InstituteHub: A Comprehensive Web Application for Institute Activity Management

An Internship report submitted in partial fulfillment of the requirements for the Award of the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Jadam Aruna Bharathi	**218W1A0519**
Jampani Jaya Prudhvi	**218W1A0521**
Makkena Divya Harshitha	**218W1A0533**
Pasam Namitha	**218W1A0544**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

V.R Siddhartha Engineering College
(Autonomous)

Approved by AICTE, NAAC A+, NBA Accredited

Affiliated to Jawaharlal Nehru Technological University, Kakinada

Vijayawada 520007

April , 2025

VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE

(Autonomous, Accredited with ‘A+’ grade by NAAC)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Internship report entitled **”InstituteHub: A Comprehensive Web Application for Institute Activity Management”** being submitted by

Jadam Aruna Bharathi	**218W1A0519**
Jampani Jaya Prudhvi	**218W1A0521**
Makkena Divya Harshitha	**218W1A0533**
Pasam Namitha	**218W1A0544**

in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING, from **Dec 2025 to Apr 2025**.

Dr.G.Anuradha, M.Tech, Ph.D
Department Internship Coordinator

Dr.D.Rajeswara Rao, M.Tech, Ph.D
Professor & HOD, CSE

Declaration

I hereby declare that the dissertation entitled **”InstituteHub: A Comprehensive Web Application for Institute Activity Management”** submitted for the B.Tech Degree is my work and the dissertation has not formed the basis for the award of any degree, associates, fellowship or any other similar titles.

Jadam Aruna Bharathi (218W1A0519)

Jampani Jaya Prudhvi (218W1A0521)

Place: Vijayawada **Makkena Divya Harshitha (218W1A0533)**

Date: 10/04/2025 **Pasam Namitha (218W1A0544)**

Acknowledgement

Behind every achievement lies an unfathomable sea of gratitude to those who activated it,without whom it would ever have come into existence. To them we lay the words of gratitude imprinted with us.

I owe my sincere thanks to our respected Principal, **Dr. A. V. Ratna Prasad** for providing us the necessary trainings and his immense support to shape us into being industry ready.

I would like to thank **Dr. D. Rajeswara Rao, Head of the Department, Computer Science and Engineering**, for the endless support and that was provided which made us challenge ourselves and accept such impeccable opportunities.

I would like to extend my gratitude to the department faculty, especially **Dr. G. Anuradha, Associate Professor & Internship Co-Ordinator** for this opportunity support, and guidance throughout the internship.

I would like to extend my gratitude to team for providing me the opportunity for this internship. I owe my acknowledgements to an equally long list of people who helped me in this internship. I would like to thank the various mentors associated with me in various use cases throughout the internship.

Jadam Aruna Bharathi (218W1A0519)

Jampani Jaya Prudhvi (218W1A0521)

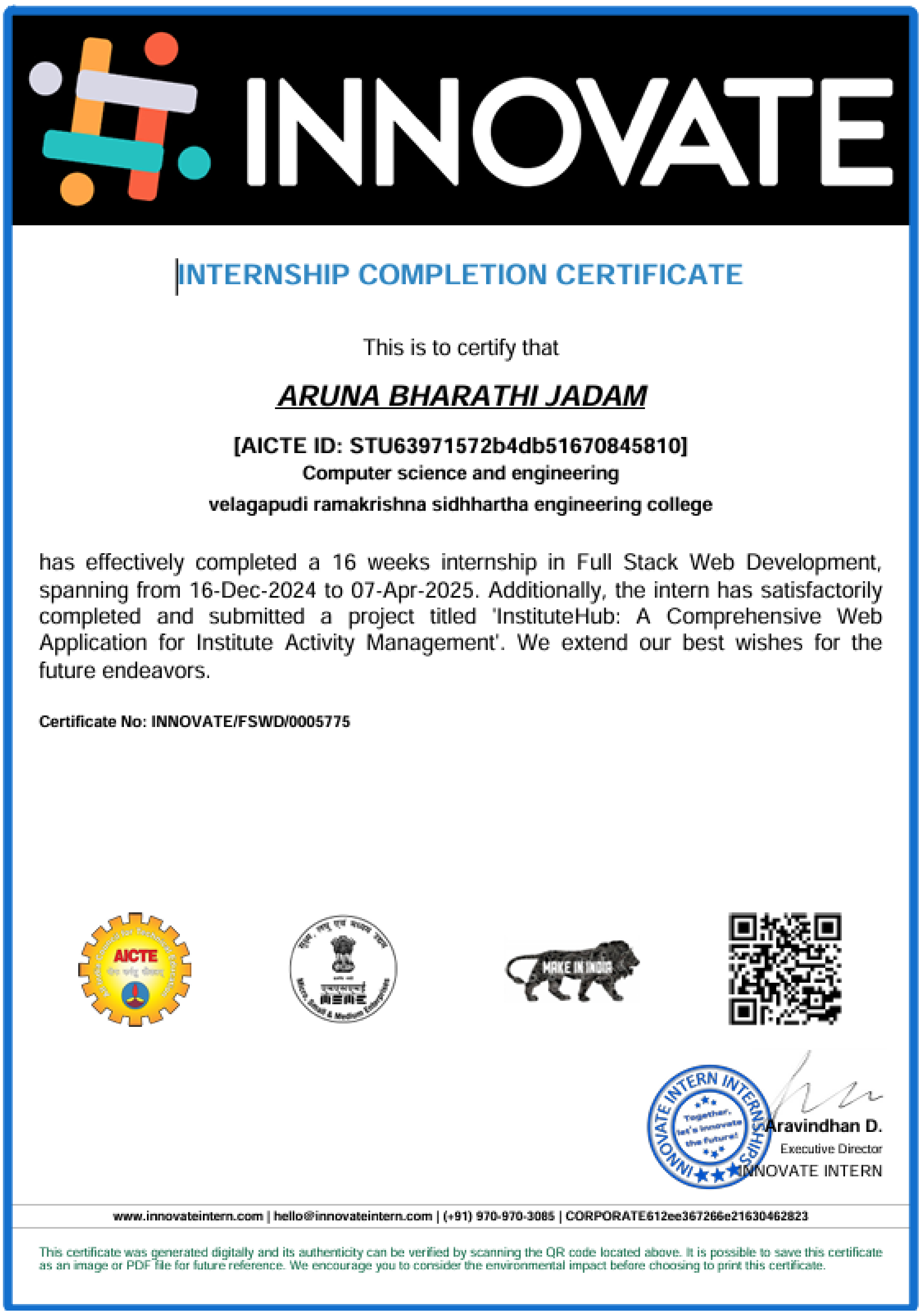
Makkena Divya Harshitha (218W1A0533)

Pasam Namitha (218W1A0544)

Place: Vijayawada

Date: 10/04/2025

Internship Offer Letter by Innovate Intern



INNOVATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

ARUNA BHARATHI JADAM

[AICTE ID: STU63971572b4db51670845810]

Computer science and engineering

velagapudi ramakrishna sidhhartha engineering college

has effectively completed a 16 weeks internship in Full Stack Web Development, spanning from 16-Dec-2024 to 07-Apr-2025. Additionally, the intern has satisfactorily completed and submitted a project titled 'InstituteHub: A Comprehensive Web Application for Institute Activity Management'. We extend our best wishes for the future endeavors.

Certificate No: INNOVATE/FSWD/0005775

AICTE

MAKE IN INDIA

INNOVATE INTERN INTERNSHIPS ★★★ Together, let's innovate the future!

