University of Reading

Department of Computer Science

Report for E-Learning Web Pages for Computer science students

Roshan Magan

Supervisor: Muhammad Shahzad

A report submitted in partial fulfilment of the requirements of
the University of Reading for the degree of
Bachelor of Science in *Computer Science*

May 1, 2023

Declaration

I, Roshan Magan, of the Department of Computer Science, University of Reading, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and the public with an interest in teaching, learning, and research.

Roshan Magan
May 1, 2023

Abstract

The aim of this project is to design and develop an e-learning website that provides an interactive and user-friendly platform for learners to access educational materials online. The website will utilize the latest web technologies to offer a range of educational resources such as interactive quizzes, discussion forums, and other multimedia content to enhance the learning experience of users.

The website will be designed with a responsive user interface that adapts to different screen sizes and devices, allowing users to access educational materials on desktop computers, laptops, tablets, and smartphones. The website will also include a secure user authentication system to protect user data and ensure privacy.

The e-learning website will be developed using the latest web development technologies such as HTML5, CSS3, JavaScript, PHP, and MySQL. These technologies will enable the development of dynamic and interactive web pages that can be accessed by learners from anywhere in the world. The website will be deployed on a secure and reliable web server to ensure maximum uptime and availability.

Overall, this e-learning website will provide an innovative and effective approach to learning, allowing users to access educational materials anytime, anywhere, and at their own pace. It has the potential to revolutionize the way education is delivered and accessed, making learning more accessible and engaging for all.

Keywords: a maximum of five keywords/keyphrase separated by commas

Report's total word count: we expect a maximum of 20,000 words (excluding reference and appendices) and about 50 - 60 pages. [A good project report can also be written in approximately 10,000 words.]

Acknowledgements

An acknowledgements section is optional. You may like to acknowledge the support and help of your supervisor(s), friends, or any other person(s), department(s), institute(s), etc. If you have been provided specific facility from department/school acknowledged so.

Contents

List of Figures

List of Tables

List of Abbreviations

SMPCS	School of Mathematical, Physical and Computational Sciences

Chapter 1

Introduction

In recent years, e-learning has emerged as a popular and effective approach to education. With the increasing availability of online resources, learners can access educational materials from anywhere, at any time. However, designing and developing e-learning websites that provide a high-quality and engaging user experience remains a challenge.

This project aims to address this challenge by designing and developing an e-learning website with three pages covering topics in virtual reality, matrix multiplication, and graph tree data structure. The website will utilize HTML, CSS, and JavaScript technologies to create a dynamic and interactive user interface. It will be designed with a responsive layout that adapts to different screen sizes and devices, making it accessible to learners from all over the world.

The virtual reality page will provide learners with an immersive experience that introduces them to the principles and applications of virtual reality technology. The matrix multiplication page will provide learners with a comprehensive understanding of matrix multiplication and the skills necessary to solve matrix problems. Finally, the graph tree data structure page will teach learners the basics of graph theory and the graph tree data structure.

This project aims to provide learners with an innovative and effective approach to learning, allowing them to access educational materials anytime, anywhere, and at their own pace. By doing so, it has the potential to revolutionize the way education is delivered and accessed, making learning more accessible and engaging for all. This report will describe the problem, the methodological approach adopted, and the significant outcomes of the project.

1.1 Background

The project at hand is an e-learning website that aims to provide learners with an immersive and engaging educational experience. The motivation behind this project is the increasing need for accessible and effective educational resources in today's digital age. With the widespread availability of technology and the internet, learners can now access educational materials from anywhere, at any time. However, designing and developing e-learning websites that provide a high-quality user experience remain challenging.

To address this challenge, the website will be designed using HTML, CSS, and JavaScript technologies to create a dynamic and interactive user interface. The website will be responsive and adaptive, allowing learners to access it from various devices, including desktops, laptops, tablets, and smartphones. Virtual reality technology has rapidly evolved over the past few years, and it has numerous applications in various fields, including education. By using virtual reality technology in education, learners can experience immersive and interactive simulations that help them better understand complex concepts. The virtual reality page of the e-learning website aims to introduce learners to the principles and applications of virtual reality technology and provide them with an immersive learning experience.

Matrix multiplication is a fundamental mathematical concept with numerous applications in various fields, including computer science, physics, and economics. The matrix multiplication page of the e-learning website aims to provide learners with a comprehensive understanding of matrix multiplication and the skills necessary to solve matrix problems. Graph theory is a branch of mathematics that studies graphs, which are mathematical structures used to model pairwise relations between objects. Graph theory has numerous applications in various fields, including computer science, physics, and social network analysis. The graph tree data structure page of the e-learning website aims to teach learners the basics of graph theory and the graph tree data structure.

By providing learners with an innovative and effective approach to learning, this project aims to revolutionize how education is delivered and accessed, making learning more accessible and engaging for all.

1.2 Problem statement

While e-learning has become increasingly popular in recent years, there is still a need for accessible and practical educational resources that provide a high-quality user experience. Designing and developing e-learning websites that engage learners and provide them with an immersive educational experience remain a challenge. In addition, the widespread availability of technology and the internet has made it easier for learners to access educational materials. Still, it has also created a need for responsive and adaptive websites, allowing learners to access them from various devices.

To address these challenges, this project aims to design and develop an e-learning website with three pages covering topics in virtual reality, matrix multiplication, and graph tree data structure.

The problem this project aims to solve is the lack of accessible and practical e-learning resources that provide a high-quality user experience. By designing and developing an e-learning website that engages learners and provides them with an immersive educational experience, this project aims to provide learners with an innovative and practical approach to learning. Additionally, by designing the website with a responsive layout, it seeks to address the need for websites that are accessible from a variety of devices.

key problems:

1. Limited accessibility: Traditional classroom-based learning requires learners to be physically present in a specific location at a specific time, which can limit access for those who are geographically distant, physically disabled, or have other commitments that prevent them from attending classes in person.

2. Limited flexibility: Traditional classroom-based learning often follows a rigid schedule with set class times and deadlines. This can limit the ability of learners to progress at their own pace and work on the material when it is most convenient for them.

3. Limited interactivity: Traditional classroom-based learning can be passive, with learners primarily listening to lectures or reading materials without much interaction. This can limit the ability of learners to engage with the material and deepen their understanding.

4. Limited engagement: Traditional classroom-based learning can be less engaging and motivating for learners, leading to lower retention of material and decreased motivation to continue learning.

5. Limited customization: Traditional classroom-based learning often follows a one-size-fits-all approach, with all learners progressing through the material simultaneously. This can limit the ability of learners to customize their learning experience to their individual needs and interests.

1.3 Aims and objectives

Aims: This project aims to provide web pages to teach computer science concepts to students. These web pages will allow students to understand these topics or concepts in a fun and interactive way, rather than the traditional classroom style. This will enable students to learn computer science concepts such as Matrices, virtual reality, and graph data structure faster by engaging with the content rather than reading the ideas from the textbooks or articles. This will also enable students to test their knowledge through online quizzes. There will be one quiz on every topic web page to test your knowledge. There also will be a login system in place for security reasons. Students can create an account and log into the website using their credentials. Apart from security, another essential reason is that I will be hosting this website on my homemade server, which will not be as powerful as the cloud-based services, which have more bandwidth than my homemade server. That's the reason I want to limit the users of my website.

Objectives: The objective of e-learning web pages (website) is to provide users with web pages which are interactive and engaging for the students to learn a specific topic in computer science. This project will use HTML5, CSS3 and javascript for the front end and PHP, SQL, and MySql for the back end will be used to create web pages.

To get user feedback will use a feedback form to collect feedback from the users. It will have its page for getting feedback.

1.4 Solution approach

The solution approach for this project involves creating a set of interactive web pages that aim to teach computer science concepts to students more engagingly and effectively. The project will use text, graphics, and interactive elements to help students learn topics such as matrices, virtual reality, and graph data structures. Additionally, a quiz will be included on each topic page to allow students to test their understanding and retention of the material.

A login system will be implemented for the website to ensure security and provide access control. The website will be hosted on a homemade server, using Docker images to support Apache, MySQL, and PHP. Challenges were faced and overcome throughout the development process, including learning new programming languages such as PHP and JavaScript.

The solution approach aims to provide a comprehensive and engaging learning experience for computer science students while providing instructors with a platform for managing and tracking student progress through integration with popular Learning Management Systems.

1.4.1 A subsection 1

You may or may not need subsections here. Depending on your project's needs, add two or more subsection(s). A section takes at least two subsections.

1.5 Summary of contributions and achievements

Throughout this project, we have achieved the following contributions and accomplishments:

1. Designed and developed a comprehensive e-learning website that allows students to learn computer science concepts in a fun and interactive way.

2. Incorporated multimedia elements such as text-to-speech, images, and interactive quizzes to engage and motivate students.

3. Implemented a secure login system to ensure the privacy of user data and mitigate potential security risks.

4. Hosted the website on a homemade server using Docker images, demonstrating an understanding of server management and deployment.

5. Conducted extensive research on e-learning and related technologies, including LMS integration, and provided recommendations for future development.

6. Gained valuable experience in programming languages such as PHP and JavaScript, improving technical skills.

7. Successfully met all project objectives and requirements, providing a robust solution to the problem of traditional classroom learning.

1.6 Organization of the report

Describe the outline of the rest of the report here. Let the reader know what to expect ahead in the report. Describe how you have organized your report.

Example: how to refer a chapter, section, subsection. This report is organised into seven chapters. Chapter 2 details the literature review of this project. In Section 3...

Note: Take care of the word like "Chapter," "Section," "Figure" etc. before the LaTeXcommand \ref{}. Otherwise, a sentence will be confusing. For example, In 2 literature review is described. In this sentence, the word "Chapter" is missing. Therefore, a reader would not know whether 2 is for a Chapter or a Section or a Figure.

