IMPLEMENT ROLE BASED ACCESS CONTROL (RBAC) FOR A SAAS APPLICATION

A NAAN MUDHALVAN REPORT

SUBMITTED BY

KARUPPAIAH M   912421106007

MANJULA DEVI V   912421106008

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SHANMUGANATHAN ENGINEERING COLLEGE

ARASAMPATTI, PUDUKKOTTAI – 622 507

YEAR & SEMESTER   : IV & VII

SUBJECT CODE   : NM1050

COURSE NAME   : SOFTWARE AS A SERVICE (SaaS)

ANNA UNIVERSITY::CHENNAI 600 025

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this Naan mudhalvan **"IMPLEMENT ROLE BASED ACCESS CONTROL (RBAC) FOR A SAAS APPLICATION"** is the bonafide work of **"KARUPPAIAH M(912421106007), MANJULA DEVI V (912421106008)"** who carried out the project work under my guidance.

<table>
<tr><td>

SIGNATURE

Mrs. D. LATHA M.E.,

NAAN MUDHALVAN COORDINATOR

ASSISTANT PROFESSOR,

Department of Electronics &

Communication Engineering,

Shanmuganathan Engineering College,

Arasampatti – 622 507

</td><td>

SIGNATURE

Dr. A.MUTHU MANICKAM M.E., Ph.D.,

HEAD OF THE DEPARTMENT

ASSISTANT PROFESSOR,

 Department of Electronics &

 Communication Engineering,

 Shanmuganathan Engineering College,

 Arasampatti – 622 507

</td></tr>
</table>

Submitted for the Project viva-voice on   _____

INTERNAL EXAMINER   EXTERNAL EXAMINER

ACKNOWLEDGMENT

At this pleasing moment having successfully completed our project report, we wish to convey our sincere thanks to our beloved chairperson **Mrs. PICHAPPA VALLIAMMAL,** correspondent **Dr. P. MANIKANDAN B.E**, director(Academic) Shri **M.SHANMUGANATHAN**, director(Administration) Shri **M. PICHAPPA** and honourable secretary **Mr. M. VISWANATHAN** for their extensive support.

I thankful to our principal **Dr. KL. MUTHURAMU M.E(W.R)., M.E(S.E)., Ph.D., FIE., M.I.S.T.E.,** Shanmuganathan engineering college, for providing the opportunity to conduct our project.

I extend our gratitude **Dr. A. MUTHU MANICKAM M.E., Ph.D.,** the head of the department of Electronics and communication engineering for providing a valuable suggestion and supports given through the study.

Gratitude never fails. We are grateful to our dynamic and effective internal guide **Asst. Prof. Mrs. D. LATHA M.E.,** for his valuable innovative suggestion, constructive interactions, constant encouragement and valuable helps that have been provided us throughout the project.

I also express my heartfelt thanks to all other staff members of Electronics communication engineering department for their support. Above all, we thank our parents, for affording us the valuable education till now.

ABSTRACT

Implementing Role-Based Access Control (RBAC) in a SaaS application involves systematically restricting system access based on user roles. This theoretical approach outlines the principles, design, and methodologies necessary to effectively implement RBAC in a SaaS environment. RBAC is a method for enforcing security policies by assigning permissions to roles rather than to individual users. In a SaaS application, RBAC helps in managing user permissions at scale, supporting security, compliance, and operational efficiency. The goal is to ensure that users only have access to data and actions necessary for their role, minimizing potential misuse or accidental changes to sensitive information. Implementing Role-Based Access Control (RBAC) in a SaaS application establishes a structured security framework that governs user access based on defined roles and associated permissions. This strategy enables SaaS applications to restrict access to sensitive data and functionality, ensuring users only perform actions permitted by their roles.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 SAAS

In today's digital landscape, Software as a Service (SaaS) applications have become the backbone of many businesses, offering scalable, flexible, and cost-effective solutions to a wide range of operational needs. The development of a SaaS application involves creating a software product that is hosted and accessed online, typically through a subscription model. These applications can range from project management tools to customer relationship management (CRM) systems, to complex enterprise resource planning (ERP) solutions.The end-to-end development of a SaaS application refers to the entire lifecycle of the software, from initial ideation and design to deployment, scaling, and ongoing maintenance. This process encompasses several stages, each crucial to building a high-performance, secure, and user-friendly application. Successful SaaS development requires careful planning, strong technical expertise, and ongoing support.

The rise of Software as a Service (SaaS) has transformed the way businesses and individuals access software solutions. Unlike traditional software models that require on-premise installation, SaaS applications are hosted in the cloud and accessed over the internet. These applications are typically subscription-based and offer a variety of services ranging from project management and CRM systems to complex enterprise solutions.

End-to-end development of a SaaS application refers to the complete process of building and delivering a SaaS product—from initial planning and ideation through to ongoing maintenance and scaling. This comprehensive development process encompasses multiple stages, including design, coding, deployment, and ongoing updates, all of which must be carefully orchestrated to create a secure, scalable, and user-friendly application.

In this context, the end-to-end development lifecycle ensures that every aspect of the application is built to meet the needs of users, perform efficiently, and remain adaptable as the user base grows or business requirements evolve.

1.2 EXISTING SYSTEM

- **On-Premises Hosting**: Traditionally, applications were hosted on dedicated servers, which required significant up-front costs and maintenance for hardware and software.
- **Monolithic Architecture**: Applications were built as a single unit, which made it challenging to scale and update specific parts of the system independently.
- **Waterfall Development Model**: Development followed a linear, phase-by-phase approach with limited flexibility to adapt to changes, often resulting in longer delivery times.
- **Manual Deployment and Updates**: Deployment was often manual, which introduced delays and increased the likelihood of errors.

1.3 PROPOSED SYSTEM

- **Cloud-Based Hosting**: SaaS applications are hosted in the cloud (e.g., AWS, Azure, Google Cloud), allowing on-demand resource allocation, high availability, and lower upfront costs.
- **Microservices or Serverless Architecture**: Applications are broken down into smaller, independent services or functions. This modular approach enables flexible scaling and easier updates for specific features without affecting the entire application.
- **Agile Development and DevOps**: Agile methodologies and DevOps practices foster collaboration, adaptability, and incremental delivery, which speeds up development cycles and allows for rapid iteration based on user feedback.
- **Automated CI/CD Pipelines**: Continuous integration and deployment pipelines streamline testing and deployment, reducing the risk of errors and enabling frequent updates to keep the application current and secure.
- **Auto-Scaling and Load Balancing**: Cloud providers offer auto-scaling and load balancing to dynamically adjust resources based on demand, optimizing performance and cost-efficiency.
- **Comprehensive Monitoring and Analytics**: Advanced monitoring tools and analytics allow real-time visibility into application performance, user behavior, and potential issues, enabling proactive maintenance and faster troubleshooting.
- **Enhanced Security and Compliance**: Modern SaaS systems incorporate robust security practices (e.g., encryption, identity management, regular audits) and are often designed with compliance frameworks in mind (e.g., GDPR, HIPAA), addressing the higher security expectations of cloud applications

CHAPTER 2

FRONTEND DEVELOPMENT

INTRODUCTION

Frontend development is the art and science of creating the user interface (UI) and user experience (UX) of a website or web application. It's the part of a website that users directly interact with, including the layout, design, and functionality.