[illegible]

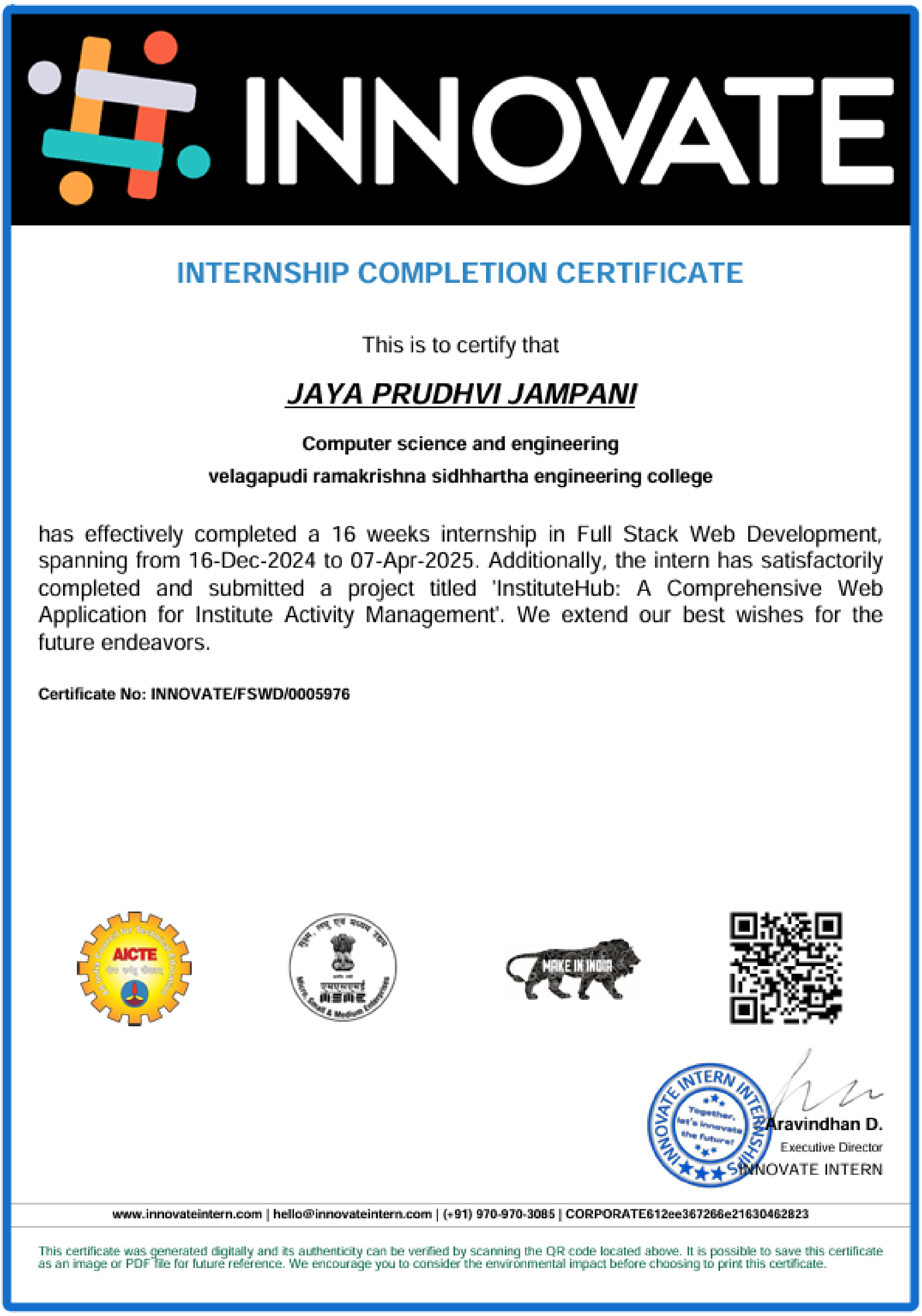
Aravindhan D.

Executive Director

INNOVATE INTERN

www.innovateintern.com | hello@innovateintern.com | (+91) 970-970-3085 | CORPORATE612ee367266e21630462823

This certificate was generated digitally and its authenticity can be verified by scanning the QR code located above. It is possible to save this certificate as an image or PDF file for future reference. We encourage you to consider the environmental impact before choosing to print this certificate.



INNOVATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

JAYA PRUDHVI JAMPANI

Computer science and engineering

velagapudi ramakrishna sidhhartha engineering college

has effectively completed a 16 weeks internship in Full Stack Web Development, spanning from 16-Dec-2024 to 07-Apr-2025. Additionally, the intern has satisfactorily completed and submitted a project titled 'InstituteHub: A Comprehensive Web Application for Institute Activity Management'. We extend our best wishes for the future endeavors.

Certificate No: INNOVATE/FSWD/0005976

AICTE

MSME

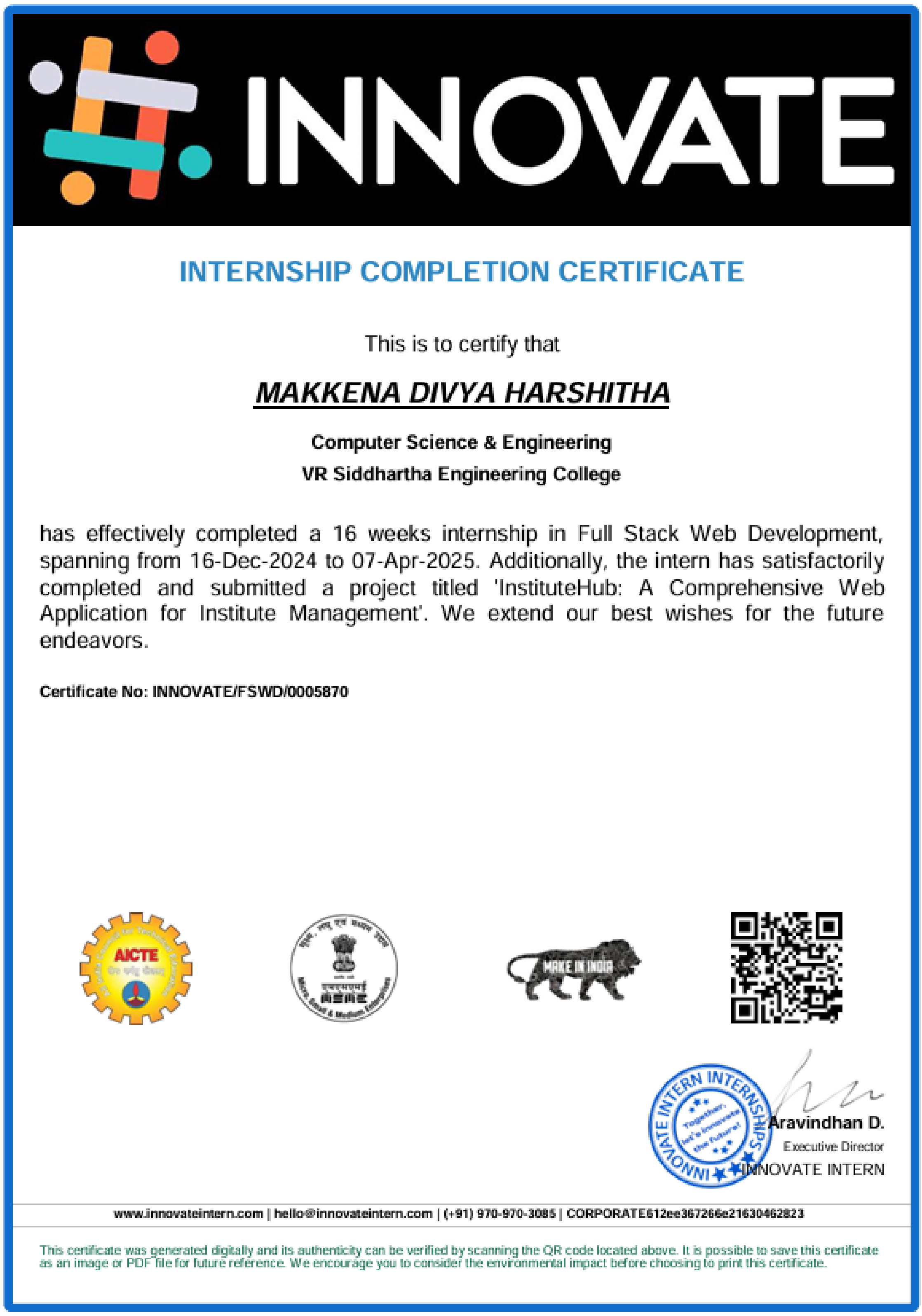
MAKE IN INDIA

INNOVATE INTERN INTERNSHIPS
Together, let's innovate the future!

