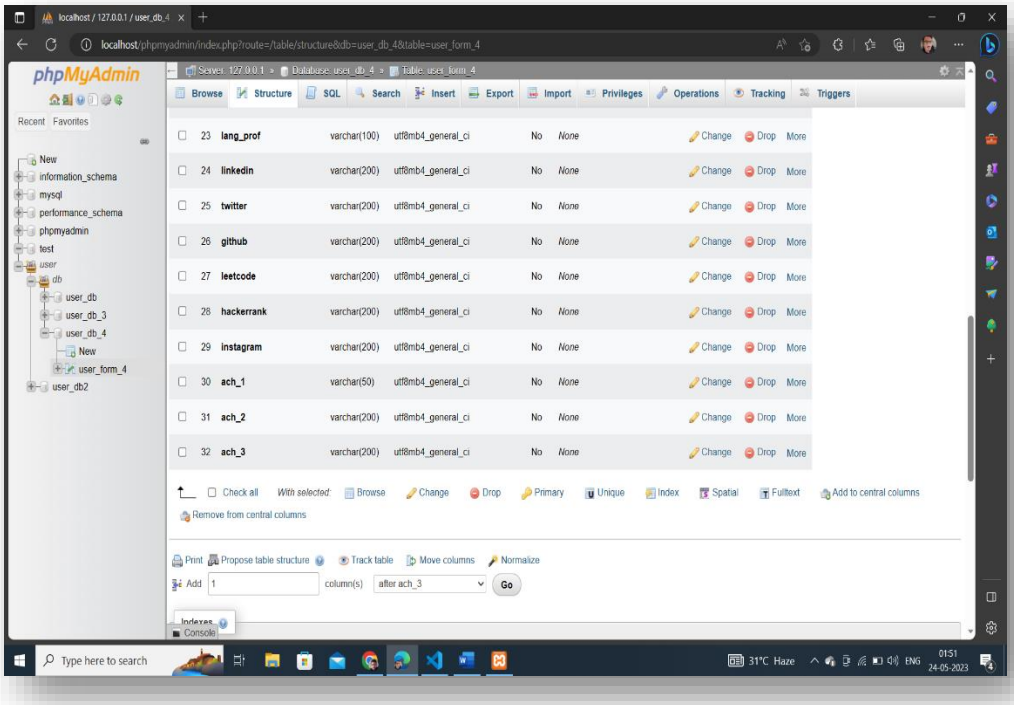

Fig 4 Structure of Database Table

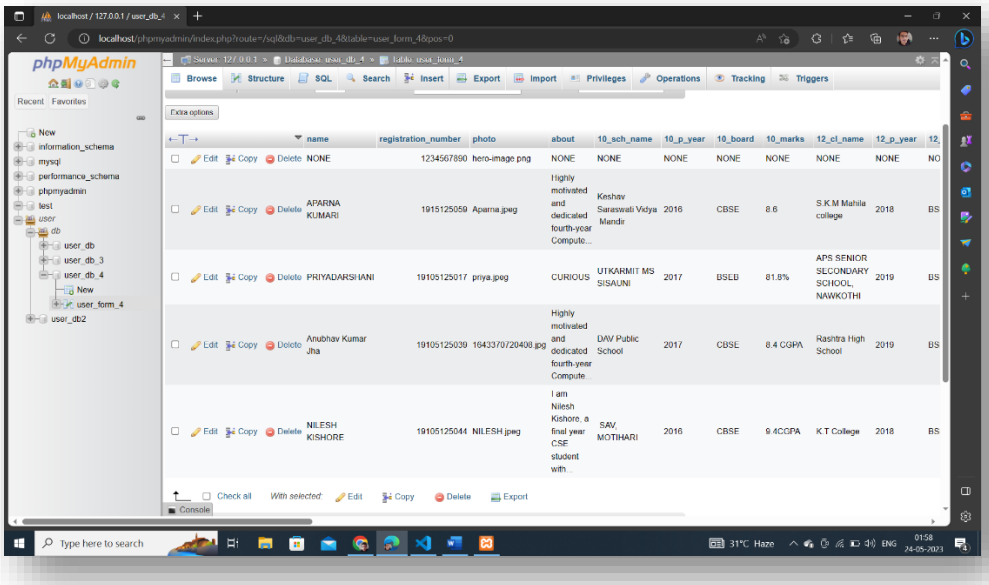

Fig 5 Database Table 1

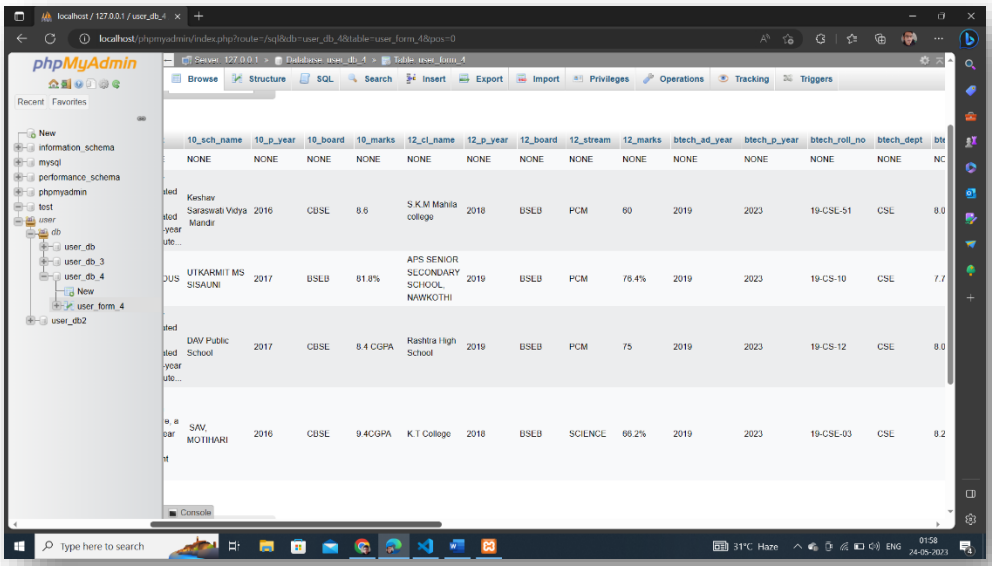

Fig 6 Database Table 2

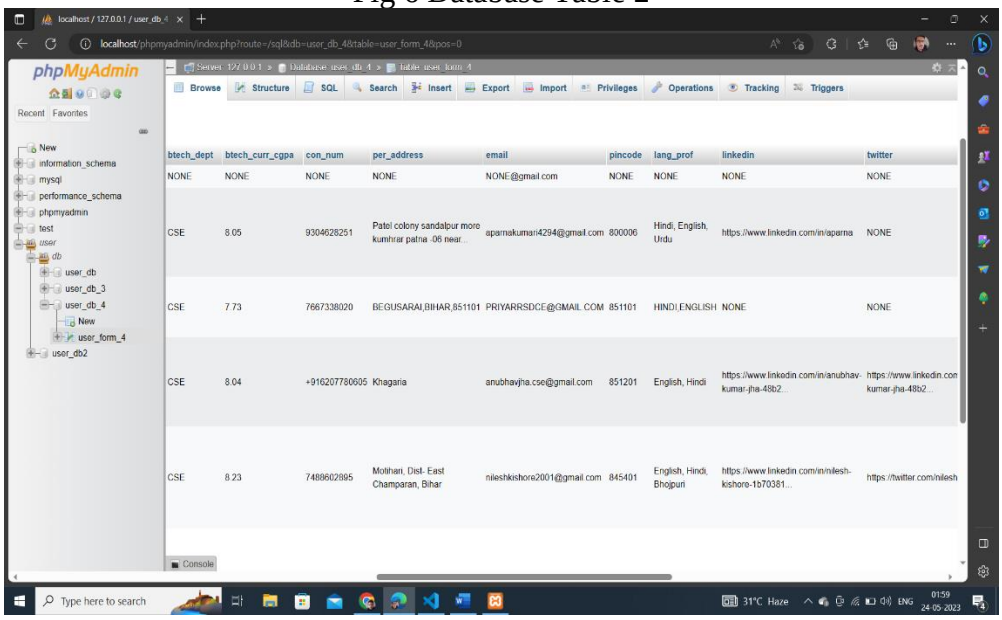

Fig 7 Database Table 3

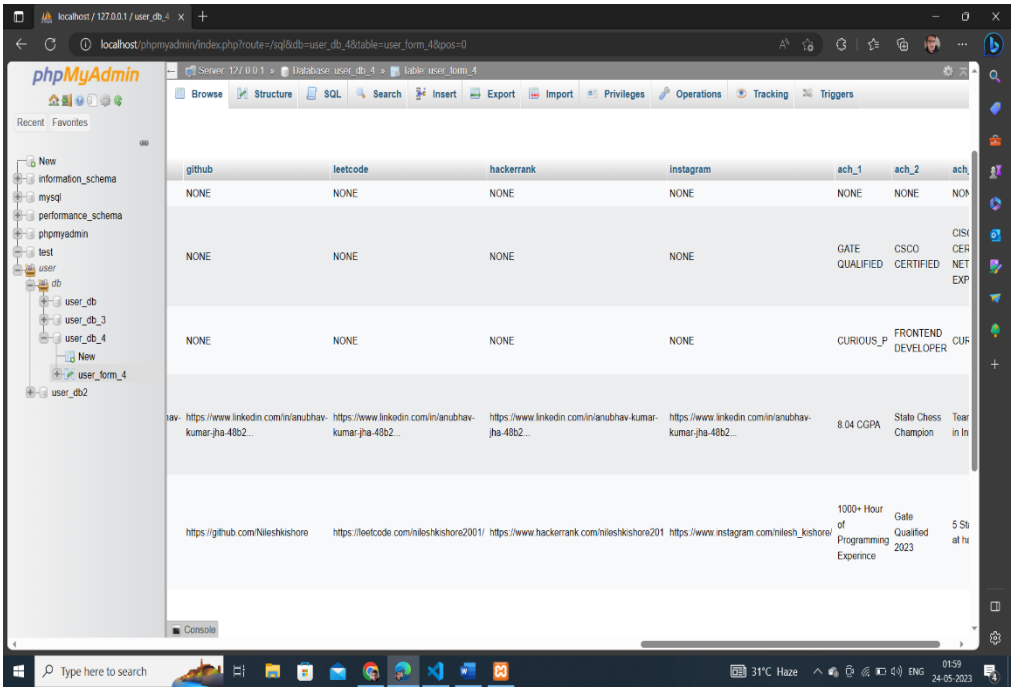

Fig 8 Database Table 4

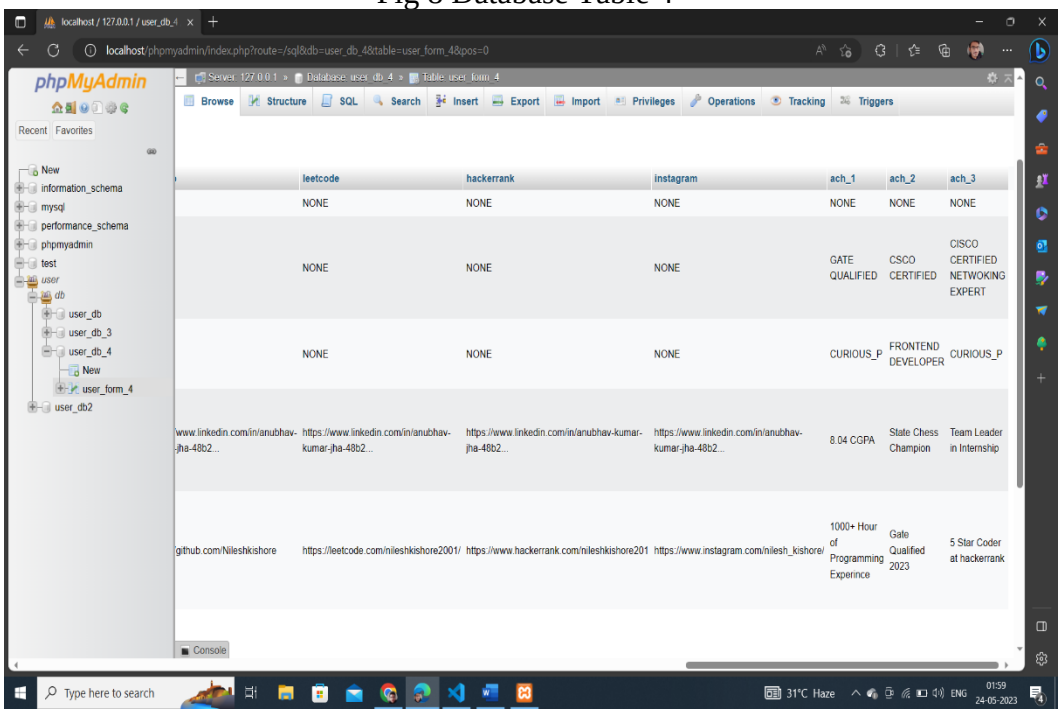

Fig 9 Database Table 5

5.2 Index Page

The main functionality of the page is a search feature. When the user submits a search query by clicking the "Search" button, the PHP code retrieves the entered registration number, sanitizes it, and performs a database query using the entered

number. If a matching record is found in the user_form_4 table, the corresponding data is stored in a session variable called 'result', and the user is redirected to the 'profile.php' page. If no match is found, the 'result' session variable is set to null, and the user is redirected back to the index page.

The HTML section of the page includes the necessary markup for the header, navigation menus (both desktop and mobile versions), a hero section with some introductory content, and testimonials from students. It also features a mentor section with an image and description. The page ends with a footer section containing quick links, contact information, and a subscription form.

Overall, the index page provides an overview of the website's purpose and features, allows users to search for specific information using a registration number, and showcases testimonials and a mentor section to highlight the achievements and credibility of the institution.