Core Technologies

To build a dynamic and engaging frontend, developers primarily rely on three core technologies:

1. **HTML (HyperText Markup Language):**
   o The structural backbone of web pages.
   o Defines the content and layout using elements like headings, paragraphs, images, links, and forms.
2. **CSS (Cascading Style Sheets):**
   o Styles the HTML elements.
   o Controls the visual appearance, including colors, fonts, spacing, and layout.
3. **JavaScript:**
   o Adds interactivity and dynamic behavior.
   o Enables features like animations, form validation, and real-time updates.

Responsibilities of a Frontend Developer

- **Design and Layout:** Creating visually appealing and user-friendly layouts.
- **User Experience (UX):** Designing intuitive and efficient user interactions.
- **Responsiveness:** Ensuring the website adapts to different screen sizes and devices.
- **Performance Optimization:** Optimizing website speed and performance.
- **Accessibility:** Making the website accessible to users with disabilities.
- **Cross-Browser Compatibility:** Ensuring the website works correctly across different browsers.

CHAPTER 3

HYPERTEXT MARKUP LANGUAGE

INTRODUCTION

HTML, or HyperText Markup Language, is the standard markup language used to create web pages. It's a combination of Hypertext, which defines the link between web pages, and Markup language, which is used to define the text document within tags to structure web pages. This language is used to annotate text so that machines can understand and manipulate it accordingly. HTML is human-readable and uses tags to define what manipulation has to be done on the text.

It uses HTML tags and attributes to describe the structure and formatting of a web page. HTML consists of various elements, that are responsible for telling search engines how to display page content. For example, headings, lists, images, links, and more.

Features of HTML

1. It is easy to learn and easy to use.
2. It is platform-independent.
3. Images, videos, and audio can be added to a web page.
4. Hypertext can be added to the text.
5. It is a markup language.

3.1 HTML ELEMENTS AND TAGS

HTML uses predefined tags and elements that instruct the browser on how to display the content. HTML elements include an opening tag, some content, and a closing tag. It's important to remember to include closing tags. If omitted, the browser applies the effect of the opening tag until the end of the page.

HTML elements are the fundamental components used to structure and organize content on a web page. They are defined by tags, which are enclosed in angle brackets (<>).

Figure 3.1 HTML Element

This section will dive into the basic structure of an HTML page, which includes essential building-block elements like doctype declaration, HTML, head, title, and body elements.

3.2 HTML PAGE STRUCTURE

The basic structure of an HTML page is shown below. It contains the essential building-block elements (i.e. doctype declaration, HTML, head, title, and body elements) upon which all web pages are created.
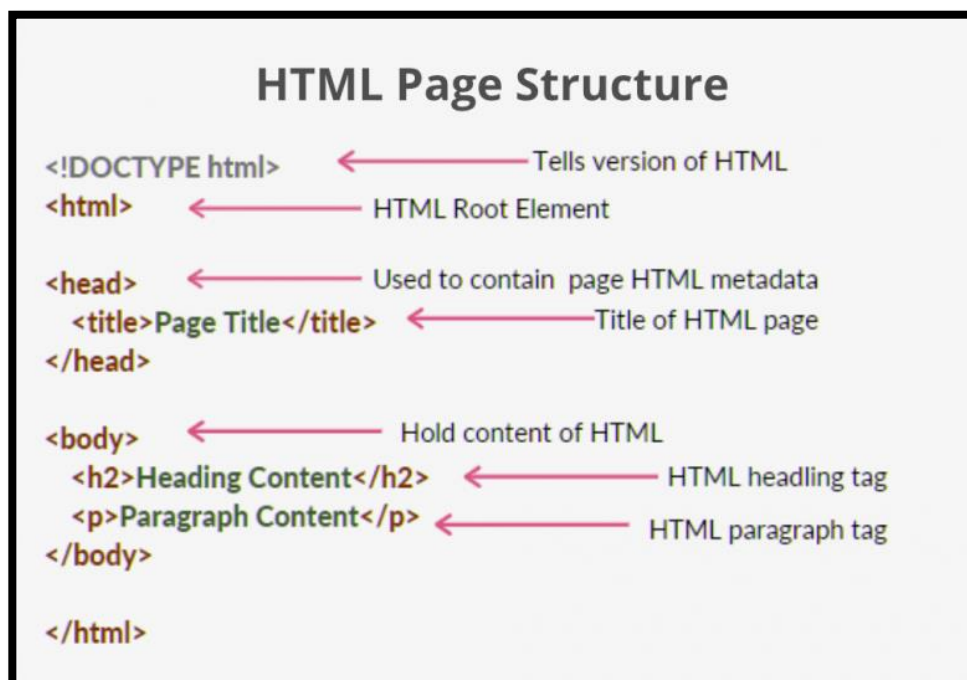
3.2 HTML Page Structure

<!DOCTYPE html> – This is the document type declaration (not technically a tag). It declares a document as being an HTML document. The doctype declaration is not case-sensitive.

<html> – This is called the HTML root element. All other elements are contained within it.

<head> – The head tag contains the "behind the scenes" elements for a webpage. Elements within the head aren't visible on the front end of a webpage. HTML elements used inside the <head> element include:

<style> – This HTML tag allows us to insert styling into our web pages and make them appealing to look at with the help of CSS.

<title> – The title is what is displayed on the top of your browser when you visit a website and contains the title of the webpage that you are viewing.

<base> – It specifies the base URL for all relative URL's in a document.

<noscript> – Defines a section of HTML that is inserted when the scripting has been turned off in the user's browser.