Chapter 2

Literature Review

2.1 Introduction:

The field of education has experienced a significant shift towards e-learning in recent years, with technology enabling students to access educational materials and resources online. E-learning has proven to be an effective approach to delivering educational content and has been widely adopted across various institutions. In this literature review, we will discuss the state-of-the-art in e-learning and examine the existing literature and products/systems related to our project.

2.2 State-of-the-Art:

E-learning has become increasingly popular over the years, with technological advancements allowing for greater accessibility and flexibility. According to a report by Ambient Insight, the global e-learning market is expected to reach $325 billion by 2025 (Kaurav, Rahul, Sneha Rajput, and Ruturaj Baber, 2019). The report also indicates that the demand for online courses has increased due to the COVID-19 pandemic. This shift towards e-learning has created new opportunities for educators to provide engaging and interactive content to students.

2.3 Existing Literature and Systems:

There are several e-learning platforms and systems available today that offer a range of educational content. One such platform is Coursera, which provides online courses from top universities and organizations. Another platform is Udemy, which offers a variety of courses created by industry professionals. These platforms have proven effective in delivering educational content to students and have gained a significant following. Regarding e-learning systems for computer science, there are several popular platforms, such as Codecademy, FutureLearn, and Khan Academy. These platforms offer a range of courses in programming, web development, data science, and other related topics. The use of virtual reality in e-learning has also gained attention in recent years, with platforms like Google Expeditions providing immersive educational experiences.

2.4 Relevance to Intended Application:

This project aims to provide web pages for teaching computer science concepts to students in a fun and interactive way. The existing literature and products/systems in e-learning have shown the potential for providing engaging educational content, and this project aims to build on these concepts. The focus on virtual reality and gamification in e-learning seeks to provide students with a unique and memorable learning experience.

2.5 Critique of Existing Work:

While there are several effective e-learning platforms and systems available, there are also challenges that need to be addressed. One challenge is more interaction between students and teachers in online courses. Another challenge is the quality of educational content, which can vary significantly across platforms. This project aims to address these challenges by providing interactive and engaging content and incorporating a quiz system for testing students' knowledge.

2.6 Summary:

In conclusion, e-learning has become a popular approach to delivering educational content, and there are several effective platforms and systems available today. Our project aims to build on these concepts by providing unique and engaging content for teaching computer science concepts. By incorporating virtual reality and gamification, we aim to provide students with a memorable and effective learning experience.

Chapter 3

Methodology

3.1 My project report text structure

Table 3.1: Example of a software engineering-type report structure

Chapter 1	Introduction	
Chapter 2	Literature Review	
Chapter 3	Methodology	
		Requirements specifications
		Analysis
		Design
		Implementations
Chapter 4	Testing and Validation	
Chapter 5	Results and Discussion	
Chapter 6	Conclusions and Future Work	
Chapter 7	Reflection	

3.1.1 Requirements specification

1. Functional Requirements

   - The website should have separate webpages for different topics such as virtual reality, matrix multiplication, and graph tree data structure.
   - Each webpage should contain interactive content such as text, images, interactive features, and animations to help students understand the concepts.
   - Each webpage should have an online quiz at the end to test the students' understanding of the topic.
   - website should have a login system for students to create accounts and track their progress.
   - The website should have an administrator account to manage user accounts and content.
   - The website should have a feedback system for users to provide comments and suggestions for improvement.

2. Non-functional Requirements

- The website should be responsive and work seamlessly on different devices such as desktops, laptops, tablets, and smartphones.
- The website should be optimized for fast loading speeds to ensure a smooth user experience.
- The website should be secure and protect user data through encryption and other security measures.
- The website should be easily maintainable and scalable for future updates and additions to the content.
- The website should adhere to web accessibility standards to ensure that users with disabilities can access the content.

3. Technical Requirements

   - The website should be built using HTML, CSS, and JavaScript. The website should use a backend programming language such as PHP or Python for server-side scripting.
   - The website should use a database management system such as MySQL or MongoDB to store user data and content.
   - The website should use a web server such as Apache or Nginx to serve the web pages.
   - The website should use version control software such as Git for code management and collaboration.
   - Visual studio code text editor, for coding and maintaining the website.

3.1.2 Analysis

The e-learning website aims to provide an interactive and engaging platform for students to learn computer science concepts such as Matrices, Virtual Reality, and Graph Data Structure. The website will allow students to access content, take online quizzes, and test their knowledge in a fun and interactive way. To achieve these goals, the project will need to fulfil the following requirements:

1. User Authentication: The website should have a secure user authentication system that allows students to create an account, log in, and access content specific to their account. This feature will ensure that only registered students can access the learning materials and online quizzes.

2. Quizzes: The website should have an online quiz system that allows students to test their knowledge of the different computer science concepts. Each topic webpage should have a quiz related to that topic, and students should be able to view their scores and track their progress.

3. Virtual Reality: The virtual reality page should have a simulation system that allows students to experience computer science concepts in a virtual environment. The system should be interactive and provide students with a realistic experience that enhances their learning, Such as using WebXR.

4. Matrix Multiplication: The Matrix Multiplication page should provide an interactive way for students to learn the concept of matrix multiplication, addition, transpose matrix, subtraction etc.; this can be done creating a matrix calculator.

5. Graph Tree Data Structure: The Graph Tree Data Structure page should provide an interactive way for students to learn about graph data structure. The page should have a simulation system that allows students to create, modify and traverse a graph data structure.

6. Compatibility: The website should be compatible with different browsers and devices to ensure students can access the content from any device or location.

By fulfilling these requirements, the e-learning website will provide an engaging and interactive platform for students to learn computer science concepts in a fun and interactive way.

3.1.3 Design

wire frames

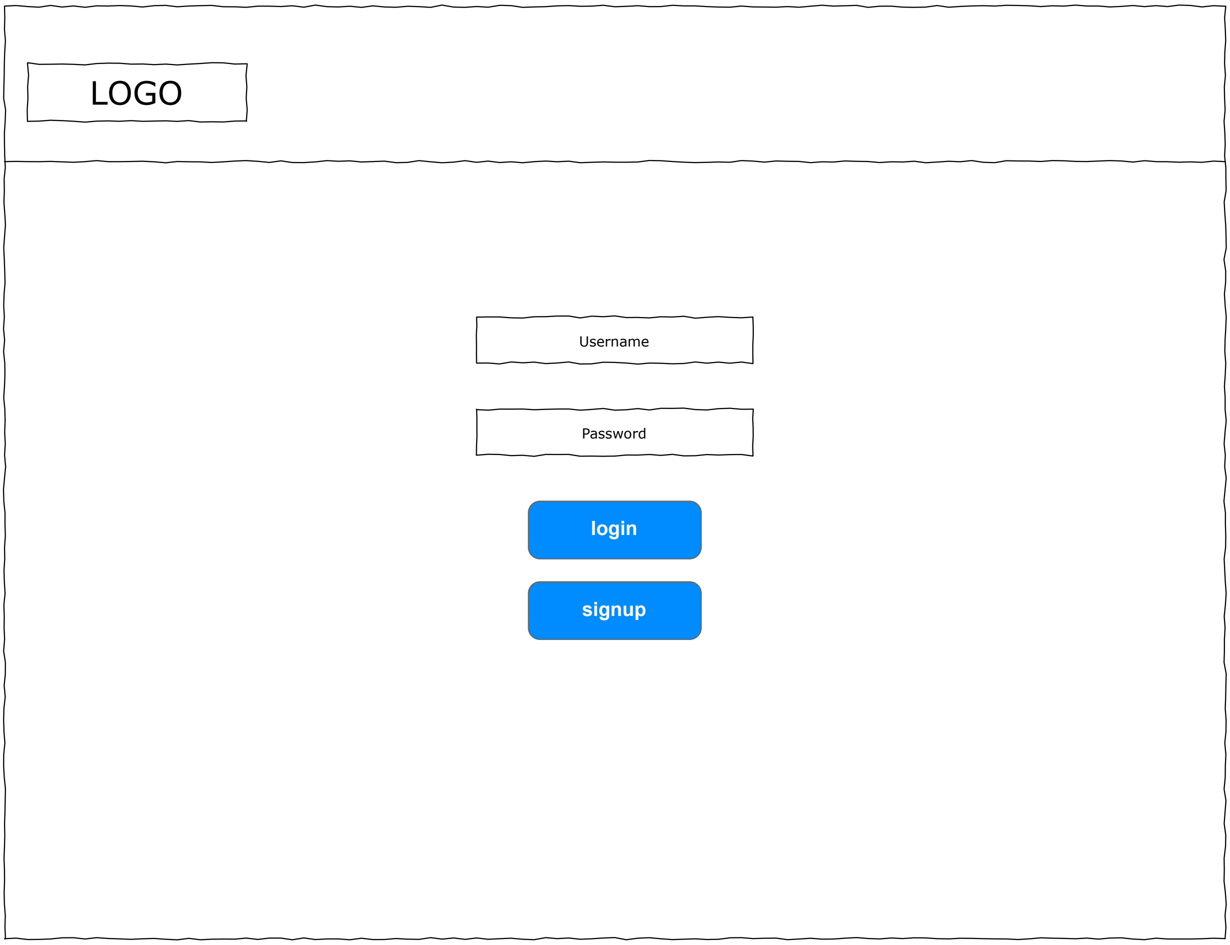


Figure 3.1: Login page wireframe design.

login page: This page design will be simple, containing two input fields, username and password; along with the input fields, there will be two buttons, one for submitting the login credentials and another for registering if the user does not have an account to log in.