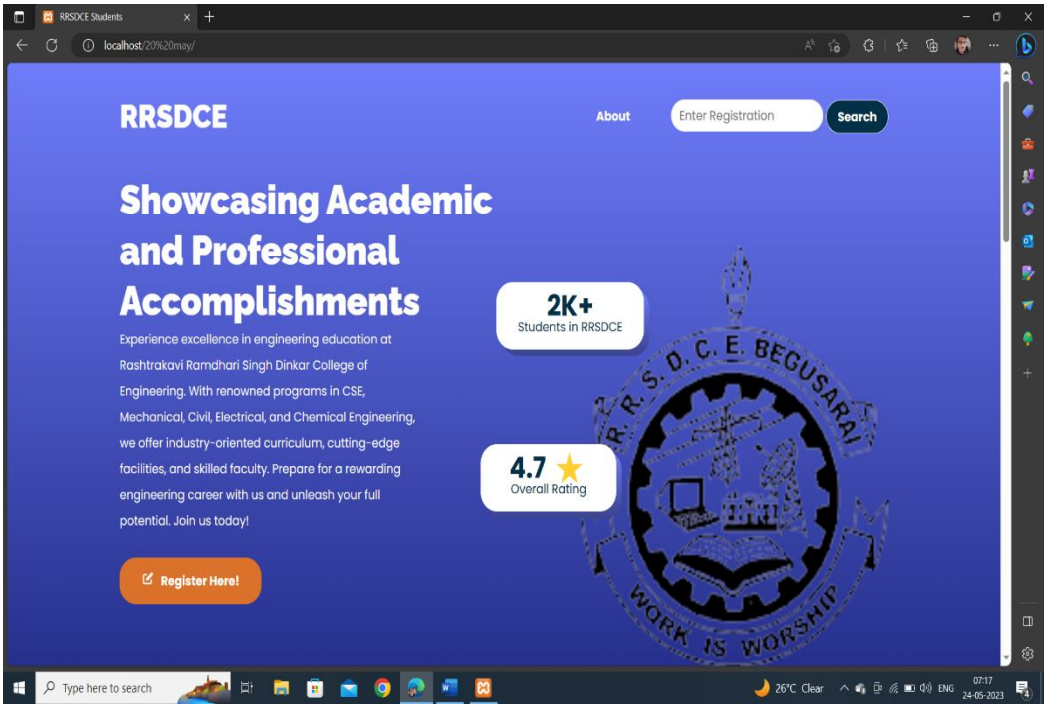

Fig 10 Index web page 1

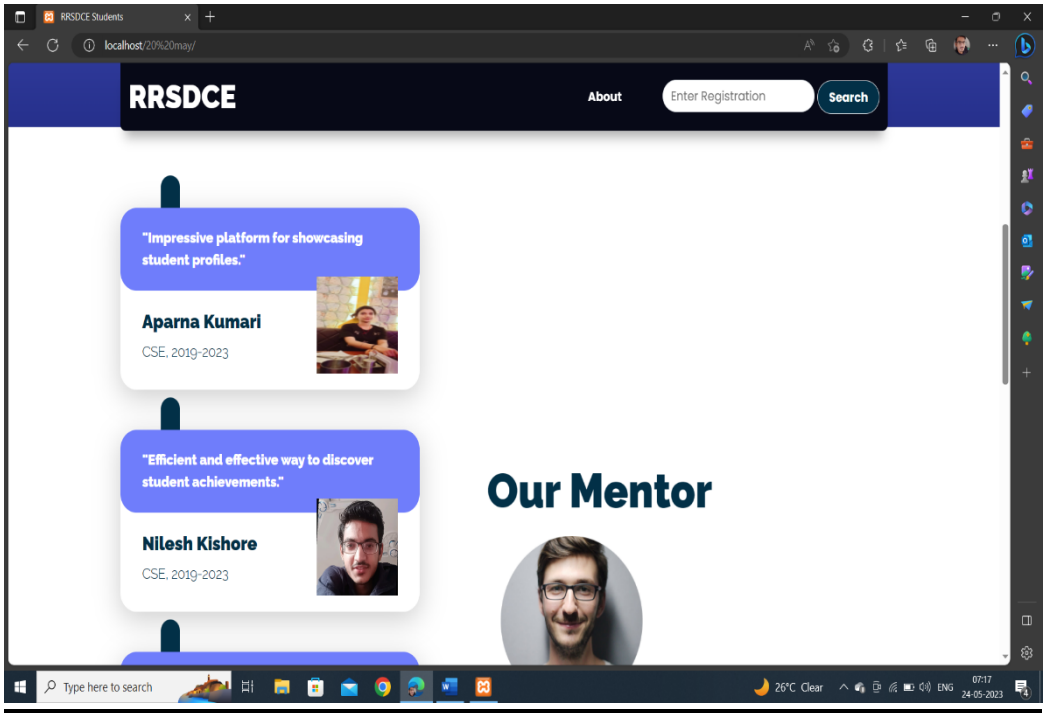

Fig 11 Index web page 2

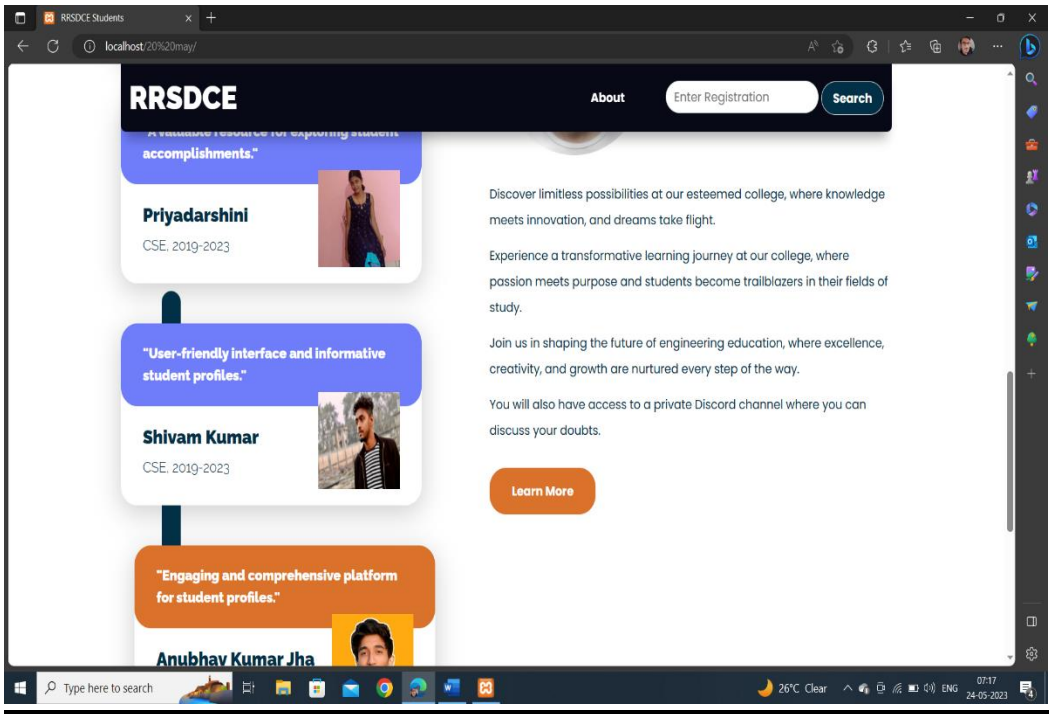

Fig 12 Index web page 3

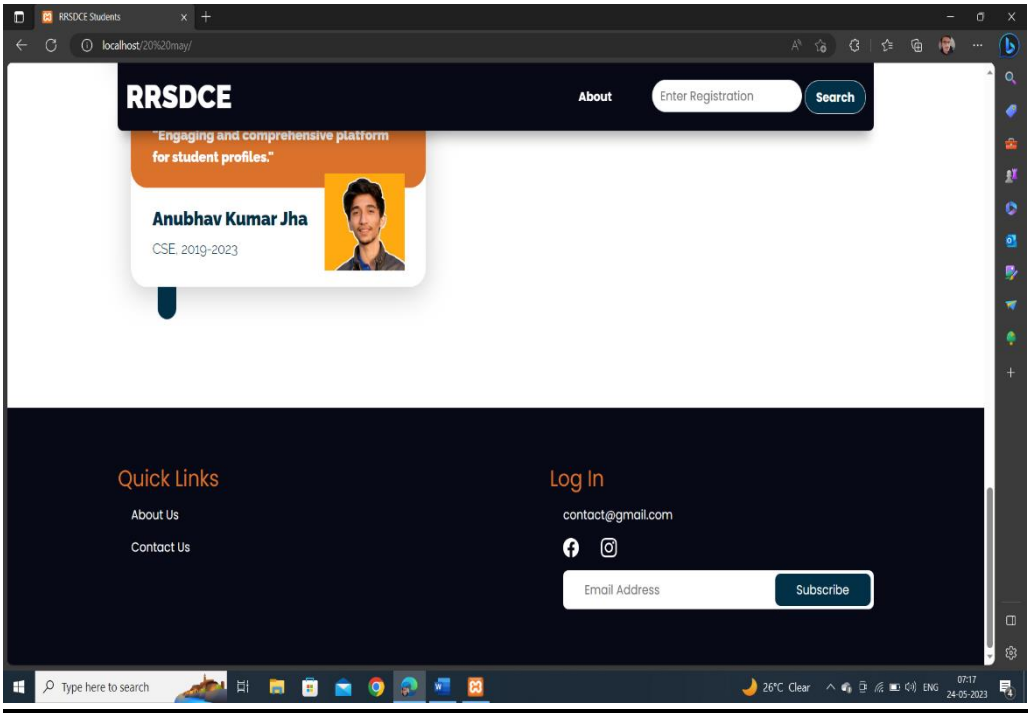

Fig 13 Index web page 4

5.3 Register Page

The purpose of this user information form is to simplify the collection and organization of personal, educational, and contact details of individuals. By providing a structured and user-friendly interface, we aim to enhance the efficiency and accuracy of data collection while ensuring a smooth user experience.

Features and Functionality:

Personal Information:

Name: Capture the full name of the user.

Registration Number: Obtain a unique identifier for the user.

Photo: Allow users to upload their photo for identification purposes.
About: Provide a space for users to share a brief description or bio.
Education Details:

10th and 12th Grade Information: Gather data related to school name, passing year, board, stream, and marks obtained.
B-Tech Details: Collect information such as admission year, passing year, roll number, department, and current CGPA.