<script> – This tag is used to add functionality to the website with the help of JavaScript.
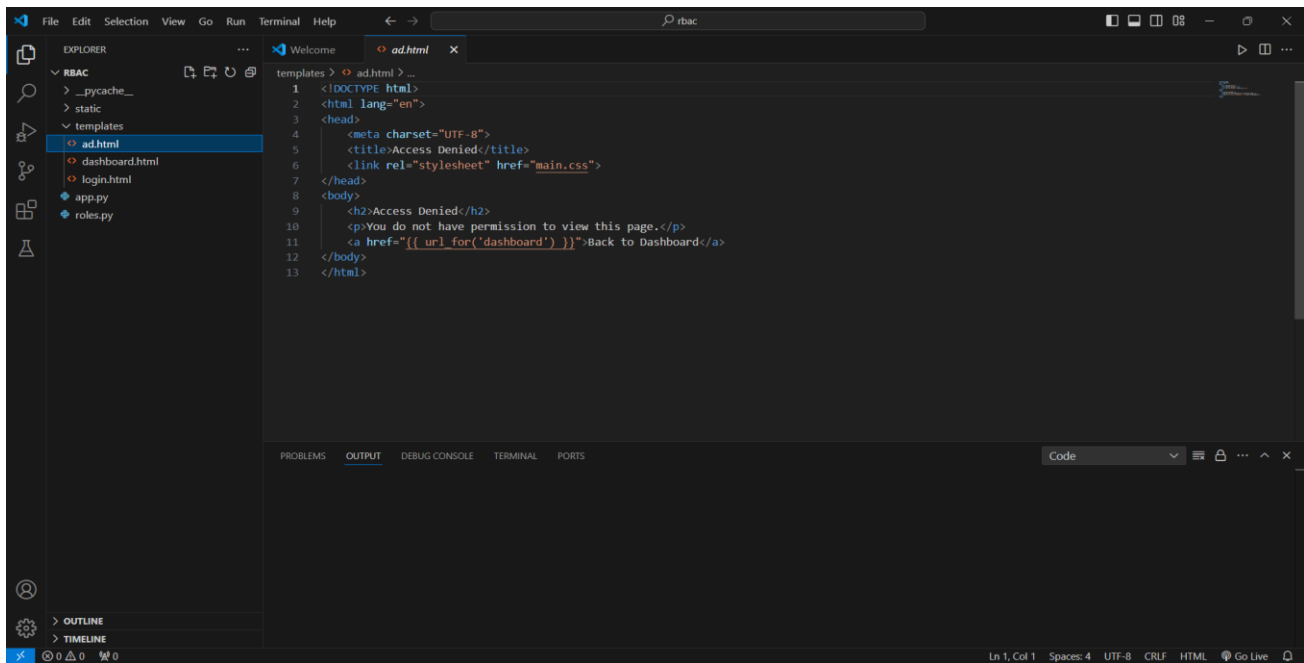
<meta> – This tag encloses the metadata of the website that must be loaded every time the website is visited. For eg:- the metadata charset allows you to use the standard UTF-8 encoding on your website. This in turn allows the users to view your webpage in the language of their choice. It is a self-closing tag.

<link> – The 'link' tag is used to tie together HTML, CSS, and JavaScript. It is self-closing.

<body> – The body tag is used to enclose all the visible content of a webpage. In other words, the body content is what the browser will show on the front end.
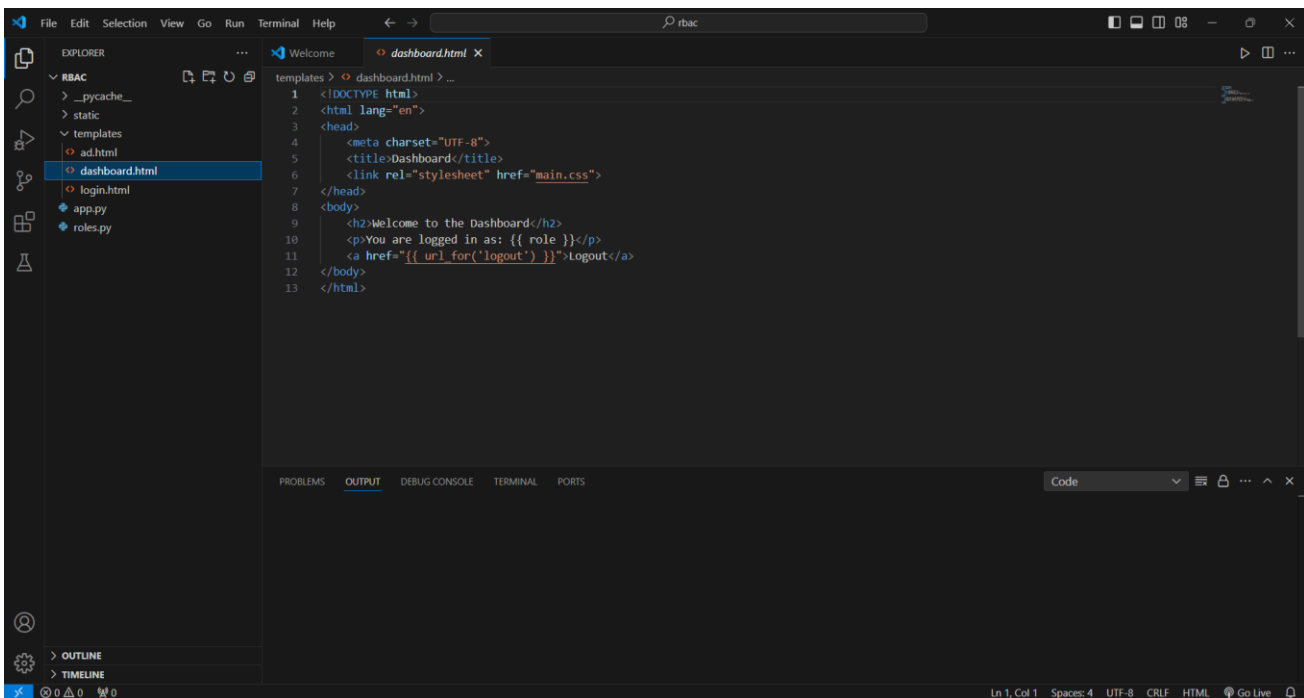
An HTML document can be created using an HTML text editor. Save the text file using the ".html" or ".htm" extension. Once saved as an HTML document, the file can be opened as a webpage in the browser.

3.3 HTML SOURCE CODE

CHAPTER 4

CASCADING STYLE SHEETS

INTRODUCTION

CSS (Cascading Style Sheets) is a language designed to simplify the process of making web pages presentable. It allows you to apply styles to HTML documents, describing how a webpage should look by prescribing colors, fonts, spacing, and positioning. CSS provides developers and designers with powerful control over the presentation of HTML elements.

HTML uses tags and CSS uses rule sets. CSS styles are applied to the HTML element using selectors. CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

- **Saves Time:** Write CSS once and reuse it across multiple HTML pages.
- **Easy Maintenance:** Change the style globally with a single modification.
- **Search Engine Friendly:** Clean coding technique that improves readability for search engines.

CSS Versions Release year

1. CSS1 – Released in 1996

   Introduced basic styling features like font properties, text alignment, spacing, and margins.

2. CSS2 – Released in 1998

   Brought improvements, including positioning, z-index, media types, and more selector options.

3. CSS3 – Introduced as a series of modules starting in 1999, with gradual adoption through the 2000s and 2010s

4. CSS4 – Never formally released as a single version, but starting around 2015, new modules aligned with what people referred to as "CSS4"

4.1 CSS SYNTAX

CSS consists of style rules that are interpreted by the browser and applied to the corresponding elements. A style rule set includes a selector and a declaration block.

Selector: Targets specific HTML elements to apply styles.