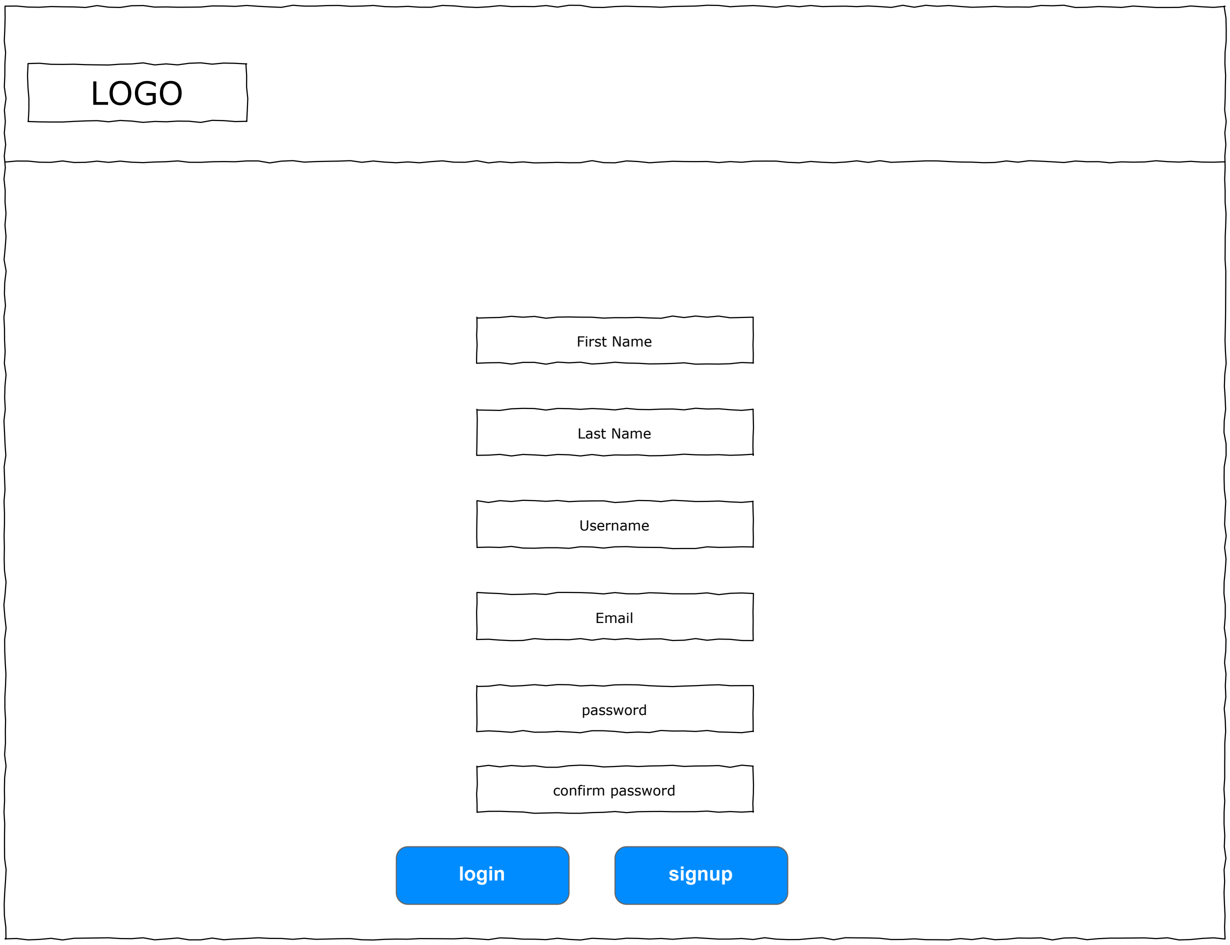


Figure 3.2: Registration page wireframe design.

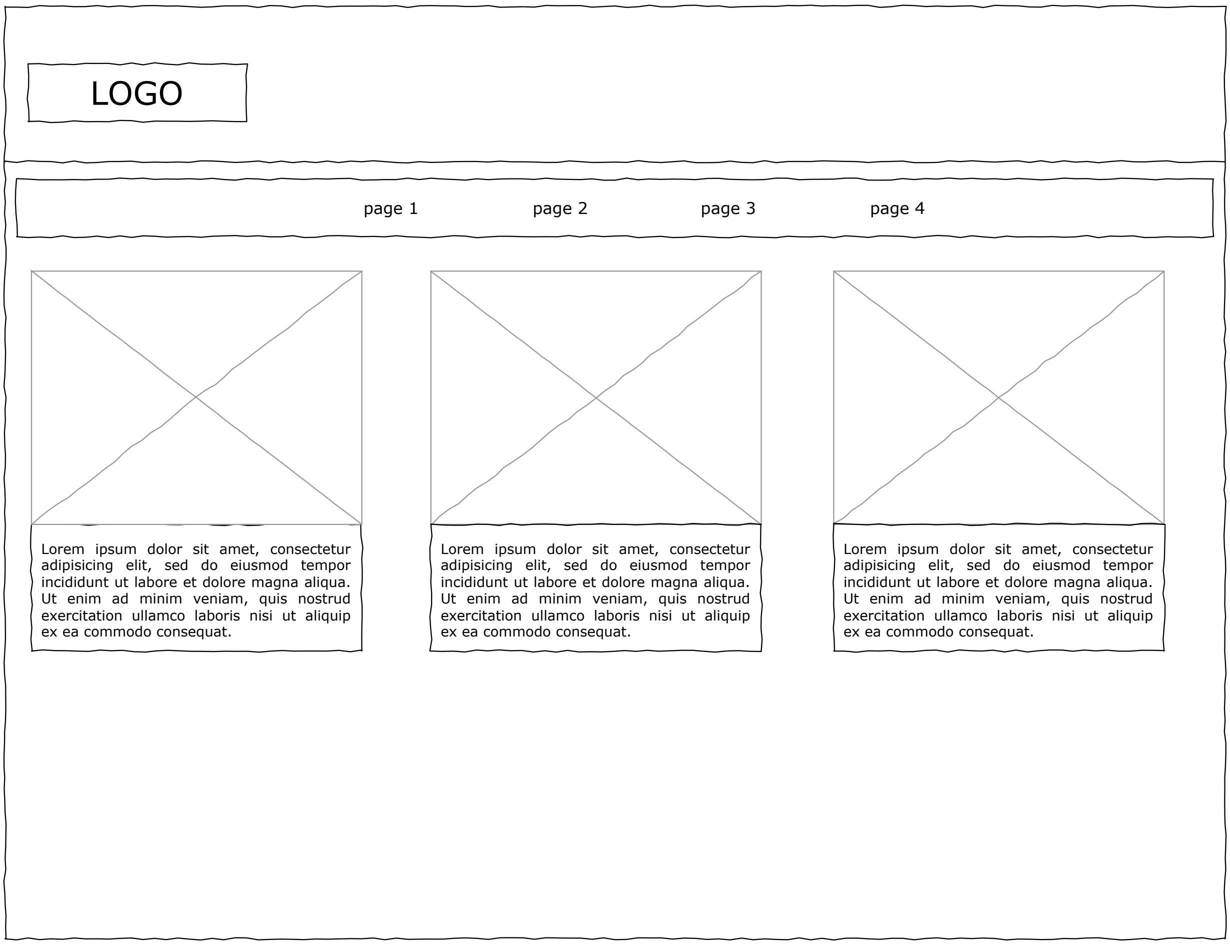
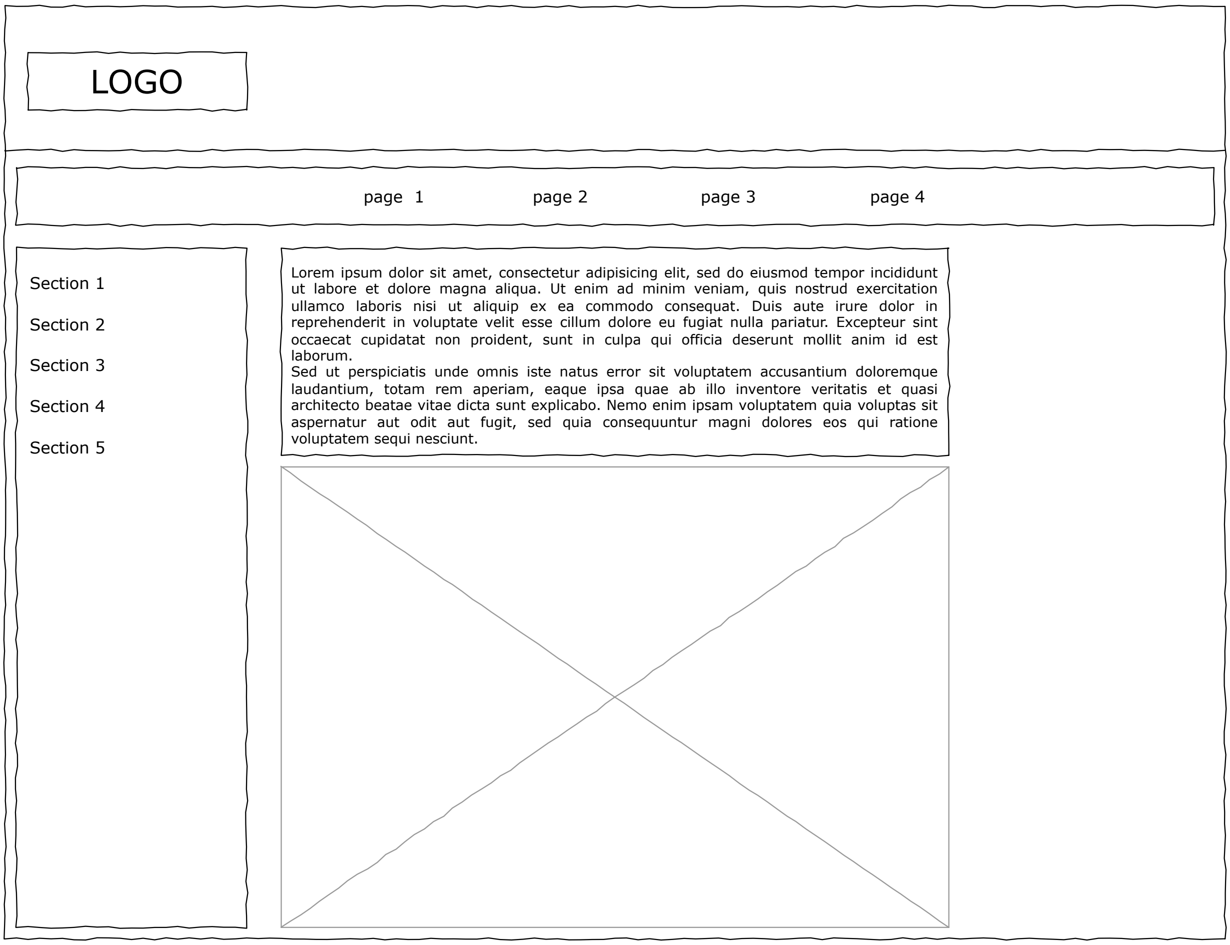