[illegible]

Aravindhan D.
Executive Director
INNOVATE INTERN

www.innovateintern.com | hello@innovateintern.com | (+91) 970-970-3085 | CORPORATE612ee367266e21630462823



INNOVATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

MAKKENA DIVYA HARSHITHA

Computer Science & Engineering

VR Siddhartha Engineering College

has effectively completed a 16 weeks internship in Full Stack Web Development, spanning from 16-Dec-2024 to 07-Apr-2025. Additionally, the intern has satisfactorily completed and submitted a project titled 'InstituteHub: A Comprehensive Web Application for Institute Management'. We extend our best wishes for the future endeavors.

Certificate No: INNOVATE/FSWD/0005870

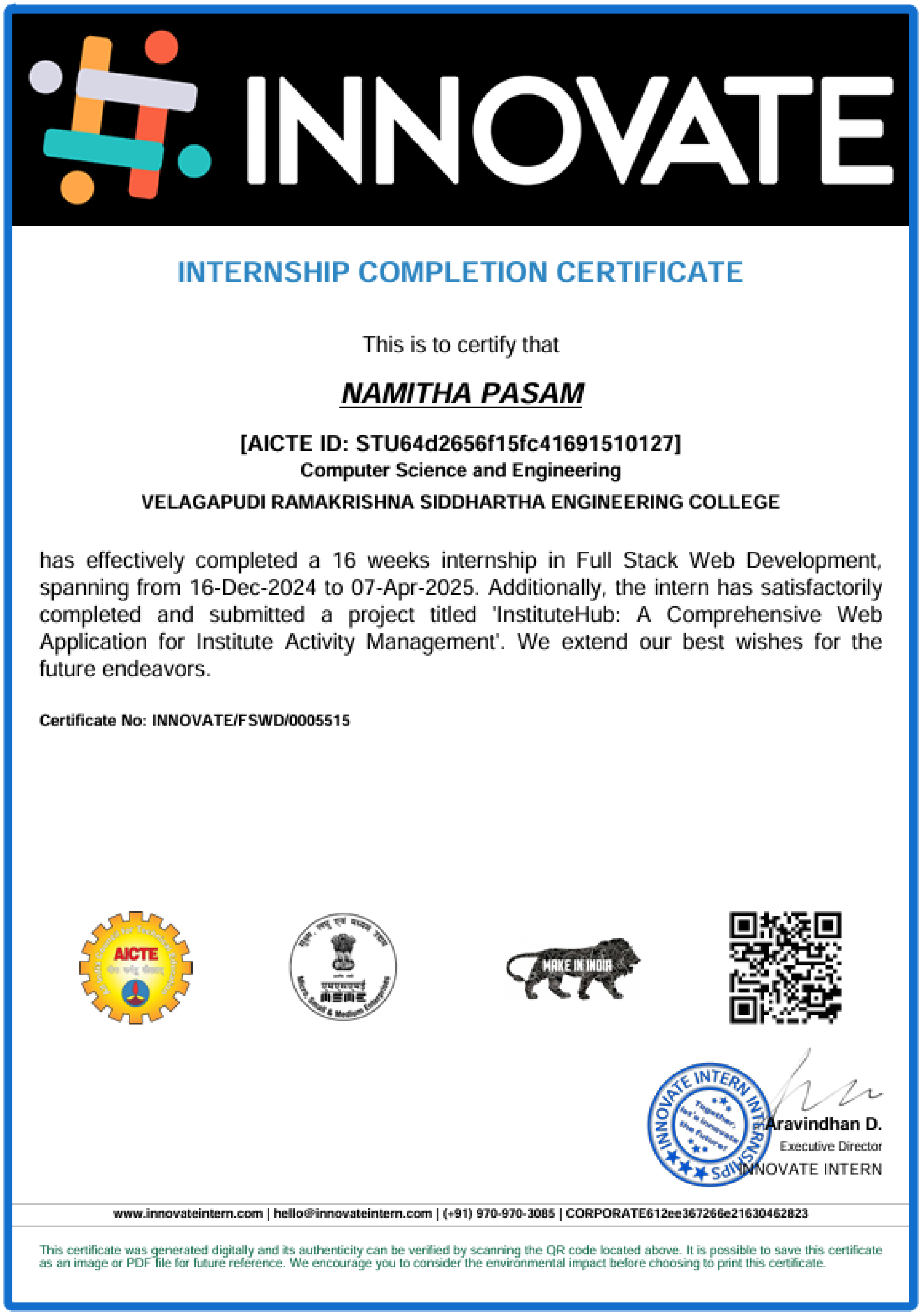
INNOVATE INTERN INTERNSHIPS
Together, let's innovate the future!

[illegible]

Aravindhan D.
Executive Director
INNOVATE INTERN

www.innovateintern.com | hello@innovateintern.com | (+91) 970-970-3085 | CORPORATE612ee367266e21630462823

This certificate was generated digitally and its authenticity can be verified by scanning the QR code located above. It is possible to save this certificate as an image or PDF file for future reference. We encourage you to consider the environmental impact before choosing to print this certificate.



INNOVATE

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

NAMITHA PASAM

[AICTE ID: STU64d2656f15fc41691510127]

Computer Science and Engineering

VELAGAPUDI RAMAKRISHNA SIDDHARTHA ENGINEERING COLLEGE

has effectively completed a 16 weeks internship in Full Stack Web Development, spanning from 16-Dec-2024 to 07-Apr-2025. Additionally, the intern has satisfactorily completed and submitted a project titled 'InstituteHub: A Comprehensive Web Application for Institute Activity Management'. We extend our best wishes for the future endeavors.

Certificate No: INNOVATE/FSWD/0005515

INNOVATE INTERNSHIPS
Together, let's innovate the future!

[illegible]

Aravindhan D.
Executive Director
INNOVATE INTERN

www.innovateintern.com | hello@innovateintern.com | (+91) 970-970-3085 | CORPORATE612ee367266e21630462823

This certificate was generated digitally and its authenticity can be verified by scanning the QR code located above. It is possible to save this certificate as an image or PDF file for future reference. We encourage you to consider the environmental impact before choosing to print this certificate.

Company Profile and External Guide Details

Innovate Intern is a technology company focused on equipping aspiring developers and professionals with hands-on training in full stack development. Established to bridge the gap between academic learning and industry requirements, Innovate Intern provides structured programs that help individuals gain practical experience in modern software development.

The company specializes in full stack development, covering both front-end and back-end technologies. It offers training in frameworks such as React.js, Angular, Node.js, Express.js, and Django, along with database management systems like MySQL, PostgreSQL, and MongoDB. By integrating cloud computing, DevOps, and API development, Innovate Intern ensures that participants develop the skills needed to build scalable and efficient web applications.

Innovate Intern collaborates with educational institutions, enterprises, and industry experts to create learning pathways that align with the evolving demands of the tech industry. Through mentorship, project-based learning, and career support, the company prepares individuals for successful careers in software development.

With a focus on innovation and quality education, Innovate Intern continues to advance technology training, helping professionals stay competitive in the ever-evolving digital landscape.