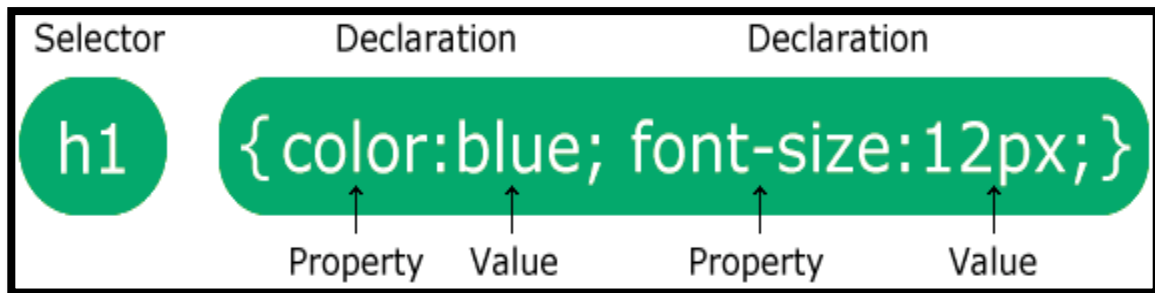**Declaration:** Combination of a property and its corresponding value.


Figure 4.1 CSS Syntax

- The selector points to the HTML element you want to style.

- The declaration block contains one or more declarations separated by semicolons.

- Each declaration includes a CSS property name and a value, separated by a colon.

- Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

4.2 CSS Modules

CSS Modules are having old CSS specifications as well as extension features.

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D and 3D Transformations
- Animations

4.3 CSS SOURCE CODE
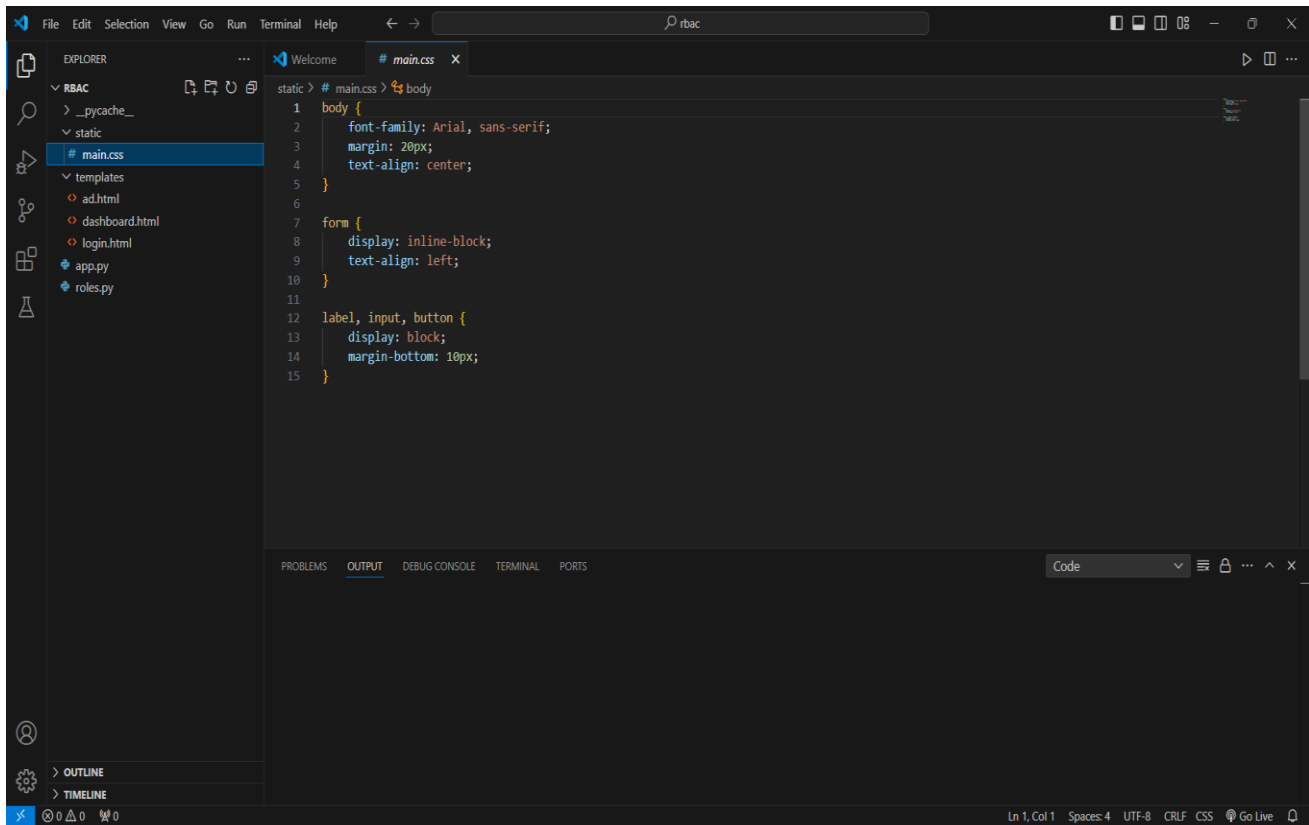
CHAPTER 5

JAVASCRIPT

INTRODUCTION

➢ JavaScript is a versatile, high-level programming language used to add interactivity, control multimedia, animate images, and much more on web pages. Initially created for front-end development, JavaScript has since expanded to server-side programming with tools like Node.js, making it a full-stack language.

➢ It is a crucial tool for creating interactive and dynamic web pages, web applications, and mobile applications.

Basic JavaScript Syntax

1. Variables: let, const, var

2. Data types: numbers, strings, booleans, arrays, objects

3. Operators: arithmetic, comparison, logical

4. Control structures: if/else, switch, loops (for, while, do-while)

5. Functions: declarations, expressions, arrow functions

➢ Client-side: It supplies objects to control a browser and its Document Object Model (DOM). Like if client-side extensions allow an application to place elements on an HTML form and respond to user events such as mouse clicks, form input, and page navigation. Useful libraries for the client side are AngularJS, ReactJS, VueJS, and so many others.

➢ Server-side: It supplies objects relevant to running JavaScript on a server. For if the server-side extensions allow an application to communicate with a database, and provide continuity of information from one invocation to another of the application, or perform file manipulations on a server. The useful framework which is the most famous these days is node.js.

> ➢ Imperative language – In this type of language we are mostly concerned about how it is to be done. It simply controls the flow of computation. The procedural programming approach, object, oriented approach comes under this as async await we are thinking about what is to be done further after the async call.

> ➢ Declarative programming – In this type of language we are concerned about how it is to be done, basically here logical computation requires.