LOGO
page 1
page 2
page 3
page 4
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Figure 3.3: Home page wireframe design



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

Figure 3.4: computer science topic pages matrix calculation, Virtual Reality and Graph Data Structure

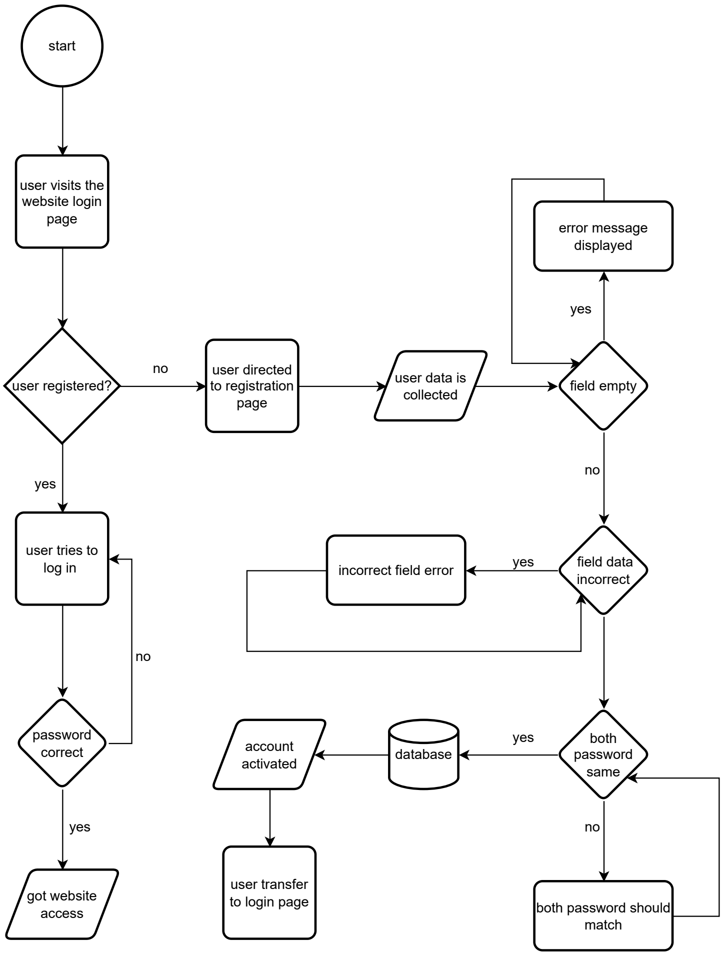


Figure 3.5: flowchart for the user login/registration

3.1.4 Implementation

Development Environment:

As mentioned in the requirement specification section, the things which will be used to develop the website such as:

Hardware

1. Windows laptop was used to access the home server.
2. To make a home server, a 10-year-old laptop was used.

Software and operating systems

1. Visual studio code will be used for coding the website.
2. Linux operating system for the home server
3. Docker for running Apache, PHP, MySql, and PhpMyAdmin (for managing databases)
4. The free domain name from DuckDns.

Front-end Implementation:

3 programming languages, such as HTML, CSS and JavaScript, use for the front-end of the development of the website. First, we will talk about HTML: The first page used on the website, which will be seen by the user when they visit the website, will be the index.html page, which is the website's login page. This is the page user must go through to access the website. The Back-end implementation section provides more details on how the login system works. Here, we will talk about the front end of the website and what users will see on their browsers when they visit it. HTML and CSS are the same for most pages, so all page designs look the same to make the user navigate pages easily without confusion. So, here only one page of HTML and CSS will be explained to avoid repetition.

$$\Huge\textbf{Index page} \tag{3.1}$$

$$\textbf{screenshot} \tag{3.2}$$

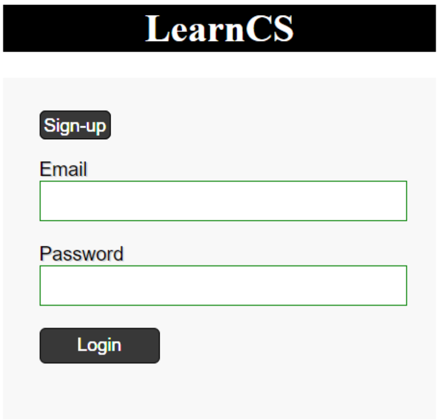


Figure 3.6: Registration page wireframe design.

Here is the code snippet of the HTML code for the index.html

```
<!DOCTYPE html> <!--opening of html page-->
  <head><!--opening of head-->
    <!--this the title of the page -->
    <title>Log in system</title>
    <!--linking a stylesheet for styling my page using css-->
    <link rel="stylesheet" type="text/css" href="loginsystem/style.css">
  </head><!--closing of header-->

  <body><!--open of body -->

  <h1>Multi User Login:</h1><!--heading of the page-->

<!--created a form to get the user input, in this case log in details email and password-->
  <form name = "frmLogin" method = "post" action = "index.php"><!--opening of the form-->

  <a class="btn" href = ’loginsystem/registration.php’>Sign-up</a><!--link to go to the registration page--><br /><br />

  Email<input type = "text" name = "email"></input><br /><br /> <!--email feild to type the email-->
  Password<input type = "password" name = "password"></input><br /><br /> <!--password field to type the password-->
  <input class="btn" type = "submit" name = "Submit" value = "Login"></input><br /> <!--button to submit the log in details-->

  <!--opning of the p tag-->
  <p class="error">
</p> <!--this is the closing p tag-->
  </form> <!--closing form tag-->

  </body> <!--this is the closing body tag-->
</html> <!--this is the closing html tag-->
```

Listing 3.1: Code snippet for index.html page

As you can see in the above snippet 3.1, the HTML part of the website is only a few lines which create a form. In the form, two text fields will take user input, such as credentials(email and password) and two buttons, one for submitting the form and another for registration, if the user is not registered.

This snippet only contains the HTML part of the index.html page, back-end code (PHP code) detail is provided in the back-end section

CSS for index page (3.3)

```
/*Style sheet for whole multi user login system*/
html{
  margin: 8%;
}
/*styling the body tag*/
body
{

  width: 360px;
  margin: 0 auto;
}

/*styling form tag*/
form
{

  text-shadow: 1px 1px #D0D0D0; /*giving a shadow to the text*/
  font-family: arial verdana, sans-serif;
  margin: 0 auto; /*setting form into the middle of the page*/
  padding: 30px;
  width: 300px;
  background-color: #F8F8F8; /*light grey*/

}

/*styling the input feilds*/
form input
{
  width: 300px;
  height: 30px;
  border: 1px solid green;
}

/*giving the red color to all the error messages*/
.error
{
  color: red;
}

/*giving the styling to the heading of the page*/
h1
{
  text-align : center;
  border-bottom: 6px #808080;
  border: 1px solid black;
  background-color: black;
  color: white;

}

/*styling for all the buttons*/
.btn
{
  width: 100px;
  border: 1px solid black;
  text-decoration: none;
  color: white;
  background-color: #383838;
  padding: 3px;
  border-radius: 5px;
  text-shadow: 0px 0px;
  font-size: 15px;

}

/*giving a green color to the success messages*/
.success
{
  color: #00E000;
}
```

```

/*button color when it hover*/
.btn:hover
{
  background-color: grey;
}
```

Listing 3.2: Code snippet for style.css page

The styles are applied to various elements of the HTML code to give the page a consistent and visually appealing appearance. The HTML selector sets the margin to '8 percent' for the whole page. The body selector styles the body tag, sets the width to 360px, and centres it on the page using margin: 0 auto;.

1. The form selector styles the form tag, sets the width to 300px, and gives it a light grey background. The text-shadow property adds a shadow to the text inside the form.
2. The form input selector styles the input fields, giving them a width of 300px, a height of 30px, and a green border of 1px.
3. The h1 selector styles the page heading, sets the text alignment to the centre and gives it a black border of 1px.
4. The .btn selector styles all the buttons, setting the width to 100px, giving them a black border of 1px, a text-decoration of none, and a background colour of 'hex 383838'. The text-shadow property sets the text shadow to 0px.
5. The .error selector gives all error messages a red colour. The .success selector gives all success messages a green colour.
6. Finally, the .btn: hover selector changes the background colour of the buttons to grey when the mouse hovers over them.