Contact Information and Social Media:

Contact Number: Record the user's phone number.
Permanent Address: Enable users to provide their permanent address.
Email: Capture the user's email address.
Pincode: Gather the pincode or postal code of the user's location.
Language Proficiency: Allow users to specify their proficiency in different languages.
Social Media Profiles: Provide fields for LinkedIn, Twitter, GitHub, LeetCode, HackerRank, and Instagram profiles.

Achievements:
Capture up to three achievements or accomplishments of the user.
Form Submission and Result:

Upon submission, the form validates and stores the information securely.
An image upload feature enables users to upload their photo, which is stored and associated with their profile.
After successful registration, users are redirected to their profile page, where they can view their submitted information.

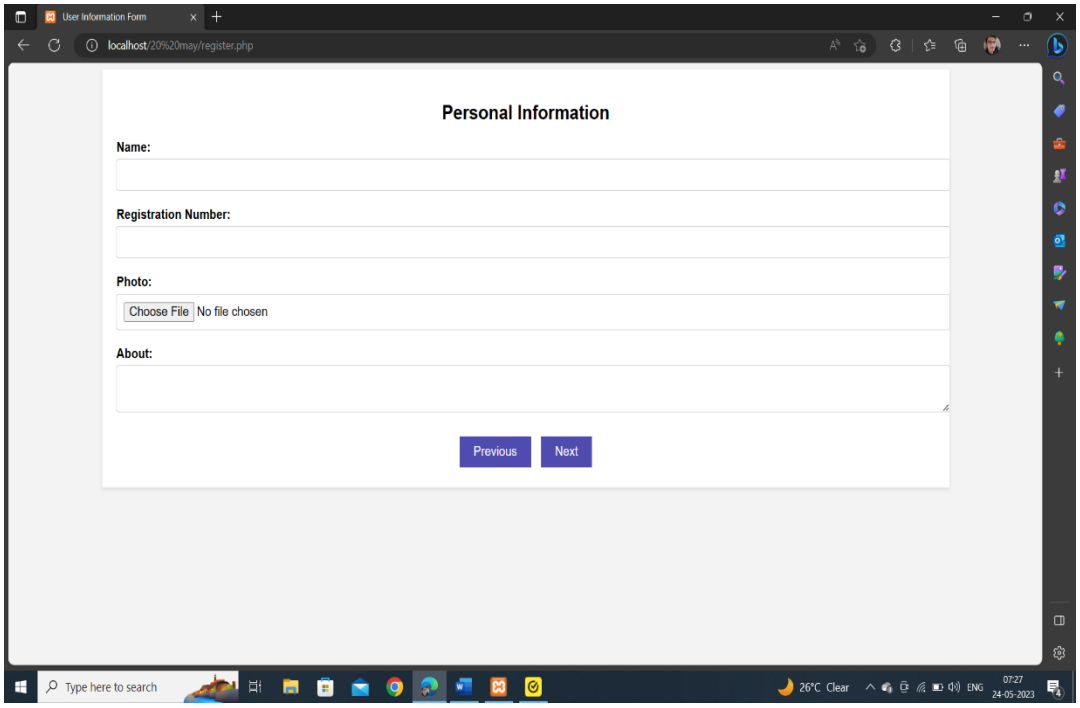

Fig 14 Register web page 1

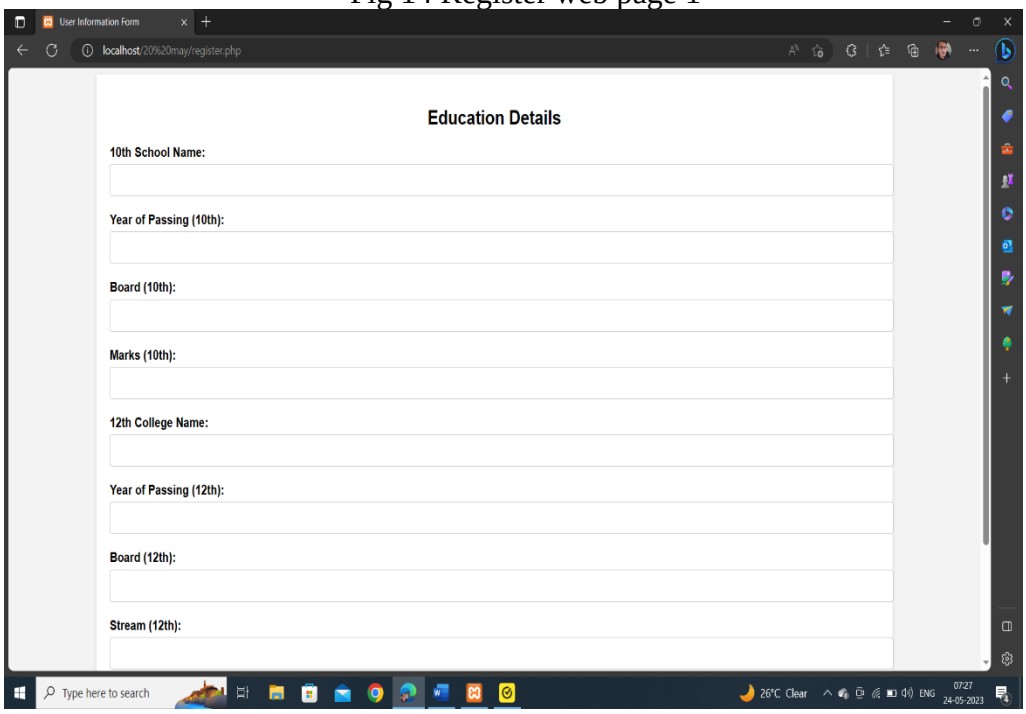

Fig 15 Register web page 2

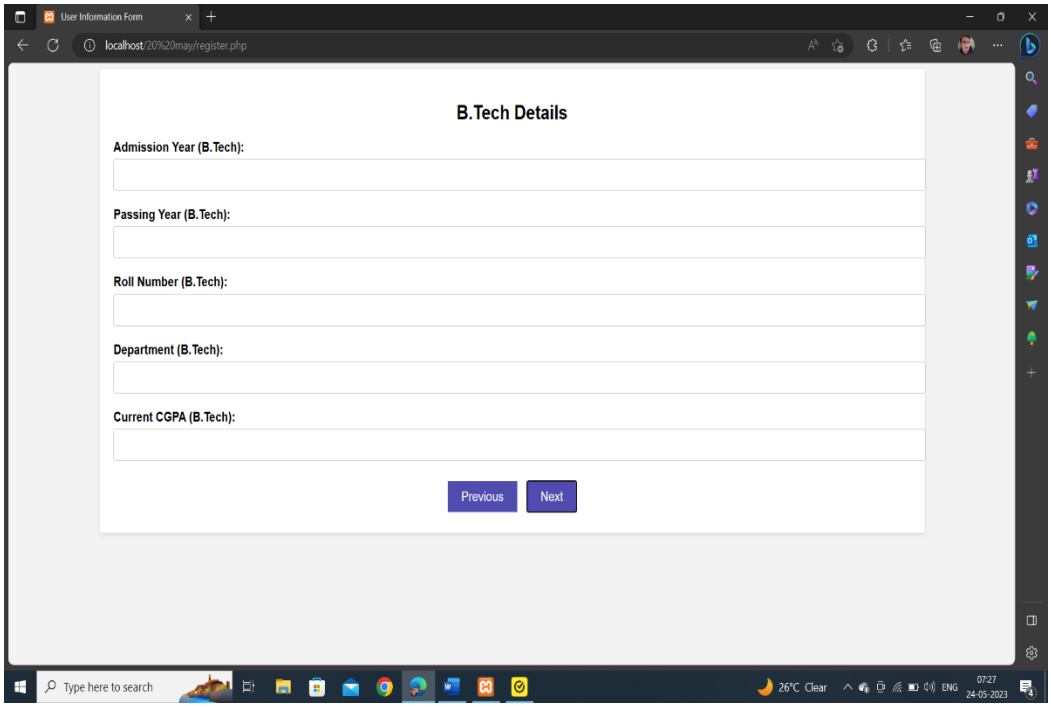

Fig 16 Register web page 3

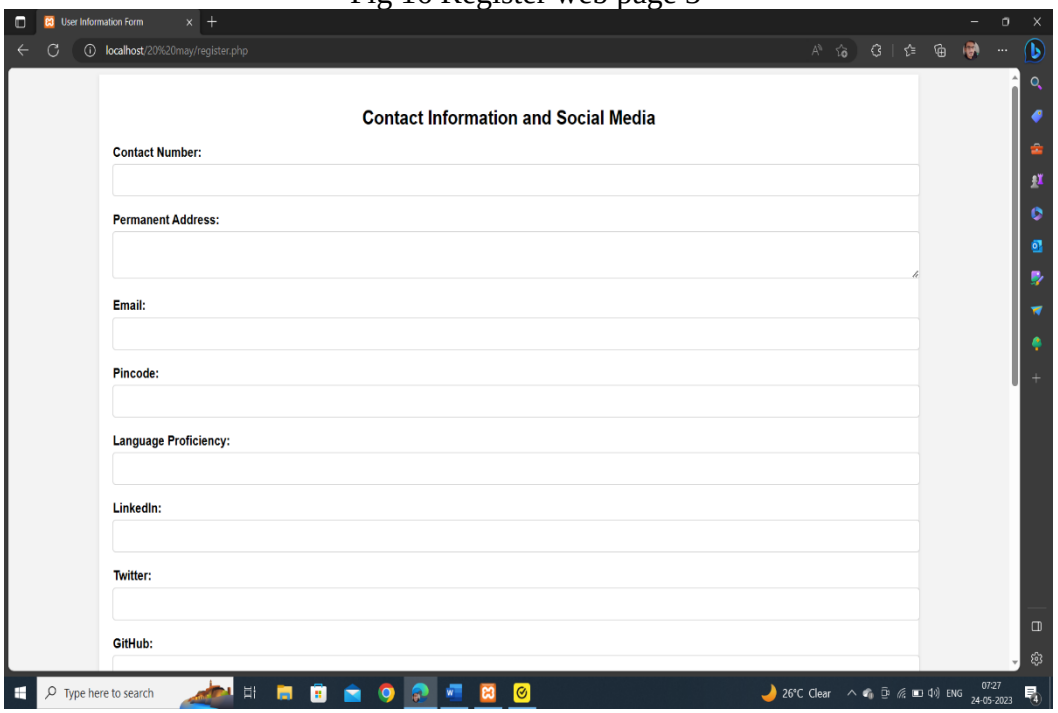

Fig 17 Register web page 4

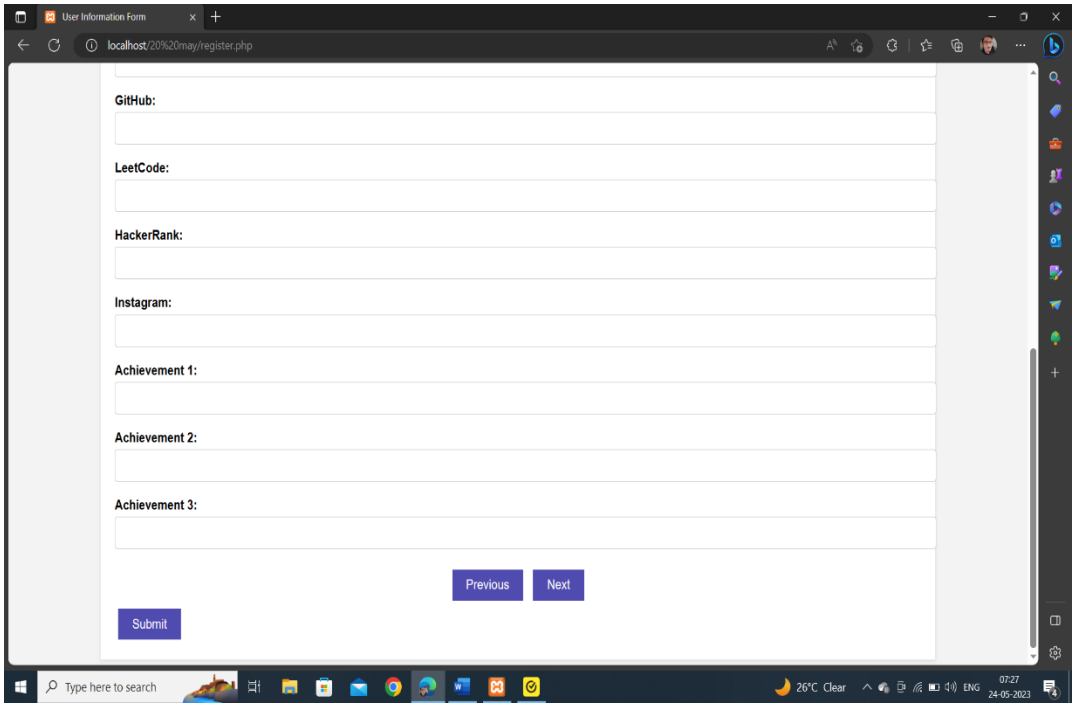

Fig 18 Register web page 5

5.4 Profile Page

This page represents a profile page for a user. It is used to create a webpage that displays information about the user, including their name, photo, introduction, achievements, skills, and contact information. Here are a few reasons why this code might be used:

Personal Portfolio: The code can be used by individuals, such as freelancers, designers, developers, or professionals from various fields, to showcase their work and skills. It provides a centralized platform to display their achievements and provide contact details for potential clients or employers.

Networking: The profile page can be used on social networking sites or professional platforms to create a professional online presence. It allows users to

highlight their expertise and connect with others in their industry.

Job Applications: When applying for jobs, having a profile page can be advantageous. It provides a comprehensive overview of the applicant's qualifications, skills, and achievements, which can help employers assess their suitability for a position.

Personal Branding: A profile page helps individuals establish their personal brand and promote themselves online. By showcasing their accomplishments and skills, they can attract opportunities and build a reputation within their industry.

Online Resume: The profile page can serve as an online resume, providing a visually appealing and interactive representation of an individual's qualifications. It can include additional elements such as the timeline, which displays the user's educational and professional journey.

Overall, the code is used to create an aesthetically pleasing and informative profile page, allowing individuals to present themselves professionally, highlight their skills and achievements, and facilitate networking and career opportunities.

Fig 19 Profile web page 1

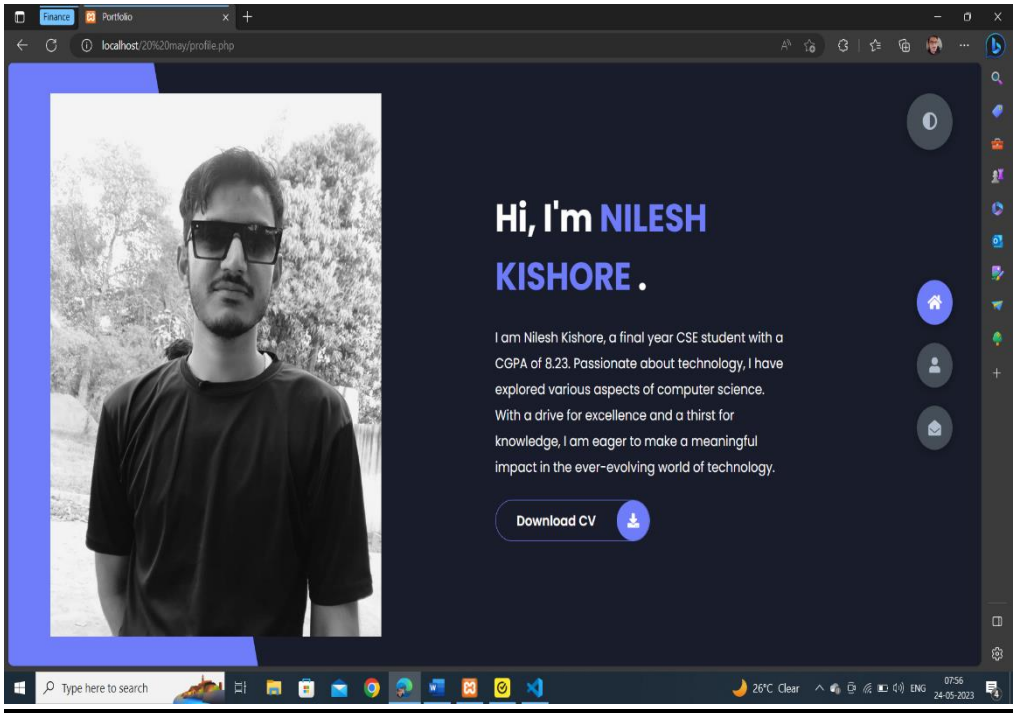

Fig 20 Profile web page 2
(Night View)