5.1 JAVASCRIPT SYNTAX

1. Variables: let, const, var

2. Operators: arithmetic (+, -, *, /), comparison (==, !=, ===, !==), logical (&&, ||, !)

3. Control structures:

   Conditional statements: if, else, switch

   Loops: for, while, do-while

4. Functions:

   Function declarations: function name() { ... }

   Function expressions: let name = function() { ... }

   Arrow functions: let name = () => { ... }

5. Object-oriented programming:

   Classes: class Name { ... }

   Inheritance: extends

   Methods: function name() { ... }

5.2 TYPES OF JAVASCRIPT
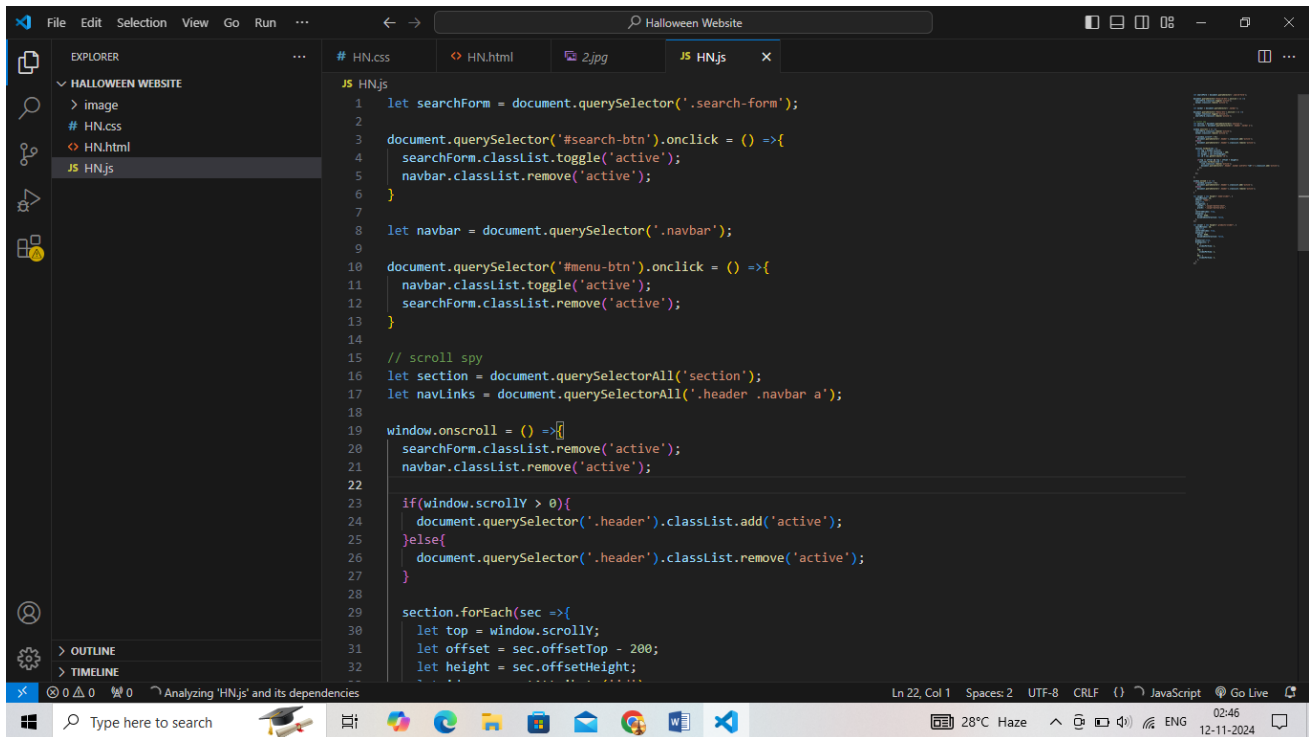
1. Client-side JavaScript: Runs on web browsers (e.g., Chrome, Firefox)

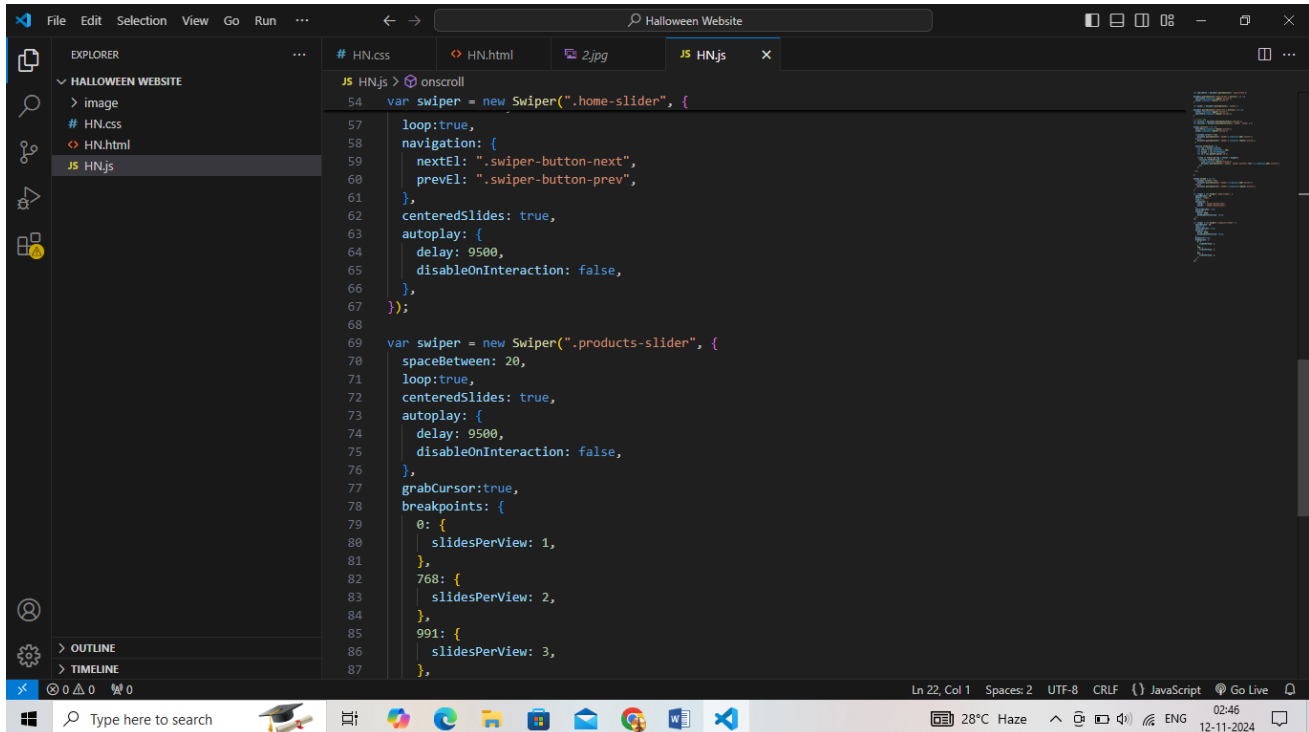2. Server-side JavaScript: Runs on servers using Node.js