There is no JavaScript for the login page because didn't need one; PHP does all the back-end

$$\textbf{Home page} \tag{3.4}$$

$$\textbf{screenshot} \tag{3.5}$$

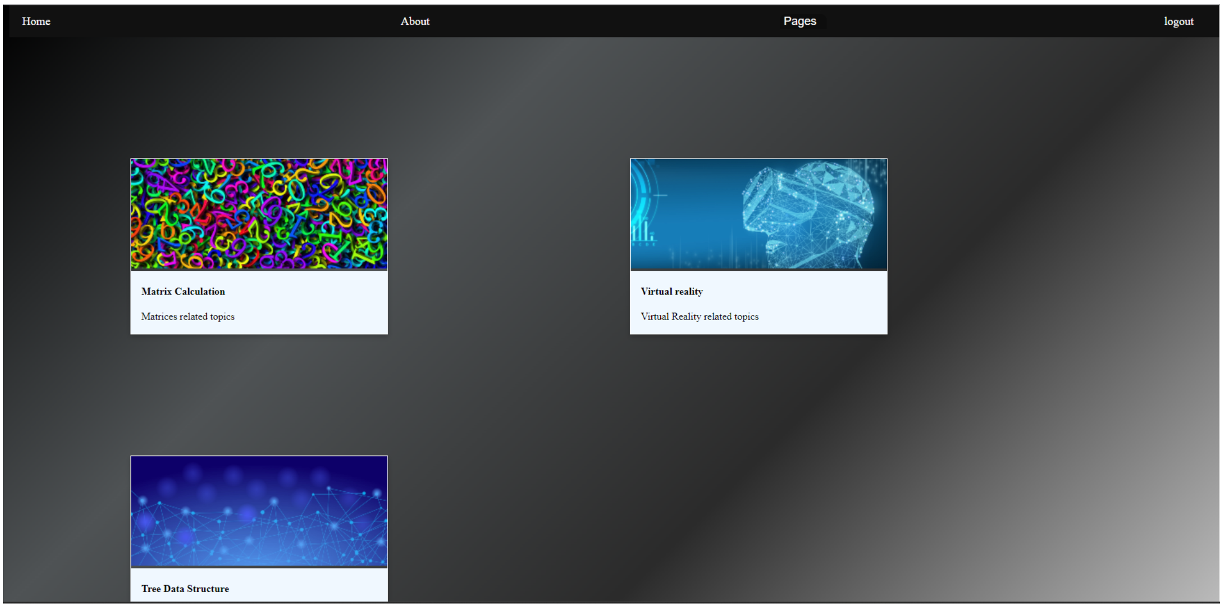


Figure 3.7: Screenshot of the home page.

```

<!DOCTYPE html>
<html>
<head>
  <title>Home</title>
  <link rel="stylesheet" href="css/home.css">
  <link rel="stylesheet" href="css/nav.css">
  <script type="text/javascript" src="javascript/navbar.js"></script>

</head>
<body>
  <div class="navbar">
    <a href="home.php">Home</a>
    <a href="about.php">About</a>
    <div class="dropdown">
      <button class="dropbtn">Pages
        <i class="fa fa-caret-down"></i>
      </button>
      <div class="dropdown-list">
        <a href="matrix.php">Matrices</a>
        <a href="VR.php">Virtual Reality</a>
    <a href="tree-data-structure.php">Tree Algorithms</a>
      </div>
    </div>
    <a href="loginsystem/logout.php">logout</a>
  </div>
  <div class="card">
    <a class="button-a" href="matrix.php">
      <img src="img/math-banner.png" alt="Avatar" style="width:100%">
    </a>
  <div class="container">
    <h4><b>Matrix Calculation</b></h4>
    <p>Matrices related topics</p>
  </div>
  </div>

  <div class="card">
    <a class="button-a" href="VR.php">
      <img src="img/vr-banner.png" alt="Avatar" style="width:100%">
    </a>
  <div class="container">
    <h4><b>Virtual reality</b></h4>
    <p>Virtual Reality related topics</p>
  </div>
  </div>

  <div class="card">
    <a class="button-a" href="tree-data-structure.php">
      <img src="img/tree-banner.png" alt="Avatar" style="width:100%">
    </a>
  <div class="container">
    <h4><b>Tree Data Structure</b></h4>
    <p>Treev Data Structure related topics</p>
  </div>
  </div>
```

Listing 3.3: Code snippet for home.php page

A home page is typically the main landing page of a website and serves as a gateway to other pages on the site. It is usually the first page that a user sees when they visit a website, and it should provide a clear overview of the site's content and purpose.

One common design element of home pages is using cards, and visual elements summarising the content on a specific page. These cards can help users quickly understand the website's offers and decide which pages to visit. So, this website contains three cards related to 3 topics.

In addition to cards, another essential feature added to the home page is a navigation bar. A navigation bar will appear at the top of the page and provides users with links to other pages on the website. The navigation bar will help users quickly navigate between pages and find the necessary information without searching for it. On the navigation bar, there are four buttons home to go to the home page, the about page to find more information about the website, the pages containing a drop-down list of buttons related to each topic, and the logout to logout of the website.

When designing a home page, it's essential to consider the user experience and make the page easy to use and navigate. This can involve using clear headings, visual cues, and other design elements to help users quickly find the necessary information. By making the home page user-friendly, you can help ensure that visitors stay on your site longer and are more likely to return in the future.

Home Page home.css file (3.6)

```

.card {
  box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
  transition: 0.3s;
  width: 400px;
  float:left;
  overflow: hidden;
  margin-right: 10%;
  margin-top: 10%;
  margin-left: 10%;
  border: 1px solid white;
}

.card:hover {
  box-shadow: 0 8px 16px 0 rgba(255, 255, 255, 0.537);
}

.container {
  padding: 2px 16px;
  background-color: aliceblue;
}

.username-txt{
  margin-top: 50%;
  border: 1px solid black;
```

Listing 3.4: Code snippet for home.php page

Added some simple styling to the home page to align the cards next to each other and give them a subtle shadow effect using a property in CSS called box-shadow. This makes it easier for users to browse through the different pages on the site and creates a visually appealing layout.

Another CSS property, ":hover" was used When users hover over the cards, the shadow effect provides a subtle visual cue to indicate that they are interactive and can be clicked on. This helps guide users and makes it easier for them to navigate the site.

My goal with the styling was to create a user-friendly and visually appealing home page that encourages visitors to explore the site further. I believe that even small design touches can make a big difference in the user experience and help to create a positive first impression.

CSS for navigation bar (3.7)

```
    body{
  margin: 0 auto;
  margin-left: 10px;
  background: linear-gradient(-45deg, #bbbbbb, #2b2b2b, #4f5355, #000000);
  background-attachment: fixed;
  background-size: cover;
  animation: gradient 15s ease infinite;
  height: 100vh;
}

/*container for nav bar*/
.navbar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    height: 50px;
    background-color: #111;
    color: #fff;
    padding: 0 20px;

  }

  /*all the a tags*/
  .navbar a {
    color: #fff;
    text-decoration: none;
    margin-right: 20px;
    font-size: 18px;
  }

  /*for hovering effect*/
  .navbar a:hover {
    color: #ccc;
  }

  /*for the dropdown effect*/
  .dropdown {
    position: relative;
    display: inline-block;
  }

  /*dropdown buttons styling*/
  .dropbtn {
    background-color: #0d0d0d;
    color: #fff;
    font-size: 18px;
    border: none;
    cursor: pointer;
    border-radius:10px 10px 0 0;
  }
```

```
/*styling for dropdown list element*/
.dropbtn i {
  margin-left: 10px;
  font-size: 14px;
}

.dropdown-list {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1;
}

.dropdown-list a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-list a:hover {
  background-color: #f1f1f1;
}

.dropdown:hover .dropdown-list {
  display: block;
}

.dropdown:hover .dropbtn {
  padding-top: 10px;
  background-color: #ccc;
}
```

Listing 3.5: Code snippet for nav.css for all pages

The most essential property added in this CSS file is the "display:none property" in class ".dropdown-list" This will hide the dropdown-list element, and it can be set as block level element using JavaScript when the user hovers over the pages button from the navigation bar.

All other properties align the buttons and remove the link decoration like underlines and default colours.

Here is the quick information about the properties use and functionality:

1. The body element has a linear gradient background that changes colour at a -45 degree angle. The background-attachment property is set to fix, and the "background-size" is set to cover. Additionally, an animation is applied to the gradient that lasts 15 seconds and repeats infinitely.

2. The ".navbar" class is a container for the navigation bar. It is displayed as a "flexbox" with space between and aligned items. The background colour is black (hex 111), and the font colour is white. Padding is set to "0" on the top and bottom and "20px" on the left and right.

3. The ".navbar" class sets the styling for the links in the navigation bar. The font colour is white, text-decoration is removed, the font size is "18px", and there is a "20px" margin on the right.

4. The ".navbar" "a:hover" class sets the font colour to grey (hex ccc) when the link is hovered over.

5. The ".dropdown" class is for a "dropdown" menu. The position is relative, and the display is set to "inline-block".

6. The ".dropbtn" class is for the button that opens the "dropdown" menu. The background is black (hex 0d0d0d), the font is white, the font size is "18px", there is no border, the cursor is a pointer, and the border-radius is set to 10px on the top.

7. The ".dropbtn i" class sets the margin on the icon's left to "10px" and the font size to "14px".

8. The ".dropdown-list" class sets the styling for the "dropdown" list. It is displayed as none, and the position is set to absolute. The background colour is set to (hex f9f9f9), the minimum width is "160px", and a box shadow of "0px 8px 16px 0px rgba(0,0,0,0.2)", and the z-index is set to 1.

9. The ".dropdown-list" a class sets the font colour to black, padding on the top and bottom is set to "12px", and on the left and right is set to "16px", text-decoration is removed, and the display is set to block.

10. The ".dropdown-list" a:hover class sets the background colour to (hex f1f1f1) when the link is hovered over.

11. The ".dropdown:hover", ".dropdown-list" class, displays the "dropdown" list when the "dropdown" button is hovered over.