Fig 21 Profile web page 3

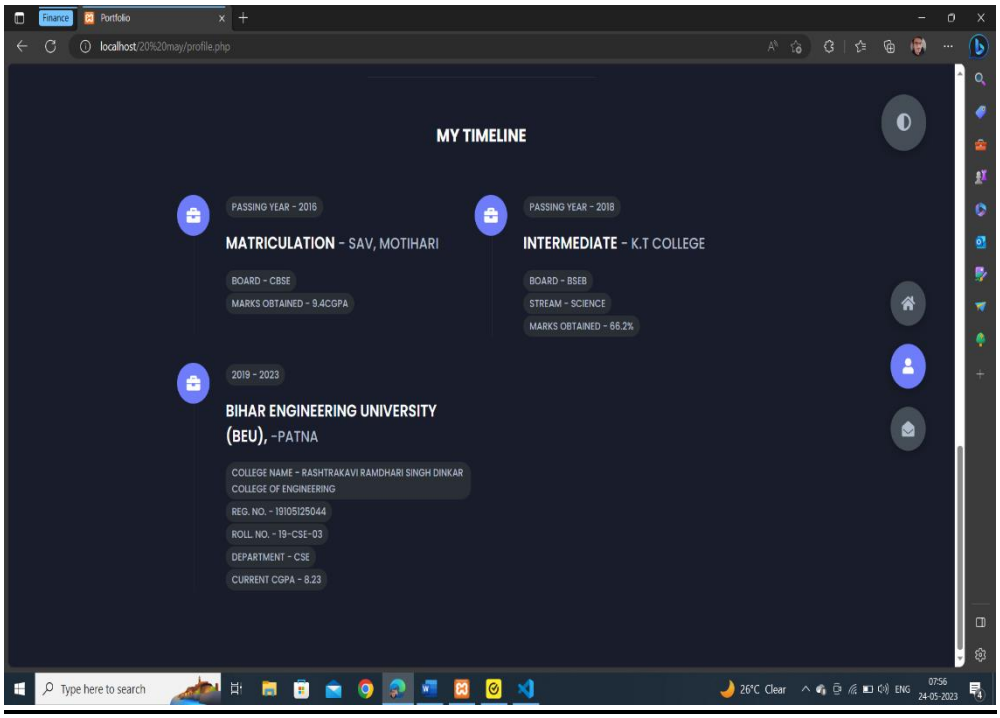

Fig 22 Profile web page 4

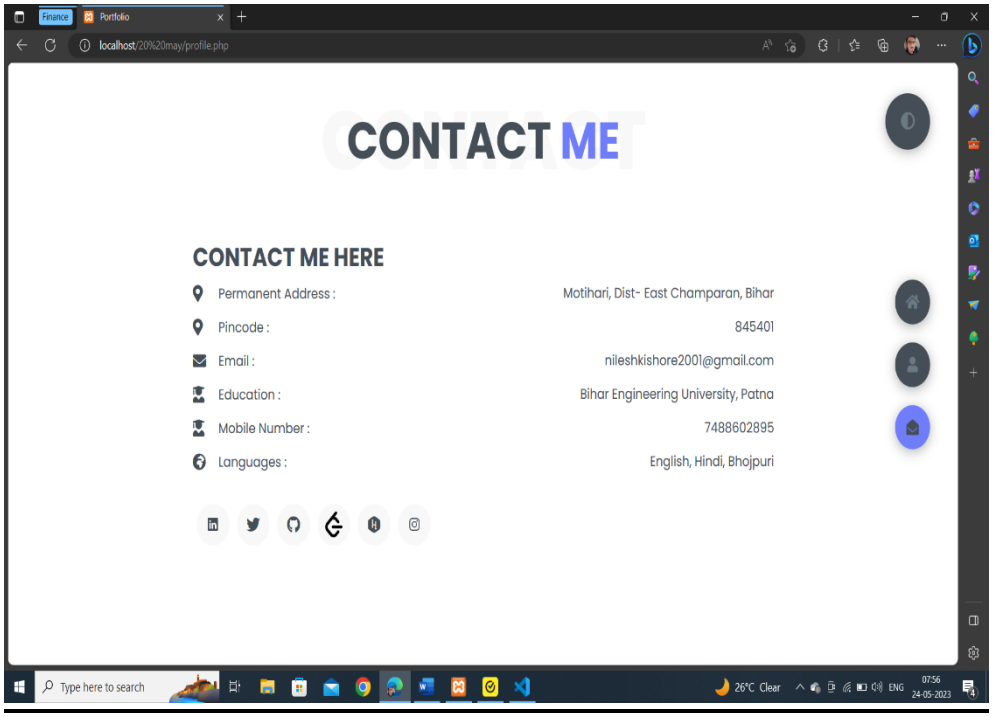

Fig 23 Profile web page 5

5.5 Responsiveness Of Page

The responsiveness of a website refers to its ability to adapt and display properly on various devices and screen sizes, including desktop computers, laptops, tablets, and smartphones. A responsive website is designed and developed to provide an optimal viewing experience, ensuring that users can easily navigate and interact with the site regardless of the device they are using.

Key characteristics of a responsive website include:

Fluid Layout: The website layout adjusts dynamically based on the screen size and resolution, utilizing flexible grids and proportion-based elements. This ensures that content is displayed appropriately and maintains readability across different devices.

Media Queries: Responsive websites use CSS media queries to detect the characteristics of the user's device and apply specific styles accordingly. Media queries enable the website to adapt its layout, font sizes, and other design elements

to provide an optimal user experience.

Flexible Images and Media: Images, videos, and other media elements are scaled and resized based on the available screen space, preventing them from overflowing or getting cut off. This allows users to view and interact with media seamlessly on any device.

Touch-Friendly Navigation: Responsive websites often include touch-friendly navigation menus and buttons to accommodate mobile and tablet users who interact with the site through touch gestures. This ensures ease of use and prevents any usability issues.

Fast Loading Times: Responsive websites prioritize performance by optimizing page load times across different devices. This involves minimizing file sizes, leveraging browser caching, and employing other techniques to deliver a smooth and efficient browsing experience.

Benefits of a responsive website include:

Improved User Experience: Responsive design ensures that users can access and navigate a website easily, regardless of the device they are using. This leads to higher user satisfaction and engagement.

Increased Mobile Traffic: With the growing usage of smartphones and tablets, having a responsive website is crucial to capture mobile traffic. Mobile users are more likely to stay on a website that provides a seamless experience, reducing bounce rates.

Better SEO Performance: Search engines, such as Google, prioritize responsive websites in their rankings. By providing a positive user experience across devices, responsive sites tend to rank higher in search engine results, leading to increased visibility and organic traffic.

Cost and Time Efficiency: Rather than developing separate websites for different

devices, a responsive design allows for a single website that adapts to various screen sizes. This reduces development and maintenance costs and saves time by eliminating the need for managing multiple versions of the same content.

In summary, a responsive website is designed to adapt and provide an optimal user experience across various devices and screen sizes. It ensures usability, accessibility, and improved performance, leading to higher user satisfaction, increased traffic, and better search engine visibility.