3. Desktop JavaScript: Runs on desktop applications using Electron or Node-Webkit

4. Mobile JavaScript: Runs on mobile apps using React Native or Angular Mobile

5.3 JAVASCRIPT SOURCE CODE

CHAPTER 6

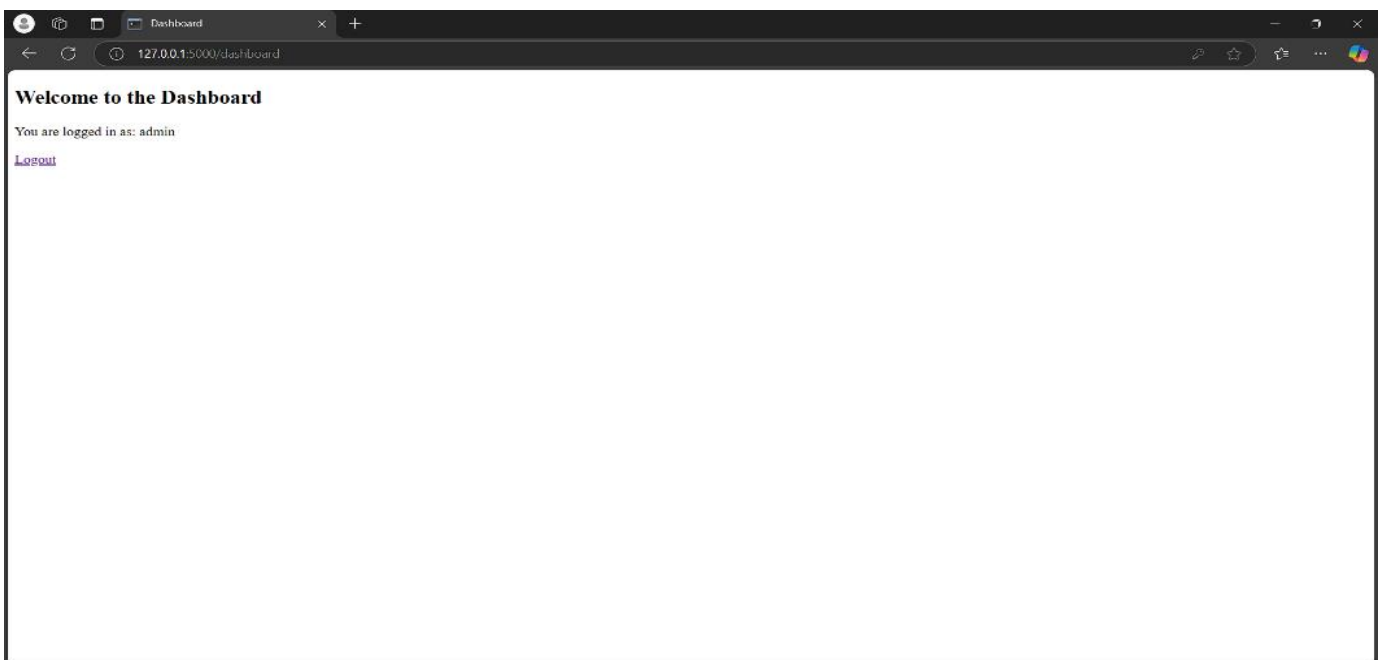# IMPLEMENT ROLL BASED ACCESS CONTROL FOR A SAAS APPLICATION

6.1 LOGIN
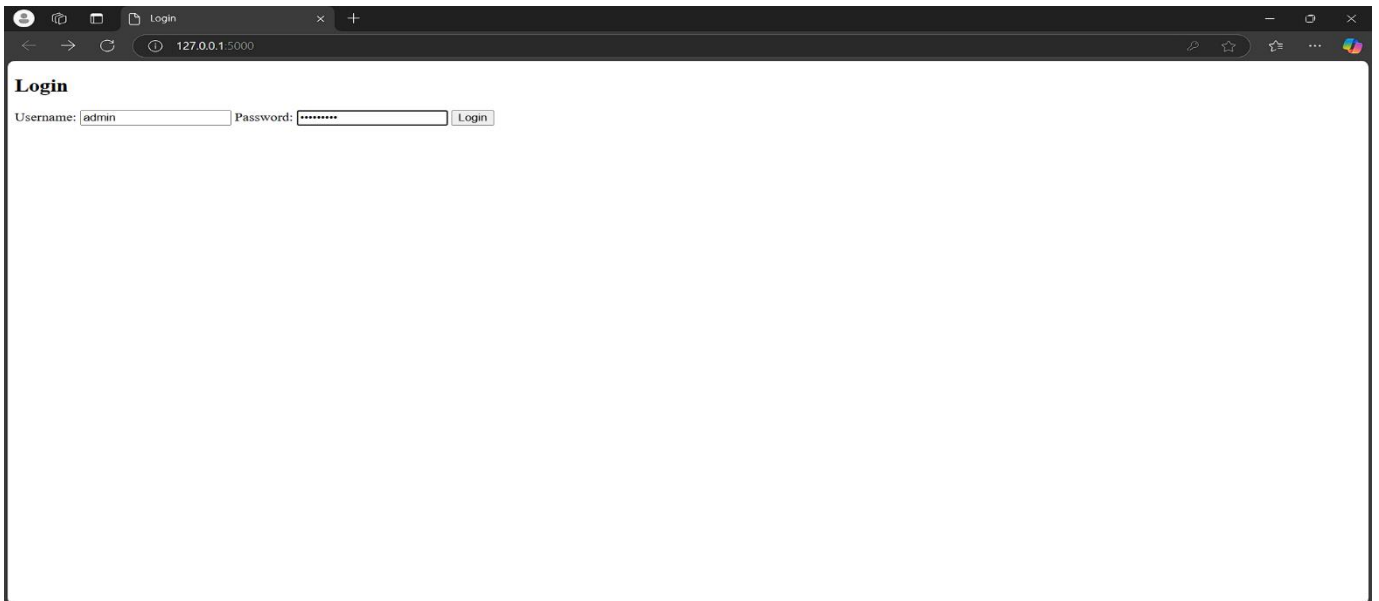
6.2 LOGIN AS ADMIN

6.3 LOGIN
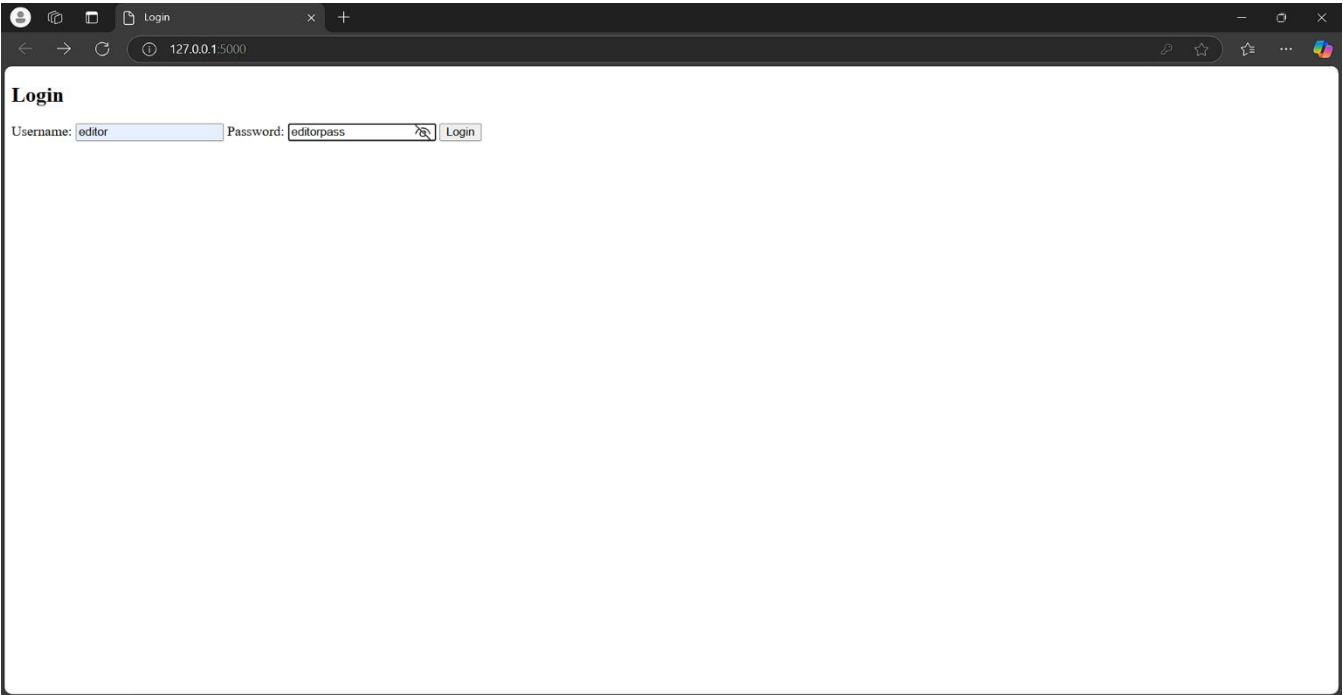
6.4 LOGIN AS EDITOR
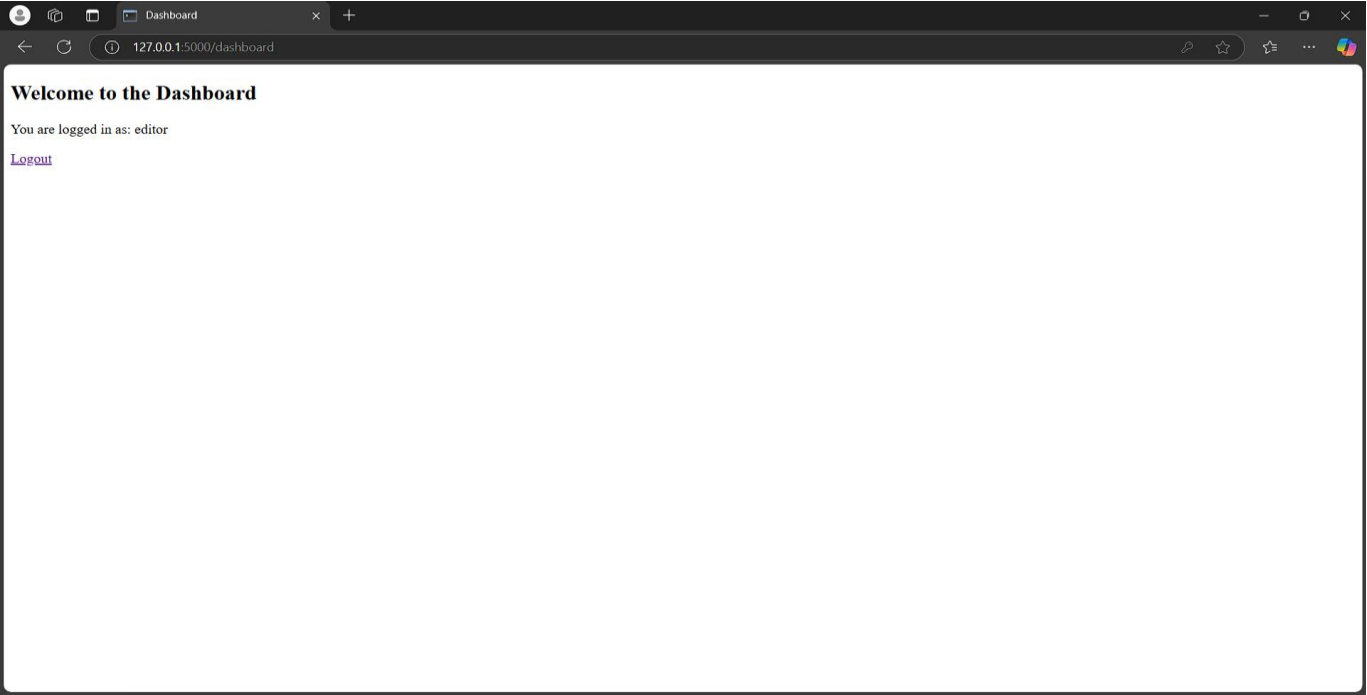
CHAPTER 7

SOFTWARE DESCRIPTION

7.1 VISUAL STUDIO CODE

Visual Studio Code (VS Code) is a open-source, lightweight, and flexible code editor developed by Microsoft. It's designed for developers to write, debug, and test their code in various programming languages.VS Code is a free code editor developed by Microsoft. It's known for being lightweight yet powerful, making it a popular choice among developers for various programming languages and environments.

Figure 7.1 VS CODE

1. Lightweight and Cross-Platform

VS Code is available for Windows, macOS, and Linux, making it a versatile choice across different operating systems.

It's designed to be lightweight and efficient, with a fast startup time and minimal system resource usage.

2. Intuitive User Interface

Editor and Sidebar: The main editor window is accompanied by a sidebar for file management, version control, and extensions.