Abstract

Effective management of academic and extracurricular activities is essential for the smooth functioning of educational institutions. Traditional methods of managing institute operations—such as event coordination, communication, and resource allocation—often involve manual processes that are time-consuming and prone to inefficiencies. This project presents InstituteHub, a centralized web-based platform developed to streamline these tasks through digital transformation. The system features role-based dashboards for students and faculty, real-time event registration tracking, and an intuitive user interface for seamless interaction. By integrating modern web technologies and database management, InstituteHub ensures efficient event scheduling, transparent participant management, and secure login-based access. Compared to conventional systems, the platform offers greater transparency, faster operations, and improved user engagement. Performance evaluation indicates that the system effectively simplifies communication and coordination in educational settings, making it a scalable and valuable tool for institutes aiming to optimize their activity management workflows.

Keywords: Institute Management System, Event Registration, Student and Faculty Dashboard, Role-Based Access Control, Web Application Development, Real-Time Data Handling.

Table of Contents

List of Figures

Chapter 1

FULL STACK DEVELOPMENT

This chapter introduces basic concepts and provides a comprehensive understanding of the project's motivation, elucidates its specific objectives, and delves into the scope of the undertaking.

1.1 Introduction to WWW

This section introduces covers fundamental concepts of the World Wide Web and its key components.

1.1.1 Protocol

A protocol is basically a standard that makes it possible to connect, communicate, and transfer data between two locations on a network. In other words, the protocols are digital languages that are implemented as networking algorithms. Users use several networks and network protocols when surfing.

1. **HTTP:** HyperText Transfer Protocol (HTTP) is a standard for information transfer on the internet. It defines the formatting and transmission of hypertext documents and specifies how web browsers should respond to requests. HTTP is used every time a browser loads a web page, transferring text, images, and multimedia over the World Wide Web.

2. **HTTPS:** HyperText Transfer Protocol Secure (HTTPS) is a secure version of HTTP that encrypts data using SSL (Secure Sockets Layer). Websites handling sensitive information, such as passwords or credit card details, use HTTPS. A website requires an SSL certificate issued by a trusted Certificate Authority (CA) to enable HTTPS. Secure websites display a padlock icon in the browser.

3. **FTP:** File Transfer Protocol (FTP) is used for transferring files online. Web developers commonly use FTP to upload website updates to a server. Unlike HTTP, which displays files in a browser, FTP is strictly for file transmission between computers—from a local computer to a remote server or vice versa.

4. **TCP/IP:** Transmission Control Protocol/Internet Protocol (TCP/IP) enables communication between different computer systems. IP assigns a unique address to each device on the internet, while TCP ensures data is transmitted, received, and reassembled correctly.

5. **SMTP:** Simple Mail Transfer Protocol (SMTP) is used for sending and distributing outbound emails. It reads the recipient's email address from the message header and places the email in an outgoing mail queue. Once delivered, SMTP removes the email from the outgoing list.

6. **TELNET:**Terminal Network (TELNET) is a TCP/IP protocol that allows a local computer to connect to a remote computer. Using the client/server model, the local computer runs a TELNET client, while the remote machine runs a TELNET server, enabling users to access and control remote systems.

7. **POP3:** Post Office Protocol 3 (POP3) is an email protocol that enables users to retrieve and manage emails from a mail server to their local device. It uses client and server message access agents (MAAs) to access messages stored in the recipient's mailbox.

1.1.2 Web Program

Web program is a code for a web page. You need a web browser to run a web page source code. Just like any program you require different languages to code a webpage. HTML is the most basic language used to create a webpage.While there are many different programming languages, the most common ones used in web development are JavaScript, HTML, CSS etc.,

1.1.3 Secure Connection

To protect the data being transferred from one machine to another, , encryption, authentication, and data integrity are essential for secure connections. Encryption ensures that unauthorized individuals cannot access sensitive data, authentication verifies the identity of users, and integrity ensures that data is not tampered with during transmission. These principles are integrated into communication protocols such as HTTPS, SSL/TLS, and VPNs. Additionally, firewalls and antivirus software provide extra layers of security by blocking malicious access and preventing cyber threats.

1.1.4 Web Developments tools

Web development involves building websites and applications for the internet or intranets. Web developers write the code that defines a site's style, layout, and interactivity. Web development consists of three main layers:

1. **Frontend Development (Client-Side Scripting):** Manages the user interface, including menus, forms, and layout.

   Stores and manages website data efficiently.

2. **Backend Development (Server-Side Scripting):** Handles background operations, processes user requests, and sends responses.

3. **Database Technolog**y: Web development tools (devtools) help developers test and debug code.

Most web browsers like Chrome, Firefox, and Edge offer built-in tools and extensions to assist developers in working with technologies like HTML, CSS, JavaScript, and the DOM.

1.1.5 Web Browser

A web browser, sometimes known as a browser, is a piece of software used to access content on the World Wide Web. Chrome, Firefox, Safari, Internet Explorer, Microsoft

Edge, Opera, and UC Browser are the most widely used browsers.

1.1.6 Web Server

A web server is a device or software that delivers web content or services to users via the internet. It processes HTTP requests and provides web pages or data in response. Web servers are essential for web hosting and vary based on customer needs and traffic demands.

Types of Web Servers:

1. **Apache Web Server:** Open-source, widely used, customizable, supports multiple OS, and handles multiple requests using multithreading.
2. **IIS Web Server:** Microsoft-developed, compatible with all Windows platforms, and maintained by Microsoft.
3. **Nginx Web Server:** Open-source, high-performance, scalable, and uses an event-driven architecture for efficient memory management.
4. **LiteSpeed Web Server:** A high-performance, commercial alternative to Apache that enhances speed and reduces costs.

1.1.7 Web Designing

A well-designed website should engage visitors, convey its message effectively, and offer a positive user experience. Key principles include:

1. **Website Purpose:** Each page should have a clear goal, whether providing information, entertainment, generating leads, or selling products.
2. **Simplicity:** Use a harmonious color scheme (max five colors) to enhance user experience. Use readable fonts (max three types) that reflect brand identity. High-quality visuals create professionalism and credibility.
3. **Navigation:** Simple, consistent navigation improves usability and keeps visitors engaged.

4. **F-Shaped Pattern Reading:** Users scan websites in an F-pattern (top and left focus); designs should accommodate this reading behavior.
5. **Visual Hierarchy:** Important elements should stand out using size, color, contrast, and whitespace.
6. **Content:** Compelling and relevant content attracts users and encourages conversions.
7. **Grid-Based Layout:** Organizes content neatly with balanced sections for a structured appearance.
8. **Load Time:** Faster websites retain visitors—optimize images and code for quick loading.
9. **Mobile-Friendly Design:** Responsive layouts ensure websites adapt to various devices for a seamless user experience.

1.2 Architecture of Web Based System

This section introduces the web-based system includes a web browser (client) that requests web pages from a web server. Pages can be static (fixed content) or dynamic (changing content). The system follows a three-tier architecture: the client tier (browser), server tier (processes requests), and database tier (stores data), ensuring seamless web interactions.

1.2.1 Web Browser

A web browser, also called as a browser, is a piece of software used to access content on the World Wide Web. Chrome, Firefox, Safari, Internet Explorer, Microsoft Edge, opera, and UC Browser are the most widely used browsers. A website is a combination of related web pages hosted on a web server.

1.2.2 Web Page

A web page is a document on the World Wide Web that can contain text, images, audio, video, and hyperlinks. Each web page has a unique URL, which allows browsers to access and display it. URLs consist of a protocol identifier (e.g., HTTP) and a resource name (e.g., gmail.com). Types of Web Pages:

1. **Static Web Pages:** Display fixed content, do not change unless modified manually, and are created using HTML CSS.
2. **Dynamic Web Pages:** Content changes based on user interaction and can be interactive.