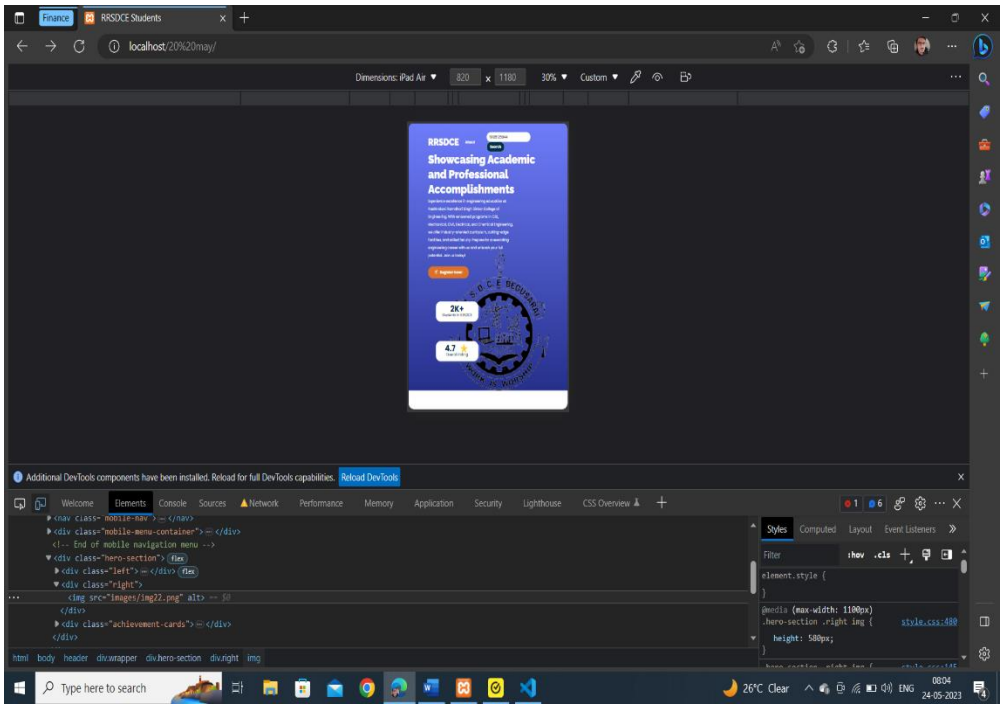

Fig 24 Responsive View 1

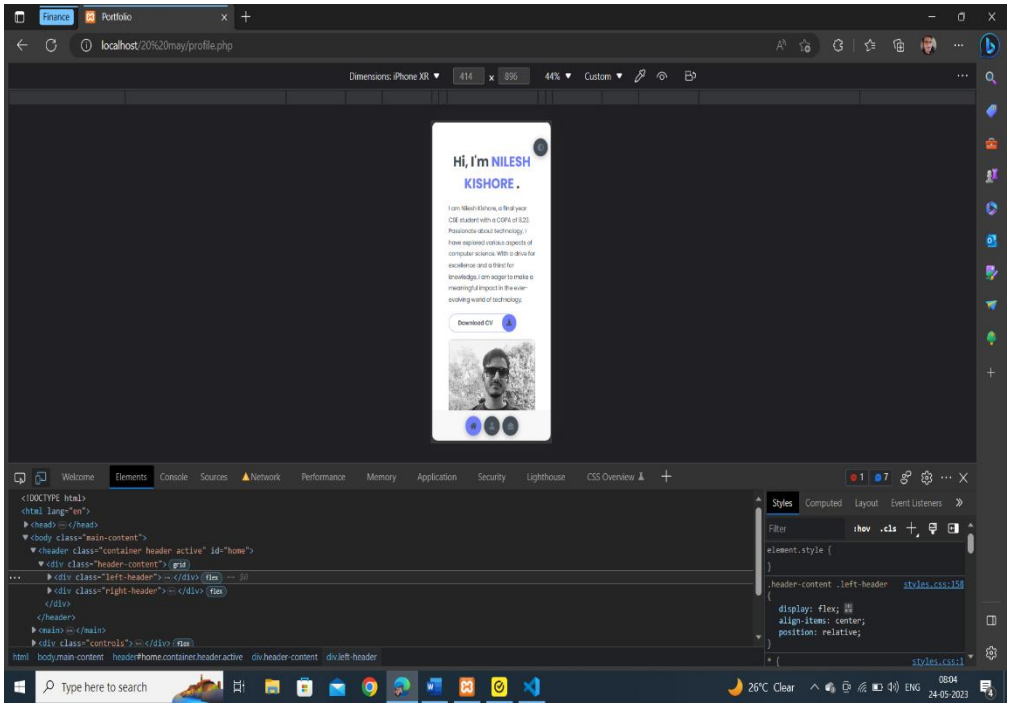

Fig 25 Responsive View 2

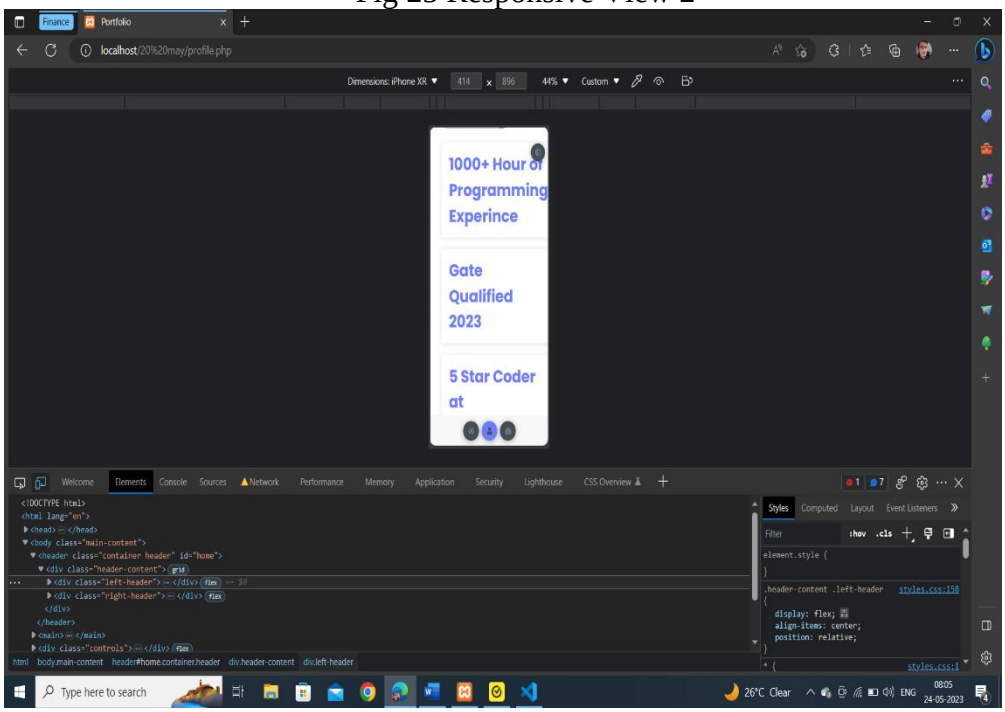

Fig 26 Responsive View 3

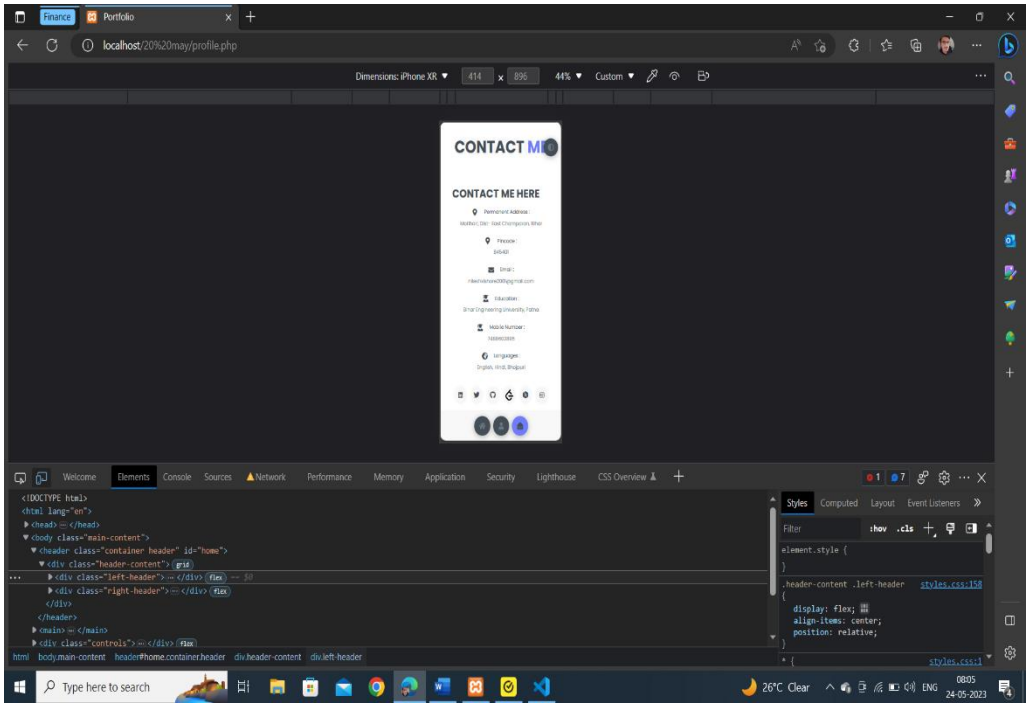

Fig 27 Responsive View 3

6. SYSTEM TESTING

System testing is a crucial phase in the Software Development Life Cycle (SDLC) that focuses on evaluating and verifying the entire system or software application as a whole. It is performed after unit testing, integration testing, and other lower-level testing activities have been completed. The goal of system testing is to ensure that the system meets the specified requirements, functions correctly, and performs as expected in a real-world environment.

6.1 Unit Test

User Registration Unit Test:

Test Case 1: Verify that a new user can successfully register with valid

credentials.

Test Case 2: Ensure that an error is displayed for invalid inputs during the registration process.

Login Unit Test:

Test Case 1: Validate that a registered user can log in with correct credentials.

Test Case 2: Check that an error message is displayed for incorrect login credentials.

Portfolio Creation Unit Test:

Test Case 1: Verify that a user can create a new portfolio with valid information.

Test Case 2: Ensure that appropriate error messages are displayed for invalid inputs during portfolio creation.

Image Uploading Unit Test:

Test Case 1: Validate that an image can be successfully uploaded to the portfolio.

Test Case 2: Verify that the appropriate error message is displayed for unsupported file formats.

Information Display Unit Test:

Test Case 1: Check that the student's basic information (name, email, etc.) is displayed correctly on the profile page.

Test Case 2: Validate that the student's education details are properly displayed on the education page.

Navigation Unit Test:

Test Case 1: Ensure that navigation links between different pages of the website work correctly.

Test Case 2: Validate that the active page is properly highlighted in the navigation menu.

It's important to ensure that the tests are repeatable, isolated from external

dependencies, and provide reliable results. You can also consider incorporating test automation tools or scripts to automate the execution of unit tests, allowing for regular and continuous testing as you make changes to the codebase.

Remember, unit testing is an iterative process, and you should continuously add and update tests as you enhance or modify the website's functionality. Regularly running the unit tests as part of your development workflow will help ensure that the website's individual components are functioning correctly and that any changes or additions do not introduce regressions or issues.

6.2 Integration Test

Integration testing is a software testing technique that focuses on testing the interactions and integration between different components or modules of a system to ensure they work together correctly. It verifies that the integrated components function as expected and meet the specified requirements when combined.

Here are the key aspects of integration testing:

Integration Points: Identify the integration points or interfaces between different components, modules, or subsystems of the system. This could include APIs, databases, messaging systems, external services, or any other interactions between system elements.

Integration Test Cases: Define integration test cases that cover various scenarios where components interact with each other. These test cases should verify the correct flow of data, proper communication, and the expected behaviour of integrated components.

Test Data Preparation: Prepare test data to simulate real-world scenarios and inputs that components exchange during integration. This includes both valid and invalid data to test boundary conditions, error handling, and exception cases.

Test Environment Setup: Set up a test environment that mirrors the production environment as closely as possible. This includes configuring databases, networks,

external services, and any other dependencies required for integration.

Test Execution: Execute the integration test cases by simulating the interactions between components. This involves triggering actions or events in one component and verifying the responses or outputs from the connected components.

Error Handling and Exception Testing: Validate the system's behaviour when encountering errors or exceptions during integration. Test cases should cover scenarios such as connection failures, timeouts, data inconsistencies, and any other error conditions that may arise during integration.

Dependency Management: Test the handling of dependencies and ensure that components can function correctly when dependent modules or services are unavailable or changed.

Test Orchestration: Coordinate the execution of integration tests in a controlled and organized manner, ensuring proper sequencing of test cases and managing dependencies between components.

Mocking and Stubs: Use mocks or stubs to simulate the behaviour of dependent components that are not available or difficult to test in the integration environment. This allows for isolated testing of specific components without relying on the complete system.