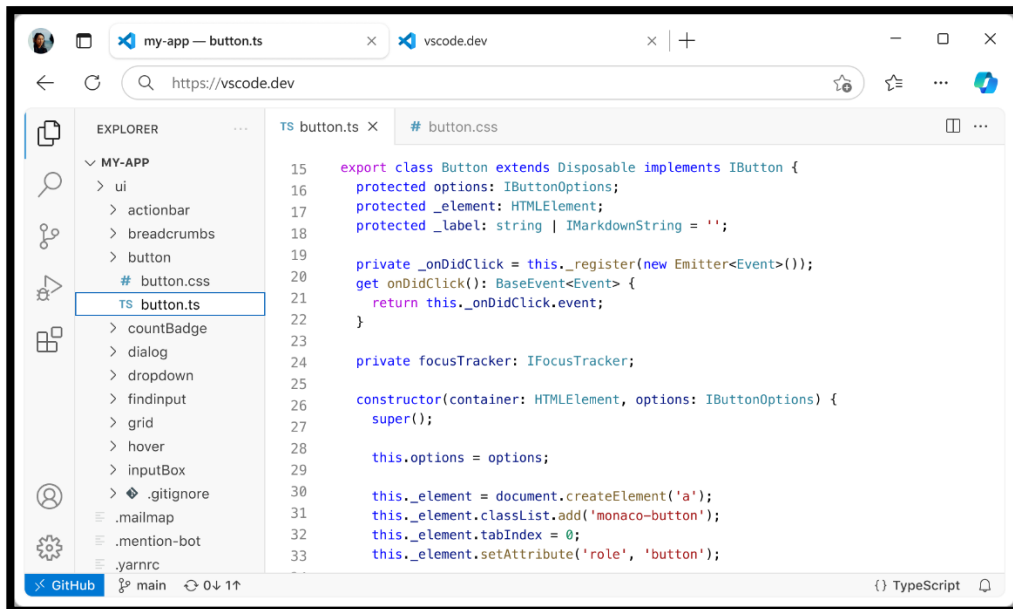Integrated Terminal: VS Code has an integrated terminal where you can run commands, scripts, or interact with your project's environment directly within the editor.

3. Extensive Language Support

VS Code natively supports JavaScript, TypeScript, HTML, and CSS, with built-in features like syntax highlighting, autocompletion, and error checking.

For other languages, VS Code's Extensions Marketplace offers support for Python, Java, C++, PHP, Go, Ruby, and many more.

4. Extensions and Customization

Extensions Marketplace: VS Code has a rich library of extensions that can add support for new languages, integrate with development frameworks, or provide additional features like linting, formatting, and debugging.

Themes and Customization: You can customize the look and feel of VS Code with themes, as well as adjust settings like font size, colors, and editor behaviors.

5. Code Intelligence and Assistance

IntelliSense: This feature provides smart code completion, parameter info, quick info, and member lists for a wide variety of languages.

Linting and Error Detection: VS Code includes built-in linting and error detection that help catch mistakes in real time.

6. Debugging Tools

VS Code has an integrated debugger that supports Node.js, JavaScript, Python, and other environments.

You can set breakpoints, watch variables, and step through code to troubleshoot and understand how your program runs.

CHAPTER 8

ADVANTAGES AND APPLICATIONS

ADVANTAGES

1. Improved data compatibility

2. Cost savings

3. Scalability

4. Ease of management

5. Improved customer experience

APPLICATIONS

1. Enterprise Resource Planning (ERP) Software

2. Customer Relationship Management (CRM)

3. Project Management Software

4. E-commerce Platforms

5. Healthcare IT Solutions

6. Financial Software

7. Education Technology (EdTech) Platforms

8. Marketing Automation

CHAPTER 9

CONCLUSION

In conclusion, Role-Based Access Control (RBAC) offers a robust, efficient, and scalable security framework for managing user access in SaaS applications. By assigning permissions to roles based on job functions and then assigning those roles to users, RBAC reduces administrative complexity, enhances security, and ensures compliance with regulatory standards. This model adheres to the principle of least privilege, giving users only the access necessary to perform their responsibilities and thereby minimizing the risk of accidental or malicious misuse of data and system functionality. The structured nature of RBAC allows SaaS applications to grow in complexity without compromising access control. This adaptability is particularly valuable in multi-tenant environments, where different clients may have distinct access requirements. By centralizing permissions within roles, administrators can quickly adjust access policies to reflect organizational or application changes without directly modifying individual user settings. This modularity not only simplifies onboarding and user management but also allows for more efficient audits, as permissions can be reviewed and updated at the role level, ensuring compliance across all users. A well-implemented RBAC system in a SaaS application also includes comprehensive logging and monitoring, which are essential for security and compliance. Logging role assignments, permission changes, and access events enables SaaS providers to maintain an auditable trail of access-related actions, facilitating internal reviews, regulatory audits, and proactive security monitoring.

CHAPTER 10

REFERENCES

1. P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft)", *Tech. Rep.*, 2011, [online] Available: www.nist.org.

2. Aniruddha S Rumale and DN Chaudhari, "Cloud Computing: Infrastructure as a Service", *International Journal of Inventive Engineering and Sciences (IJIES)*, vol. 1, no. 3, pp. 1-7, February 2013, ISSN 23199598.

3. Steven Muthula, *Understanding the Cloud Computing Stack: SaaS PaaS IaaS and Big Data*, Nov 2015, [online] Available: https://www.linkedin.com/pulse/understanding-cloud-computingstack-saas-paas-iaas-big-steven-murhula.

4. H. Beard, "Cloud Computing Best Practices: For Managing and Measuring Processes for On-Demand Computing", *Applications and Data centers in the Cloud with SLAs*, 2011.

5. AS Rumale and DN Chaudhari, "Post Login Authentication of User for Service Usage Authorization in Cloud Computing", *IEEE conference*.

6. Ivanka Menken and Gerard Blokdijk, Cloud Computing Certification Kit Specialist: Software as a Service & Web Applications, The Art of Service, 2010.

7. R. Buyya, S.K. Garg and R.N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges Architecture and Solutions", *IEEE International Conference on Cloud and Service Computing*, pp. 1-10, 2011.

8. B. Daley and A. Rudolph, "Accurately monitoring cloud slas" in cloudbook Journal, 3000 San Hill Road Suite 3–100, Menlo Park, CA 94402:Active Book Press, vol. 1, no. 4, pp. 29-31, 2010.

9. AS Rumale and DN Chaudhari, "Cloud Computing: Service Level Agreements(SLA)", *INTERNATIONAL JOURNAL OF SCIENTIFIC & ENGINEERING RESEARCH*, vol. 4, no. 9, pp. 1-5, SEPTEMBER 2013, ISSN 2229-5518.

10. Michael P. McGrath, Understanding PaaS, 1005 Gravenstein Highway North, Sebastopol, CA 95472:O Reilly Media, Inc., 2012.

CERTIFICATE

GUVI

Google for Education Partner

ISO 9001-27001 CERTIFIED

KARUPPAIAH MURUGAN

is here by awarded the certificate of achievement
for the successful completion of

First SaaS Application
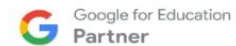
M. Arunprakash

M. Arunprakash
Founder and CEO,
GUVI Geek Networks.

Certificate issued on : October 24 2024

Verified certificate ID: k725J7RVq9S4y51765

Verify certificate at: www.guvi.in/certificate?id=k725J7RVq9S4y51765

GUVI

Google for Education Partner

ISO 9001-27001 CERTIFIED

V.MANJULA DEVI

is here by awarded the certificate of achievement
for the successful completion of

First SaaS Application

M. Arunprakash

M. Arunprakash
Founder and CEO,
GUVI Geek Networks.

Certificate issued on : October 24 2024

Verified certificate ID: 7t6Nj1WEs658707992

Verify certificate at: www.guvi.in/certificate?id=7t6Nj1WEs658707992