1.2.3 Tier

A layer (or tier) in a web-based system's architecture refers to its structural division into three parts:

1. **Presentation Layer (Client Layer):** Communicates with browsers using HTML CSS to render static or dynamic content.
2. **Application Layer (Logical Layer):**Handles business logic using programming languages like Java, C, Python, etc. and acts as middleware between the presentation and database layers.
3. **Data Layer (Database Layer) :**Manages and stores data using SQL databases like MySQL, MSSQL, or Oracle.

1.2.4 Web Application architecture (WAA)

Web application architecture defines how applications, databases, and middleware interact over the internet, allowing multiple programs to run simultaneously. When a user enters a web address in the browser and sends a request, the server responds by transmitting the necessary files. The browser then processes these files and displays the requested webpage, enabling user interaction. This architecture is crucial for modern applications, as a significant portion of global network traffic relies on web-based communication. Besides ensuring efficiency, a well-designed web application architecture must also focus on resilience, scalability, stability, and security to provide a seamless user experience.

1.2.5 Components of web applications

Web application components are classified into UI/UX components, which include dashboards, notifications, and settings, and structural components, consisting of client and server sides. The client component, built with JS, CSS, and HTML, runs in the user's browser, while the server component, created with languages like Java, Python, and PHP, manages app logic and databases.

Web application component models vary based on servers and databases. The simplest, one server and one database, lacks reliability. A more stable model uses multiple web servers with one database, ensuring continuous operation unless the database fails. The most effective model has multiple web servers and databases, reducing failure points and improving performance.

Web application architecture is categorized into three types. Single-Page Applications (SPAs) update content dynamically without reloading the page, enhancing user experience. Microservices architecture allows independent development of small services, improving efficiency. Serverless architecture outsources server management to cloud providers, allowing developers to focus on application logic rather than infrastructure.

1.2.6 Advice for Web App Development

Web application that is operational cannot be referred to as ”the finest.” A web application must be more than just functional to be considered outstanding. Numerous considerations should be made during the construction of a web application to guarantee that it can deliver the best performance possible. Web application must:

1. Prevent frequent collisions

2. Have the flexibility to scale up or down.

3. Be easy to use
4. Be able to react more quickly
5. Implement automatic Installations
6. Error Logs
7. Not contain any points of failure.
8. Provide a consistent and uniform response to the question.
9. Adherence to the most recent standards and technologies
10. Increased security measures should be used to reduce the likelihood of harmful intrusions

1.3 Introduction to HTML

HTML, which stands for Hyper Text Markup Language, is the standard markup language for creating layouts and structures for web pages **HTMLIntro**. It is responsible for describing the structure of a webpage through a series of elements, labeled pieces of content such as headings, paragraphs, links, and so on. HTML elements instruct the browser on how to display the content and are crucial for creating visually appealing and user-friendly web pages. This section presents the fundamental concepts of HTML, commonly used to create web pages.

1.3.1 Basic Structure

Code 1.1 is a basic HTML document structure **Structure** with a title, a heading, and a paragraph. which is saved as .htm or .html extension:

```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Page Title</title>
</head>
<body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
</body>
</html>
```

Listing 1.1: Basic structure of a HTML document

1.3.2 Elements

A start tag, some content, and an end tag define an HTML element. The following is a simple example:

```html
<h1>My First Heading</h1>
```

HTML elements can be nested (this means that elements can contain other elements). All HTML documents consist of nested HTML elements. For example:

```html
<body><h1>My First Headings/hl <p>My first paragraph.</p></body>
```

HTML elements with no content are called empty elements. For example:

```html
<p>This is a <br> paragraph with a line break.</p><hr>
```

Remember that HTML is Not Case Sensitive. The HTML standard does not require lowercase tags, but W3C recommends lowercase in HTML.

1.3.3 Attributes

All HTML elements can have attributes. Attributes provide additional information about elements. Attributes are always specified in the start tag. Attributes usually come in name/value pairs like: name=”value”. The <a> tag defines a hyperlink. The href attribute specifies the URL of the page the link goes to. For example:

```html
<a href="https://www.abc.com">Visit Link
```

The <img> tag is used to embed an image in an HTML page. The src attribute specifies the path to the image to be displayed. For example:

```html
<img  src="img.jpg">
```

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels).

```html
<img src="img_girl.jpg" width="500" height="600">
```

The style attribute is used to add styles to an element, such as color, font, size, and more. For example:

```html
<p style="color:red;">This is a red paragraph</p>
```

The title attribute defines some extra information about an element. The value of the title attribute will be displayed as a tooltip when you mouse over the element. For example:

```html
<p title="I'm a tooltip"> This is a paragraph</p>
```

The HTML standard does not require quotes around attribute values. However, W3C recommends quotes in HTML. Double quotes around attribute values are the most common in HTML, but single quotes can also be used.

1.3.4 Formatting

These HTML tags are used for text formatting, including bold, italic, emphasized, and other styles. They enhance readability and convey semantic meaning to content. Following are some formatting tags:

1. **<b>:** Bold text
2. **<strong>:** Important text
3. **<i>:** Italic text
4. **<em>:** Emphasized text
5. **<mark>:** Marked text
6. **<small>:** Small text
7. **<del>:** Deleted text
8. **<ins>:** Inserted text
9. **<sub>:** Subscript text
10. **<sup>:** Superscript text

1.4 Introduction to CSS

CSS (Cascading style sheets) is a language used to describe how HTML elements are displayed on different media **CSSIntro**. It defines styles for web pages, including design, layout, and display on various devices. CSS can control multiple web pages' layouts at once using a rule set consisting of a selector and declaration block. There are three ways of inserting a style sheet, including an external style sheet that can change a website's look by changing one file. An external style sheet is written in any text editor and saved with a .css extension. This section introduces important concepts related to CSS.

1.4.1 Types of CSS

Folowing are the different types of CSS:

1. **Internal CSS:** An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section. For example:

```
<style> p { font-size: 18px; }</style>
```

2. **Inline CSS:** An inline style may be used to apply a unique style for a single element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

```
<p style="font-size: 18px;">This is an example of inline CSS.</p>
```

3. **External CSS:** External CSS involves defining styles in a separate .css file linked to HTML documents using the <link> tag. This method promotes consistency, easy maintenance, and separation of concerns in web development.

```
<link rel="stylesheet" type="text/css" href="styles.css">
```

When there is more than one style specified for an HTML element, all the styles in a page will ”cascade” into a new ”virtual” style sheet by the following rules, where number one has the highest priority: Inline style (inside an HTML element) External and internal style sheets (in the head section) Browser default So, an inline style has the highest priority and will override external and internal styles and browser defaults.

1.4.2 Selectors

Selectors are used to ”find” (or select) the HTML elements you want to style. To group selectors, separate each selector with a comma. The universal selector (*) selects all HTML elements on the page. We can divide them into five categories:

1. Simple selectors (select elements based on name, id, class)
2. Combinator selectors (select elements based on a specific relationship between them)
3. Pseudo-class selectors (select elements based on a certain state)
4. Pseudo-elements selectors (select and style a part of an element)
5. Attribute selectors (select elements based on an attribute or attribute value)

Code 1.2 is a simple example for the five types of selectors:

```css
/* Simple selectors */
h1 {
    color: blue;
}

.paragraph {
    font-size: 16px;
}

/* Combinator selectors */
h1 + p {
    font-style: italic;
}

/* Pseudo-class selectors */
a:hover {
    color: red;
}

/* Pseudo-elements selectors */
p::first-line {
    font-weight: bold;
}
```

```

/* Attribute selectors */
[title="link"] {
    text-decoration: none;
    color: green;
}
```

Listing 1.2: Different types of CSS Selectors

1.4.3 Color

In CSS, a color can be specified as: RGB value using the formula: rgb(red, green, blue). Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255. For example

```
body { background-color: rgb(0, 123, 255); }
```

1.4.4 Box model

There is a box that surrounds each HTML element. It includes content, padding, borders, and margins, with content being the innermost layer **Theboxmodel**. Following are the components of a box model:

1. **Content:** The actual content of the box, such as text, images, or other media.
2. **Padding:** The space between the content and the border.
3. **Border:** The boundary around the padding and content.
4. **Margin:** The space outside the border, creating separation between elements.