12. The ".dropdown:hover" ".dropbtn" class changes the padding on the top of the "dropdown" button to "10px" and the background colour to grey (hex ccc) when the "dropdown" button is hovered over.

JavaScript for Navigation bar (3.8)

```
    // JavaScript to toggle the dropdown menu

var dropdown = document.getElementsByClassName("dropdown-btn");
var i;

for (i = 0; i < dropdown.length; i++) {
  dropdown[i].addEventListener("click", function() {
    this.classList.toggle("active");
    var dropdownContent = this.nextElementSibling;
    if (dropdownContent.style.display === "block") {
      dropdownContent.style.display = "none";
    } else {
      dropdownContent.style.display = "block";
    }
  });
}
```

Listing 3.6: Code snippet for navbar.js for all pages

This JavaScript code snippet toggles the dropdown menu when a user clicks on it. The code selects all the elements with the class name "dropdown-btn" using the getElementsByClassName method and then adds an event listener to each using a for loop.

When a user clicks on a dropdown button, the addEventListener method triggers a function that toggles the "active" class of the clicked button using the classList.toggle method. Then, the code selects the next element sibling of the clicked button, which should be the dropdown content, and checks if it is currently displayed or not using the style.display property. by default the list is set to display property to none in the nav.css file.

If the dropdown content is currently displayed, the code sets its style.display property to "none", which hides it. If it is not currently displayed, the code sets its style.display property to "block", which shows it.

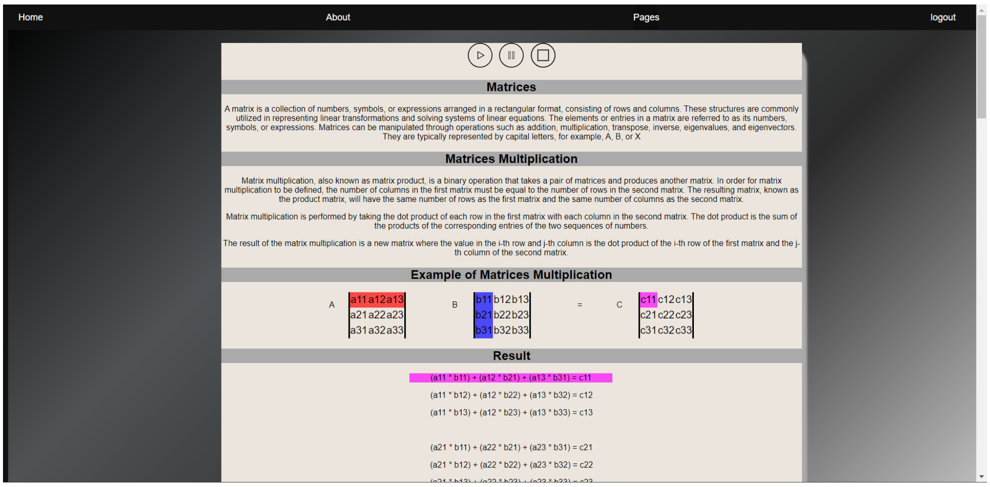
```

<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="css/nav.css">
  <script type="text/javascript" src="javascript/navbar.js"></script>

</head>
<body>
  <div class="navbar">
    <a href="home.php">Home</a>
    <a href="about.php">About</a>
    <div class="dropdown">
      <button class="dropbtn">Pages
        <i class="fa fa-caret-down"></i>
      </button>
      <div class="dropdown-list">
        <a href="matrix.php">Matrices</a>
        <a href="VR.php">Virtual Reality</a>
    <a href="tree-data-structure.php">Tree Algorithms</a>
      </div>
    </div>
    <a href="loginsystem/logout.php">logout</a>
  </div>
```

Listing 3.7: Code snippet of the navbar HTML for all the pages

Note: The navigation code snippet 3.7 for HTML, 3.5 CSS and 3.6JavaScript will be the same for all pages to keep the design consistent throughout the website.

Matrix page (3.9)

screenshot (3.10)



Matrices

A matrix is a collection of numbers, symbols, or expressions arranged in a rectangular format, consisting of rows and columns. These structures are commonly utilized in representing linear transformations and solving systems of linear equations. The elements or entries in a matrix are referred to as its numbers, symbols, or expressions. Matrices can be manipulated through operations such as addition, multiplication, transpose, inverse, eigenvalues, and eigenvectors. They are typically represented by capital letters, for example, A, B, or X

Matrices Multiplication

Matrix multiplication, also known as matrix product, is a binary operation that takes a pair of matrices and produces another matrix. In order for matrix multiplication to be defined, the number of columns in the first matrix must be equal to the number of rows in the second matrix. The resulting matrix, known as the product matrix, will have the same number of rows as the first matrix and the same number of columns as the second matrix.

Matrix multiplication is performed by taking the dot product of each row in the first matrix with each column in the second matrix. The dot product is the sum of the products of the corresponding entries of the two sequences of numbers.

The result of the matrix multiplication is a new matrix where the value in the i-th row and j-th column is the dot product of the i-th row of the first matrix and the j-th column of the second matrix.

Example of Matrices Multiplication

$$A \begin{vmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{vmatrix} \quad B \begin{vmatrix} b11 & b12 & b13 \\ b21 & b22 & b23 \\ b31 & b32 & b33 \end{vmatrix} = \quad C \begin{vmatrix} c11 & c12 & c13 \\ c21 & c22 & c23 \\ c31 & c32 & c33 \end{vmatrix}$$

Result

(a11 * b11) + (a12 * b21) + (a13 * b31) = c11

(a11 * b12) + (a12 * b22) + (a13 * b32) = c12

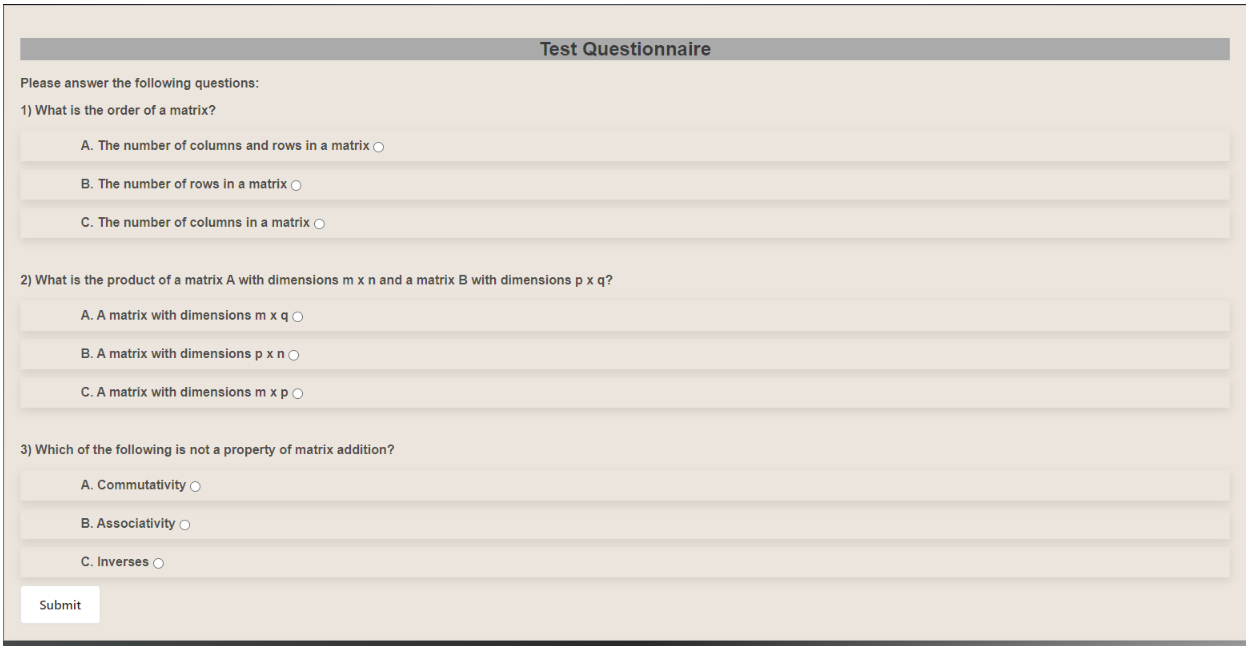
(a11 * b13) + (a12 * b23) + (a13 * b33) = c13

(a21 * b11) + (a22 * b21) + (a23 * b31) = c21

(a21 * b12) + (a22 * b22) + (a23 * b32) = c22

(a21 * b13) + (a22 * b23) + (a23 * b33) = c23

Figure 3.8: matrix page screenshot.



Test Questionnaire

Please answer the following questions:

1) What is the order of a matrix?

A. The number of columns and rows in a matrix ○

B. The number of rows in a matrix ○

C. The number of columns in a matrix ○

2) What is the product of a matrix A with dimensions m x n and a matrix B with dimensions p x q?