Reporting and Defect Management: Document and track any issues or defects encountered during integration testing. Provide detailed reports and log any discrepancies, failures, or unexpected behaviour observed during the integration process.

Integration testing helps identify issues that may arise when combining different components or modules, such as compatibility problems, communication errors, data inconsistencies, and functional defects. By performing integration testing,

you can ensure that the system functions as a cohesive whole, delivering the intended functionality and meeting the overall system requirements.

It's important to note that integration testing is typically performed after unit testing and before system testing. It bridges the gap between individual component testing and the overall system testing, ensuring that the integrated components work harmoniously together.

7. FUTURE SCOPE

The college student portfolio website we have built has great potential for future expansion and integration with our college website. Here are some future scope considerations for our website:

Enhanced User Experience: We can continuously work on improving the user interface and experience of our website. We can incorporate modern design trends, responsiveness for different devices, and intuitive navigation to make it more user-friendly.

Expanded Functionality: We can consider adding additional features and functionalities to our website to provide more value to our users. This could include features like a blog section, project showcase, event calendar, discussion forums, or integration with social media platforms.

Integration with College Systems: We can integrate our student portfolio website with our existing college systems and platforms. This could involve connecting with our college's student information system, course management system, or online learning platforms to provide a seamless user experience and streamline data exchange.

Collaboration and Networking: We can enable students to connect and collaborate with each other through our website. We can implement features such as messaging, discussion boards, or a networking platform to foster communication and collaboration among students from different courses or departments.

Alumni Network: We can develop an alumni network section within our website to facilitate networking and interaction between our current students and alumni. This can include features like alumni profiles, job/internship opportunities, and mentorship programs.

Personalization and Customization: We can provide our users with the ability to customize and personalize their portfolios. We can offer options to choose themes, layouts, color schemes, and the ability to showcase specific projects or achievements based on individual preferences.

Integration with College Website: We can work towards integrating our student portfolio website with our official college website. This integration can provide a seamless experience for our students, enabling them to access their portfolios directly through our college's website and leveraging our existing authentication systems.

Analytics and Insights: We can implement analytics and tracking mechanisms to gather data about user interactions, popular portfolio sections, and user behaviour. We can use this data to gain insights into user preferences, identify areas for improvement, and make data-driven decisions for our website enhancements.

Mobile Application: We can consider developing a mobile application for our student portfolio website, allowing our users to access their portfolios on the go

and providing additional convenience and flexibility.

Continuous Improvement: We can regularly update and enhance our website based on user feedback, emerging technologies, and changing user needs. We can stay up to date with the latest trends in web development and incorporate new features and functionalities to keep our website relevant and competitive.

By considering these future scope aspects, we can ensure that our college student portfolio website continues to evolve and meet the changing needs of our students and the college community. We can regularly gather feedback from our users and stakeholders to identify areas for improvement and prioritize new features and integrations.

8. CONCLUSION

In conclusion, the college student portfolio website we have developed holds immense potential for future growth and integration with our college website. By focusing on enhancing user experience, expanding functionality, and integrating with college systems, we can create a comprehensive platform that caters to the needs of our students and fosters collaboration and networking. The ability to personalize and customize portfolios, along with seamless integration with our college website, will provide a cohesive user experience. By leveraging analytics and insights, we can make data-driven decisions to continuously improve our website and stay up to date with emerging technologies and trends. With the inclusion of a mobile application and the commitment to continuous improvement, our college student portfolio website will remain relevant and meet the evolving needs of our student community. By considering the future scope outlined above, we can ensure that our website remains a valuable asset for our college and contributes to the overall success of our students' academic and professional journeys.

9. REFERENCE

SOURCE CODE LINK –

https://github.com/Nileshkishore/COLLEGE_MINOR