Total element width = width + left padding + right padding + left border + right border + left margin + right margin

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

Code 1.3 is an example for box model:

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Box Model Example</title>
    <style>
        .box {
            width: 100px;
            height: 100px;
            background-color: lightblue;
            padding: 20px;
            border: 2px solid blue;
            margin: 20px;
        }
    </style>
</head>
<body>
    <div class="box">Content</div>
</body>
</html>
```

Listing 1.3: CSS Box model

1.5 Introduction to JavaScript

JavaScript (JS) is a lightweight, interpreted programming language commonly used to create dynamic and interactive web pages. It enables developers to implement complex features such as animations, form validation, event handling, and real-time content updates. JavaScript can be used alongside HTML and CSS to enhance web functionality and is widely supported by all modern web browsers.

1.5.1 Basic Structure

Code below demonstrates a basic JavaScript structure that outputs a message using the alert() function:

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Example</title>
</head>
<body>
    <h1>Welcome to JavaScript</h1>
    <script>
        alert("Hello, JavaScript!");
    </script>
```

```html
</body>
</html>
```
Listing 1.4: Basic structure of a JavaScript program

1.5.2 Variables and Data Types

JavaScript variables store data values and can be declared using var, let, or const. JavaScript supports multiple data types, including strings, numbers, and booleans.

```javascript
let name = "John";  // String
var age = 25;       // Number
const isStudent = true;  // Boolean
```
Listing 1.5: JavaScript Variables and Data Types

1.5.3 Operators

JavaScript provides various operators for performing operations on variables and values.

- **Arithmetic Operators**: + (addition), - (subtraction), * (multiplication), / (division), % (modulus)
- **Comparison Operators**: == (equal), === (strict equal), != (not equal), > (greater than), < (less than)
- **Logical Operators**: && (AND), || (OR), ! (NOT)

```javascript
let result = 10 + 5;  // 15
console.log(result > 10 && result < 20);  // true
```
Listing 1.6: JavaScript Operators

1.5.4 Functions

Functions in JavaScript allow code reuse and organization. They are defined using the function keyword.

```javascript
function greet(name) {
    return "Hello, " + name + "!";
}
console.log(greet("Alice")); // Output: Hello, Alice!
```
Listing 1.7: JavaScript Functions

1.5.5 Events

JavaScript interacts with HTML elements through events such as clicks, mouseovers, and keypresses.

```
<button onclick="showMessage()">Click Me</button>
<script>
    function showMessage() {
        alert("Button Clicked!");
    }
</script>
```

Listing 1.8: JavaScript Events

1.5.6 Conditional Statements

Conditional statements like if, else, and switch control the flow of execution based on conditions.

```
let age = 18;
if (age >= 18) {
    console.log("You are an adult.");
} else {
    console.log("You are a minor.");
}
```

Listing 1.9: JavaScript Conditional Statements

1.5.7 Loops

Loops iterate over code blocks multiple times.

For Loop:

```
for (let i = 0; i < 5; i++) {
    console.log("Iteration " + i);
}
```

Listing 1.10: JavaScript For Loop

While Loop:

```
let count = 0;
while (count < 3) {
    console.log("Count: " + count);
    count++;
}
```

Listing 1.11: JavaScript While Loop

1.5.8 DOM Manipulation

JavaScript modifies HTML elements dynamically using the Document Object Model (DOM).

```
<p id="demo">Hello!</p>
<button onclick="changeText()">Change Text</button>
<script>
    function changeText() {
        document.getElementById("demo").innerHTML = "Text Changed!";
    }
</script>
```

Listing 1.12: JavaScript DOM Manipulation

1.5.9 JavaScript Objects

Objects store key-value pairs and are fundamental in JavaScript.

```
let person = {
    firstName: "John",
    lastName: "Doe",
    age: 30,
    greet: function() {
        return "Hello, " + this.firstName;
    }
};
console.log(person.greet()); // Output: Hello, John
```

Listing 1.13: JavaScript Objects

1.5.10 JavaScript Arrays

Arrays hold multiple values and provide various methods for manipulation.

```
let fruits = ["Apple", "Banana", "Cherry"];
console.log(fruits[1]);  // Output: Banana
```

Listing 1.14: JavaScript Arrays

This section provides an overview of JavaScript fundamentals, enabling developers to create interactive web applications.

Chapter 2

CASE STUDY

2.1 Institute Management Portal

2.2 Introduction

An Institute Management Portal is a web application designed to streamline communication and operations among students, faculty, and administrative staff. This project involves developing a dynamic, role-based system using **HTML, CSS (Bootstrap)**, **JavaScript**, and **Python (Flask)** for backend operations.

The portal provides separate interfaces for Admins, Faculty, and Students, allowing seamless registration, login, announcements, task assignments, and event management.

2.3 Background

Managing academic and institutional processes through manual or outdated systems often leads to inefficiencies and miscommunication. With digital transformation in education, there's a growing need for centralized platforms that provide real-time access to information for all stakeholders.

This portal leverages modern web development technologies to bridge the communication gap, automate routine processes, and offer personalized experiences for different user roles within the institution.

2.4 Objective

The primary objective is to develop a **full-stack, role-based college management system** that includes:

- Secure Registration/Login System for Faculty, and Students

- Announcement Dashboard for sharing updates and notices
- Task Module for faculty to assign and students to submit
- Event Management, including registrations and visibility by date
- **Student Event Registration:** Students can register for upcoming college events
- Centralized Interface for managing user data and activities
- Responsive UI built using Bootstrap for mobile and desktop support

2.5 Relevance

- Enables centralized and paperless college communication
- Reduces manual coordination between departments and students
- Allows task/assignment tracking for better academic transparency
- Offers a practical demonstration of full-stack web development
- Supports real-time user interactions with API-style backend architecture

2.6 Problem Statement

Traditional methods of handling academic processes often result in miscommunication, redundant effort, and lack of transparency. Most colleges lack an integrated platform that combines:

- Real-time task assignment & submissions
- Role-specific announcements
- Seamless event registration and visibility

This project addresses these concerns by:

- Providing an intuitive, responsive interface for all users
- Automating user management and content delivery
- Simplifying announcements, events, and task workflows

2.7 Scope of the Project

The portal includes the following core functionalities:

- **User Roles:** Faculty, Student
- **Registration & Login:** Secure account creation and login
- **Dashboard Views:** Personalized views and access for each role
- **Task Management:** Faculty can assign tasks; students can submit responses
- **Announcements:** Admin and faculty can post relevant notices
- **Events Module:** Displays upcoming college events

Technologies Used

- **Frontend:** HTML, CSS (Bootstrap), JavaScript
- **Backend:** Python (Flask)
- **Database:** MySQL (based on project needs)
- **Tools** VS Code, Mysql Workbenc
- **Design:** Responsive UI with mobile support

Chapter 3
Implementation

3.1 Methodology

The methodology adopted for this project involves structured software development principles including planning, system design, modular implementation, and iterative testing. The overall architecture follows a role-based access control mechanism and is divided into the following modules:

3.1.1 User Authentication and Role Management

A centralized login system is developed to authenticate users based on:

- Username
- Password
- Role (Student / Faculty / Admin)

User credentials are stored in a secure SQLite database under a 'users' table. Sessions are used to maintain login state and control access.