A. A matrix with dimensions m x q ○

B. A matrix with dimensions p x n ○

C. A matrix with dimensions m x p ○

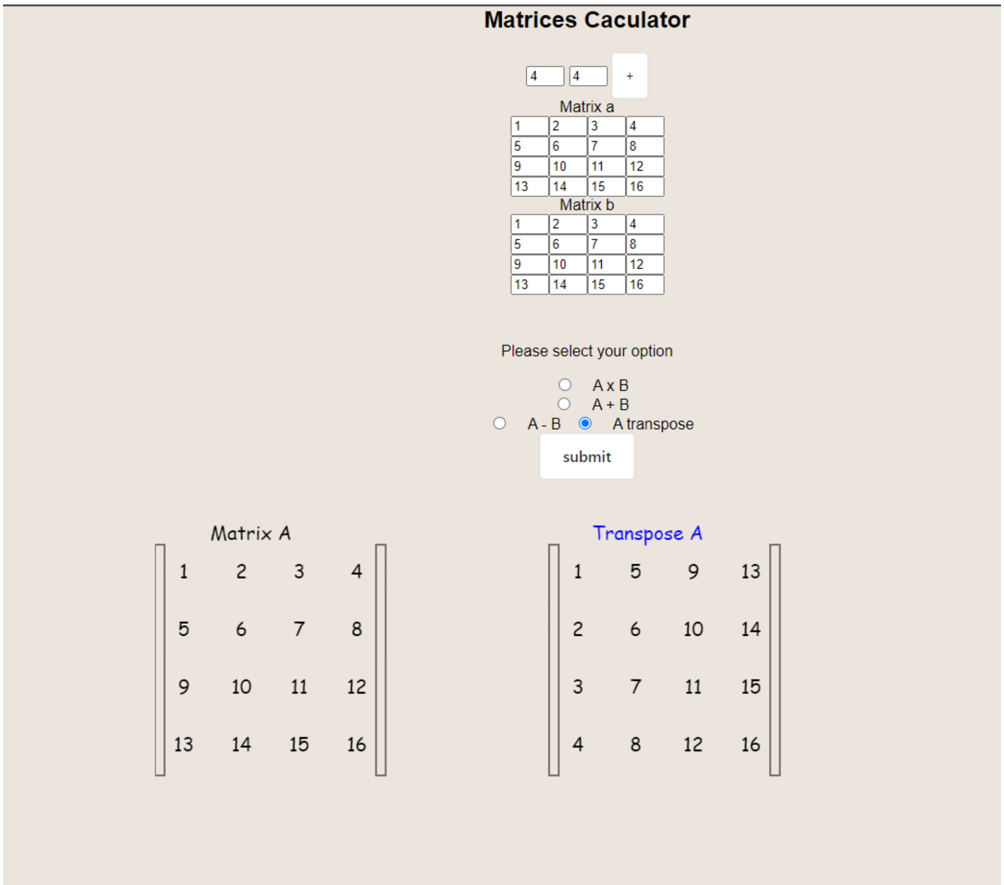
3) Which of the following is not a property of matrix addition?

A. Commutativity ○

B. Associativity ○

C. Inverses ○

Figure 3.9: matrix page screenshot 2.



Matrices Caculator

4 4 +

Matrix a

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Matrix b

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Please select your option

○ A x B
○ A + B
○ A - B ● A transpose

submit

Matrix A

$$\begin{vmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{vmatrix}$$

Transpose A

$$\begin{vmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{vmatrix}$$

Figure 3.10: matrix page screenshot3.

There are three different sections for this page and other e-learning pages.

1. learning material- which provides information on the topic
2. questionnaire - with this section, users can test their knowledge on the topic.
3. interactive section - this will be the section where users can interactively learn the topic, i.e. on the matrix page, there is a matrix calculator.

Throughout all three pages section structure is the same for consistency.

HTML for matrix.php page (section one (learning material))

```
       <div id="container-box">
        <div class="dif-matrices">
        <div class=buttons>
      <button id=play onclick ="textToSpeach()"></button> &nbsp;
      <button id=pause onclick ="textToSpeach()"></button> &nbsp;
      <button id=stop onclick ="textToSpeach()"></button>
    </div>
          <article>
          <h1>Matrices</h1>
              <p>information on matrix</p>


          <h1>Matrices Multiplication</h1>

            <p>information on matrix multiplication</p>

          <div class="multi-box">
              <h2>Example of Matrices Multiplication</h2>

              <div id="tables">

                    <table class="table-a">

                    </table>

                    <table class="table-b">

                  </table>
                  <span><p>=</p></span>
                    &nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp  &nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp
                  <table class="table-c">

                </table>
              </div>
              <h2>Result</h2>
              <div class=result-out>

              </div>
          </div>
            <div class="add-box">
                <h2>Matrices Addition</h2>
                    <div class=matrices-addition>

                    </div>
                    </article>
            </div>

```

```
                </div>
```

Listing 3.8: Code snippet of the matrix html page section 1

This is an HTML code for a webpage that explains matrices and matrix multiplication. It starts with a container box that includes two sections: "Matrices" and "Matrices Multiplication." The "Matrices" section briefly defines matrices and their operations. The "Matrices Multiplication" section explains matrix multiplication and provides an example of how it works.

The matrix multiplication example is presented using tables A, B, and C. Below the tables, the "Result" section provides a step-by-step calculation of how each entry in matrix C is obtained using the dot product formula.

The "dif-matrices" class contains the main content of the webpage. It includes three buttons: "play," "pause," and "stop," which are linked to a JavaScript function named "textToSpeach()" that allows the user to listen to the text. The webpage also includes CSS styling to format the tables, buttons, and text.

HTML for the matrix.php page (section two(questionnaire))

```

          <div id="questions">
            <form id="test-questions">
            <h2>Test Questionnaire</h2>
            <p class="question">Please answer the following questions:</p>
            <div>
            <p class="question">1) What is the order of a matrix?</p>

            <label for="q1-A">A. The number of columns and rows in a matrix
                <input type="radio" id="q1-A" name="question1" value="A" required>
            </label>
            <label for="q1-B">B. The number of rows in a matri
                <input type="radio" id="q1-B" name="question1" value="B">
            </label>
            <label for="q1-C">C. The number of columns in a matrix
                <input type="radio" id="q1-C" name="question1" value="C">
            </label>
            </div>
          <br>
          <div>
            <p class="question">2) What is the product of a matrix A with dimensions m x n and a matrix B with dimensions p x q?</p>
            <label for="q2-A">A. A matrix with dimensions m x q
                <input type="radio" id="q2-A" name="question2" value="A" required>
            </label>

            <label for="q2-B">B. A matrix with dimensions p x n
                <input type="radio" id="q2-B" name="question2" value="B">
            </label>
            <label for="q2-C">C. A matrix with dimensions m x p
                <input type="radio" id="q3-C" name="question2" value="C">
            </label>
          </div>
          <br>
          <div>
            <p class="question">3) Which of the following is not a property of matrix addition?</p>
            <label for="q3-A">A. Commutativity
                <input type="radio" id="q3-A" name="question3" value="A" required>
            </label>
            <label for="q3-B">B. Associativity
                <input type="radio" id="q3-B" name="question3" value="B">
            </label>
            <label for="q3-C">C. Inverses
                <input type="radio" id="q3-C" name="question3" value="C">
            </label>
          </div>
          <button type="submit" id="myBtn">Submit</button>

          <!-- The Modal -->
        <div id="myModal" class="modal">

        <!-- Modal content -->
        <div class="modal-content">
          <span class="close">&times;</span>
          <p id="test-result"></p>
        </div>

        </div>

        </form>
```

Listing 3.9: Code snippet of the matrix html page for section 2

HTML form with three multiple-choice questions about matrices. The questions are labelled with a number and a statement, and each question has three possible answers labelled with the letters A, B, and C. The form requires the user to select an answer for each question by clicking one of the radio buttons next to the answer choices. The "required" attribute on the first radio button of each question makes it mandatory to select an answer before submitting the form. There is also a submit button at the end of the form. When the user submits the form, it triggers a JavaScript function to display the user's test results in a modal window.

HTML for matrix.php page (Matrix calculator section 3)

```
 <div class="container" id="container-matrix">
          <h2>Matrices Caculator</h2>
        <form class="Matrix_multi" action="index.html" method="post">
          <input type="text" id="rowSize" value="4"/>
          <input type="text" id="colSize" value="4"/>
            <button id="create-f" type="button" name="button">+</button>
      <div class="matrix">
        <p class="A">Matrix a</p>
        <div id="Matrix_a">
        <!--matrix a will be defined here through javascript-->
        </div>

          <p class="B">Matrix b</p>
        <div id="Matrix_b">
           <!--matrix b will be defined here through javascript-->
        </div>
        </div>
        <style>
              .inputA,.inputB,.inputC{
                width:40px;
              }
```

```
                  #rowSize,#colSize{
                    width:40px;
                  }


                </style>
                <div id="radio_buttons">
                  <p class="option">Please select your option</p>
                  <input type="radio" id="times" name="arth" value="multi">
                  <label for="AxB">A x B</label><br>
                  <input type="radio" id="plus" name="arth" value="add">
                  <label for="A+B">A + B</label><br>
                  <input type="radio" id="minus" name="arth" value="sub">
                  <label for="A-B">A - B</label>
                  <input type="radio" id="trans" name="arth" value="trans">
                  <label for="A-trans">A transpose</label>
                </div>


            <div id="M_result">
              <input type="button" id="mButton" value="submit"/>
            </div>
                  <br>
        <div class="matrices-out">
          <canvas id="myCanvas" width="900" height="1000"></canvas>
        </div>
          <p>Result</p>
          <div id="Matrix_c">
                 <!--matrix c will be defined here through javascript-->
          </div>
          </form>
          </div>
            <script type="module" src="matrix.js"></script>
            <script type="text/javascript" src="javascript/quiz.js"></script>
```

Listing 3.10: Code snippet of the matrix html page for section 3

HTML code for a matrix calculator. The code includes input fields for the size and values of two matrices (A and B), radio buttons to select the desired operation (multiplication, addition, subtraction, or transpose), and an output field for the resulting matrix (C).

The script "matrix.js" contain all the functionality of the calculator, which is explained in a listing 3.12, containing the code to perform the matrix calculations based on the user input.