3.1.2 Dashboard Functionalities

After login, each user is redirected to their respective dashboard:

- **Student Dashboard:**
  - Fetches user profile and displays all upcoming events from 'events' table.
  - Displays user-specific registrations from 'tech events', 'cultural events', and 'sports events'.
  - Lists announcements posted by faculty.

    - Lists all tasks.

- **Faculty Dashboard:**
    - Fetches all events created by the logged-in faculty.
    - Allows posting new announcements and assigning tasks.
    - Provides a form to filter and view event-wise registrations.

3.1.3 Event Management System

Faculty members can create events by specifying:

- Title
- Description
- Category (Tech / Cultural / Sports)
- Date

These are stored in the ‘events‘ table and made available to students on their dashboard. Students can register for these events, and registrations are stored in their respective category tables.

3.1.4 Announcements and Tasks

Faculty users can post announcements to be viewed by all students. They can also assign tasks to individuals or groups. These are stored in ‘announcements‘ and ‘tasks‘ tables respectively and are displayed on the student dashboard.

3.1.5 View Registrations

Faculty members can filter event registrations based on the type of event. This allows them to view participants and manage event logistics.

3.1.6 Security and Session Handling

The application uses Flask sessions to manage login states. All protected routes check for valid session information to prevent unauthorized access.

3.1.7 Testing and Deployment

The system was tested across all user flows—login, event registration, announcement posting, and logout—to ensure end-to-end functionality. Flask's built-in development server was used for testing and debugging.

This modular approach enables easy scaling, maintainability, and adaptability of the system for real institutional use.

Chapter 4

Results and Analysis

The Institute Management System was successfully implemented, providing a centralized platform for managing students, faculty, courses, and event registrations. The system ensures smooth navigation through role-based logins and offers different dashboards and features for Admin, Student, and Faculty users. The user interface is intuitive, responsive, and accessible across devices.

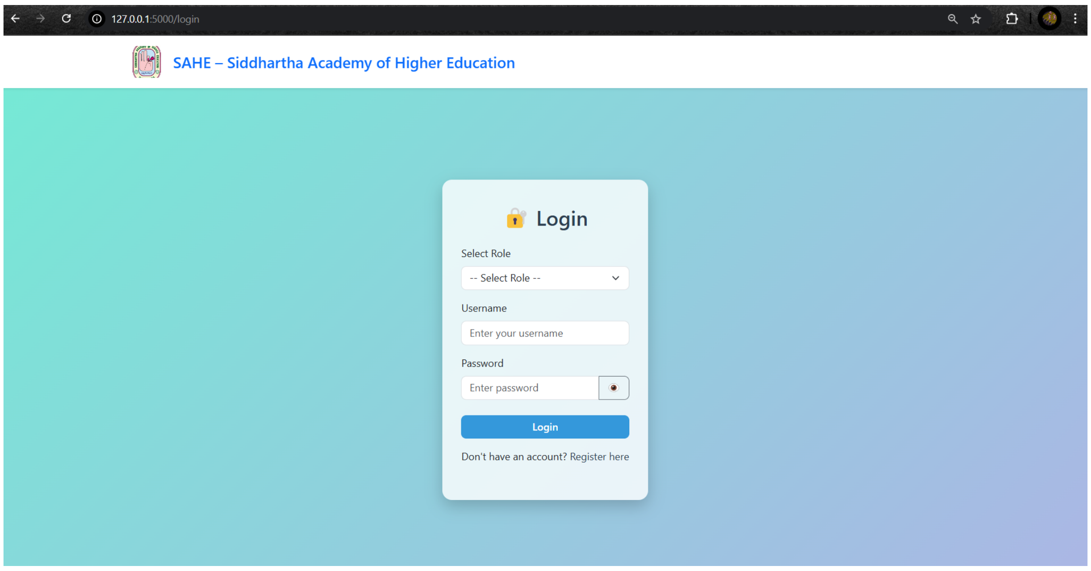
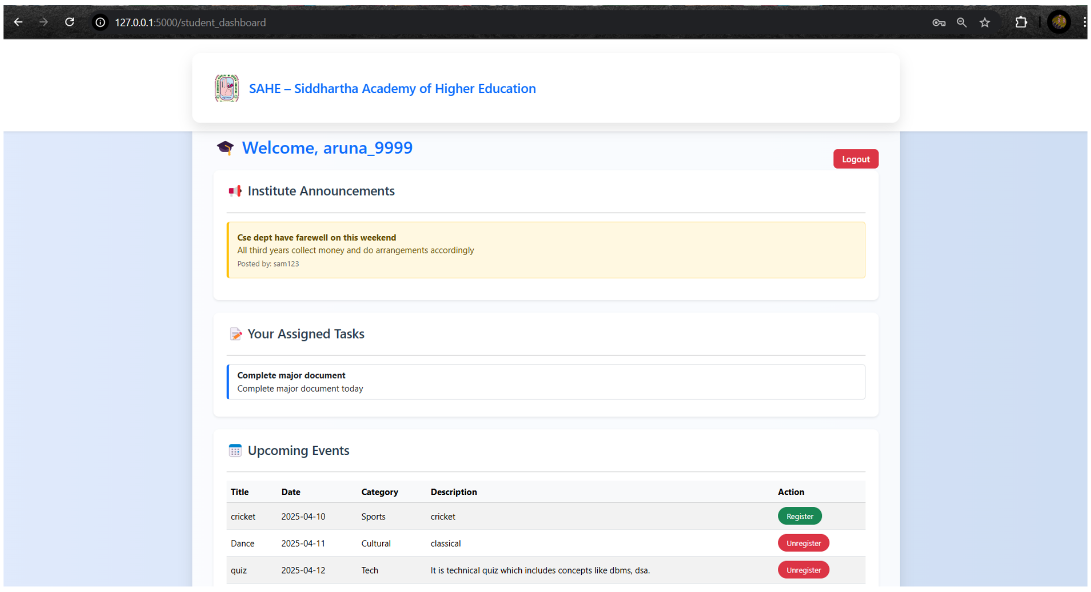


Figure 4.1: Login Page

Figure 4.1 shows the login page for the system. It includes fields for email and password with a ”Login” button. Upon successful login, the user is redirected to their respective dashboard based on their role (Admin, Faculty, or Student).


127.0.0.1:5000/student_dashboard

SAHE – Siddhartha Academy of Higher Education

Welcome, aruna_9999

Institute Announcements

Cse dept have farewell on this weekend
All third years collect money and do arrangements accordingly
Posted by: sam123

Your Assigned Tasks

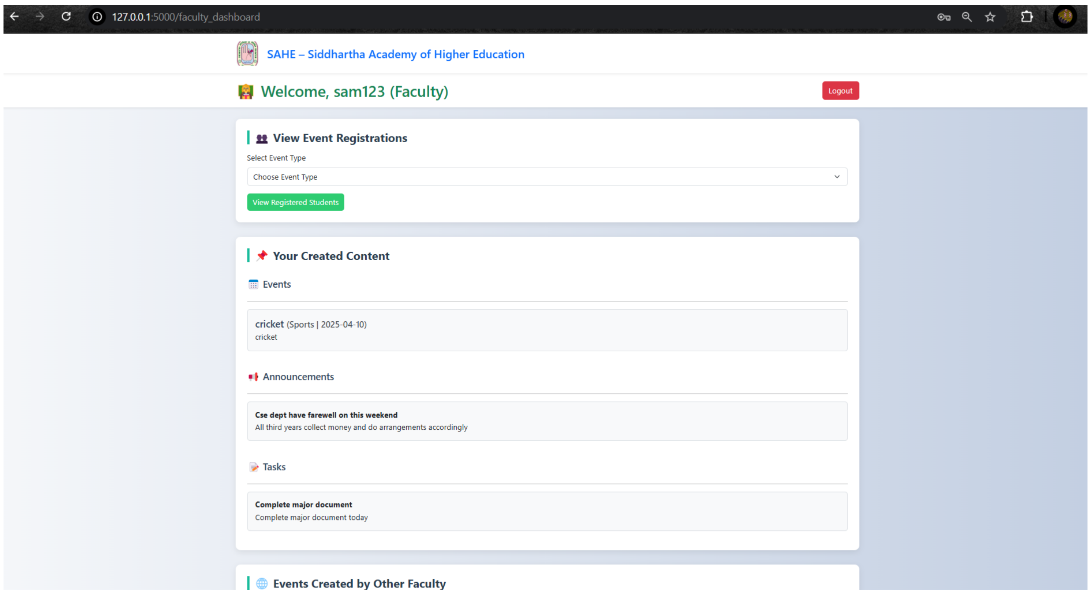
Complete major document
Complete major document today

Upcoming Events

Title	Date	Category	Description	Action
cricket	2025-04-10	Sports	cricket	Register
Dance	2025-04-11	Cultural	classical	Unregister
quiz	2025-04-12	Tech	It is technical quiz which includes concepts like dbms, dsa.	Unregister

Figure 4.2: Student Dashboard

Figure 4.2 shows the dashboard view for students. It displays modules such as profile, Announcements, Tasks and a list of available events for registration. Students can also access their registered event history from this panel.


127.0.0.1:5000/faculty_dashboard

SAHE – Siddhartha Academy of Higher Education

Welcome, sam123 (Faculty)

View Event Registrations

Select Event Type: Choose Event Type

Your Created Content

Events

cricket (Sports | 2025-04-10)
cricket

Announcements

Cse dept have farewell on this weekend
All third years collect money and do arrangements accordingly

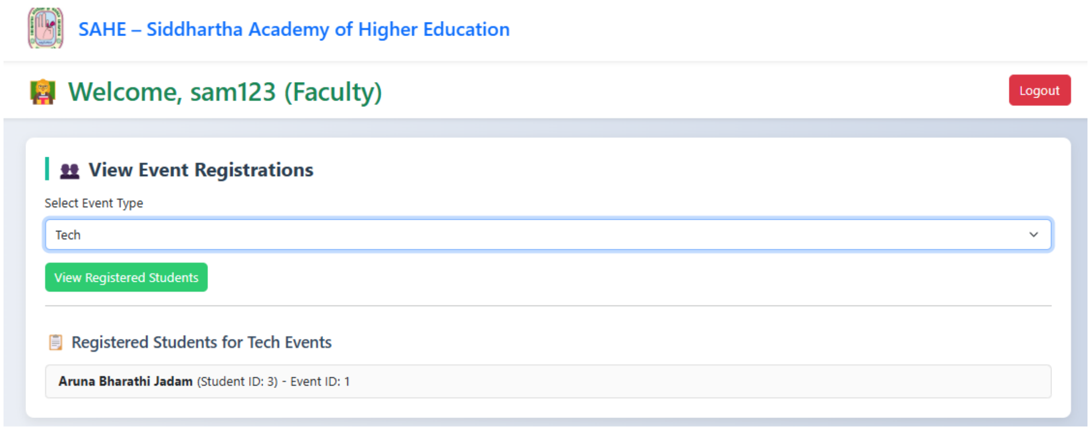
Tasks

Complete major document
Complete major document today

Events Created by Other Faculty

Figure 4.3: Faculty Dashboard

Figure 4.3 shows the faculty dashboard. It allows faculty to view their events, create events, announcements and tasks.


SAHE – Siddhartha Academy of Higher Education

Welcome, sam123 (Faculty)

Logout

View Event Registrations

Select Event Type

Tech

View Registered Students

Registered Students for Tech Events

Aruna Bharathi Jadam (Student ID: 3) - Event ID: 1

Figure 4.4: Event Registration Page

Figure 4.2 illustrates the student's event registration page. It displays a list of upcoming events created by the admin. Students can register for events by clicking the "Register" button. Once registered, their names are stored and associated with the selected event.

Figure 4.4 illustrates the Faculty members can access a dedicated dashboard where they can select any event from the list and view the detailed information of all students who have registered for that particular event. This functionality helps in efficient event coordination and participant management.

Overall, the Institute Management System fulfills the objective of streamlining academic and event management tasks by offering a user-friendly interface with role-based access and real-time operations.

Overall, the Institute Management System fulfills the objective of streamlining academic and event management tasks by offering a user-friendly interface with role-based access and real-time operations.

Chapter 5
Conclusion and Future Work

The Institute Management System was successfully developed to simplify and digitize event management and communication between students and faculty within an academic environment. This web-based application features a secure login system, dedicated dashboards for students and faculty, event creation by faculty, and seamless student registration for upcoming events. The system dynamically displays registered participants for each event in the faculty dashboard, enhancing transparency and real-time monitoring. The user interface is clean and responsive, ensuring a smooth experience across all roles.

Future enhancements may include integrating automated email or SMS notifications for event reminders, adding a calendar view for tracking upcoming events, implementing a feedback and rating system for event evaluation, enabling real-time editing or cancellation of events, and introducing analytics dashboards for visualizing participation metrics. Furthermore, incorporating AI-driven event recommendations based on student interests and previous participation, along with enhanced security features like multi-factor authentication, can significantly improve user experience and overall functionality. This project sets a strong foundation for a comprehensive digital solution in academic institutions.

REFERENCES

1. **Fathi, M., Kashani, M. H., Jameii, S. M., Mahdipour, E.** (2021). *Design Architecture Of An Integrated Student Activities Management System For Higher Education.* Archives of Computational Methods in Engineering, 29, 1247–1275. https://doi.org/10.1007/s11831-021-09616-4.

2. **Divya, S.** (2024). *Activity management as a Web service.* Knowledge and Information Systems.
https://doi.org/10.1007/s10115-024-02332-y.

3. **Salman, A. G., Kanigoro, B., Heryadi, Y.** (2015). *A comprehensive design model for integrating business processes in web applications.* 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 281–285. IEEE. https://doi.org/10.1109/ICACSIS.2015.7415153.

4. **Kunjumon, C., Nair, S. S., Rajan, S. D., Suresh, P., Preetha, S. L.** (2018). *A survey on web modeling approaches for ubiquitous web applications.* 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), 262–264. IEEE. https://doi.org/10.1109/ICEDSS.2018.8544320.

5. **Kareem, F. Q., Abdulazeez, A. M., Hasan, D. A.** (2021). *University libraries: step towards a web based knowledge management system.* Asian Journal of Research in Computer Science, 9(3), 13–24. https://doi.org/10.9734/ajrcos/2021/v9i330219.

6. **Taivalsaari, A., Mikkonen, T., Pautasso, C., Systä, K.** (2021). *Full Stack Is Not What It Used to Be.* In: Web Engineering. ICWE 2021. Lecture Notes in Computer Science, vol 12706. Springer, Cham. https://doi.org/10.1007/978-3-030-74296-6_28.

7. **Northwood, C.** (2018). *The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer.* Apress. https:

//doi.org/10.1007/978-1-4842-4152-3.

8. **Eck, F.** (2019). *Full Stack Web Development with Jakarta EE and Vue.js.* Apress. https://doi.org/10.1007/978-1-4842-6342-6.

9. **Walsh, B.** (2023). *AI Full Stack: Application Development.* In: Productionizing AI. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-8817-7_7.

10. **Kumar, S., Gupta, S.** (2022). *E-Commerce Web Portal Using Full-Stack Open-Source Technologies.* In: Advances in Computing and Intelligent Systems. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-19-3575-6_2.