CSS for Matrix page

```
    body{
  width: auto;
  text-align: center;
}
#container-box{
  width: 60%;
  text-align: center;
  margin-left: 22%;
  background-color: rgb(159, 157, 157);
  box-shadow: 10px 5px 4px #aaaaaa;
  border-radius: 10px 40px 40px 10px;
}
```

```

.container{
  /* border: 1px solid #BFBFBF; */
  background-color: rgb(235, 229, 221);
  /* box-shadow: 10px 5px 4px #aaaaaa; */
}
.container h2{
  background-color: rgb(235, 229, 221);
}
.matrix{
  display: block;

}
.dif-matrices{
  background-color: rgb(235, 229, 221);
}

h1,h2,h3{
  background-color: #aaaaaa;
}
.multi-box{
  /* border: 1px solid black;  */
  background-color: rgb(235, 229, 221);
}
.A{
  display: inline;
}
.B{
  display: inline;

}
#test-questions {
  border: 1px solid black;
  background-color: rgb(235, 229, 221);
}

/* #myCanvas{
  border: 1px solid black;
} */

.Matrix_multi{
  background-color: rgb(235, 229, 221);
}
/*
#M_Result{
  margin-top: 50px;
  border: 1px solid black;
}

#Matrix_c{
  border: 1px solid black;
}

/* radio buttons */

#radio_buttons{
  margin-top: 50px;
/* border: 1px solid black; */
}

#create-f {
  width: 2%;
  background-color: #FFFFFF;
  border: 1px solid rgba(0, 0, 0, 0.1);
  border-radius: .25rem;
  box-shadow: rgba(0, 0, 0, 0.02) 0 1px 3px 0;
  box-sizing: border-box;
  color: rgba(0, 0, 0, 0.85);
}
#mButton{
  align-items: center;
  background-color: #FFFFFF;
  border: 1px solid rgba(0, 0, 0, 0.1);
  border-radius: .25rem;
  box-shadow: rgba(0, 0, 0, 0.02) 0 1px 3px 0;
  box-sizing: border-box;
  color: rgba(0, 0, 0, 0.85);
  cursor: pointer;
  display: inline-flex;
  font-family: system-ui,-apple-system,system-ui,"Helvetica Neue",
    Helvetica,Arial,sans-serif;
  font-size: 16px;
  font-weight: 600;
  justify-content: center;
  line-height: 1.25;
  margin: 0;
  min-height: 3rem;
  padding: calc(.875rem - 1px) calc(1.5rem - 1px);
  position: relative;
  text-decoration: none;
  transition: all 250ms;
  user-select: none;
  -webkit-user-select: none;
  touch-action: manipulation;
  vertical-align: baseline;
  width: auto;
}


#mButton:hover,
#mButton:focus {
  border-color: rgba(0, 0, 0, 0.15);
  box-shadow: rgba(0, 0, 0, 0.1) 0 4px 12px;
  color: rgba(0, 0, 0, 0.65);
}

#mButton:hover {
  transform: translateY(-1px);
}

#mButton:active {
  background-color: #F0F0F1;
  border-color: rgba(0, 0, 0, 0.15);
  box-shadow: rgba(0, 0, 0, 0.06) 0 2px 4px;
  color: rgba(0, 0, 0, 0.65);
  transform: translateY(0);
}



#tables{
display: flex;
```

```
justify-content: center;

}
.table-a,.table-b{
 margin-right: 8%;
}
.table-a,.table-b,.table-c{
  border-collapse: collapse;
  border-left: black solid;
  border-right: black solid;
}

td {
  width: 30px;
  height: 30px;
  font-size: 20px;
  text-align: center;
}

.result-out p:nth-child(odd):nth-child(1) {
  width: 35%;
  margin-left: 32.4%;
  animation-name: highlight-result;
  animation-duration: 2s;
}

.table-a tr:nth-child(odd):nth-child(1){
  animation-name: highlight-rows;
  animation-duration: 2s;

}
.table-b td:nth-child(odd):nth-child(1){
  animation-name: highlight-col;
  animation-duration: 2s;
}

.tr-1 td:nth-child(1){
  animation-name: highlight-cell;
  animation-duration: 2s;
}
@keyframes highlight-rows {
  0%{background-color: transparent}
  100%{background-color: rgb(255, 0, 0);}
}
@keyframes highlight-col {
  0%{background-color: transparent}
  100%{background-color: rgb(0, 0, 255);}
}

@keyframes highlight-cell {
  0%{background-color: transparent}
  100%{background-color: rgb(255, 0, 255);}
}

@keyframes highlight-result {
  0%{background-color: transparent}
  100%{background-color: rgb(255, 0, 255);}
}

.animate {
  animation-play-state: running;
```

```
}
```

Listing 3.11: Code snippet of the matrix page css

Here's a breakdown of what the code does:

1. The ”body” selector targets the entire web page and sets the width to ”auto” (the default value), and centres the text.

2. The ”#container-box” selector targets a specific element with the ID ”container-box” and sets its width to 60%, aligns the text to the centre, adds a left margin of 22%, sets the background colour to a greyish tone, adds a box shadow effect, and rounds the corners of the box.

3. The ”.container” selector targets elements with the class ”container” and sets the background colour to a lighter shade of grey.

4. The ”.container h2” selector targets any ”h2” heading element that is a child of an element with the class ”container” and sets the background colour to the same lighter shade of grey used for the container class.

5. The ”.matrix” selector targets elements with the class ”matrix” and sets the display property to ”block”.

6. The ”.dif-matrices” selector targets elements with the class ”dif-matrices” and sets the background colour to the same lighter shade of grey used for the container class.

7. The ”h1,h2,h3” selector targets all heading elements and sets the background colour to a dark grey.

8. The ”.multi-box” selector targets elements with the class ”multi-box” and sets the background colour to the same lighter shade of grey used for the container class.

9. The ”.A” and ”.B” selectors target elements with the classes ”A” and ”B,” respectively, and set the display property to ”inline.”

10. The ”#test-questions” selector targets an element with the ID ”test-questions” and sets the border to a black line and the background colour to the same lighter shade of grey used for the container class.

11. The ”.Matrix_multi” selector targets elements with the class ”Matrix_multi” and sets the background colour to the same lighter shade of grey used for the container class.

12. The ”#create-f” selector targets an element with the ID ”create-f” and sets several properties to create a button-like appearance, including a white background colour, a border, a box shadow, a border-radius, and padding.

13. The ”#mButton” selector targets an element with the ID ”mButton” and sets similar properties as the ”#create-f” selector, as well as properties for the cursor, font, and transition effects.

14. The ”#tables” selector targets an element with the ID ”tables” and sets the display property to ”flex” and the justify-content property to ”centre.”

15. The ".table-a", ".table-b", and ".table-c" selectors target elements with the classes "table-a," "table-b," and "table-c," respectively, and set the border-collapse property to "collapse" and the border properties to black lines.

16. The "td" selector targets all table cells and sets the width, height, font size, and text alignment properties.

17. The ".result-out p:nth-child(odd):nth-child(1)" selector targets every other odd-numbered first child paragraph element within an element with the class "result-out" and sets properties for width, left margin, and an animation effect.

18. The first rule applies to the odd-numbered first child paragraph elements within a container element with the class "result-out". When these elements are encountered, they are set to have a width of 35% and a left margin of 32.4%. They are also given an animation effect named "highlight-result" that lasts 2 seconds.

19. The second rule applies to odd-numbered first-child rows within a table element with class "table-a". When these rows are encountered, they are given an animation effect named "highlight-rows" lasting 2 seconds. This animation changes the background colour of the rows from transparent to red.

20. The third rule applies to odd-numbered first-child cells within a table element with class "table-b". When these cells encounter, they are given an animation effect named "highlight-col" lasting 2 seconds. This animation changes the background colour of the cells from transparent to blue.

21. The fourth rule applies to the first cell of the first row within a table element with class "tr-1". When this cell is encountered, it is given an animation effect named "highlight-cell" lasting 2 seconds. This animation changes the background colour of the cell from transparent to magenta.

22. The next four key frames define the behaviour of the animations. They specify that each animation should start with a transparent background and end with a specific colour. This can be seen in this screnshot 3.8

23. A rule for a class named "animate" sets the animation-play-state property to "running". This class can be added to any element on the page to trigger its animation.

JavaScript for Matrix page (3.11)

This JavaScript code contains all the functionality of the matrix calculator, such as:

1. matrix addition

2. matrix multiplication

3. matrix subtraction

4. matrix transpose

I will break down all functionality by sections using code snippets.

```javascript
// Importing all functions from the 'inverse.js' file and assigning them to the 'inverse' variable
import * as inverse from '/javascript/inverse.js';

// When the window loads, assign the 'mCalculation' function to the click event of the 'mButton' button
window.onload = function(){
  var button = document.getElementById('mButton');
  button.addEventListener("click", mCalculation);
}

// Initialize variables to store the number of rows and columns in the matrix, as well as a counter and an incrementor
var row = parseInt(document.getElementById('rowSize').value);
var column = parseInt(document.getElementById('colSize').value);
var count = 0;
var inc = 0;

// Set initial positions for text and matrix display on the canvas
let textPosX = 100;
let textPosY = 40;
let mPosX = 50;
let mPosY = 410;

// Get the canvas element and create a 2D context object for drawing on it
var canvas = document.getElementById("myCanvas");
let ctx = canvas.getContext("2d");

// Initialize an empty matrix array
var matrix = [];

// Set the font and alignment properties for the canvas context
ctx.font = "20px Comic Sans MS";
ctx.textAlign = "center";
ctx.textBaseline = "middle";

// Function to increment the counter variable
function increament(){
  count+=1;
}

// Get the 'create-f' button element and assign the 'createFeild' function to its click event
let create_f = document.getElementById('create-f');
create_f.addEventListener("click", createFeild);

// Function to perform matrix calculations
function mCalculation(){

}

// Function to multiply two matrices
function multiplyMatrices(a, b) {

}

// Function to add two matrices
function addingMatrices(a,b){

}
```

```
// Function to subtract two matrices
function subMatrices(a,b){

}

// Function to create a matrix input field
function createFeild(){

}

// Function to clear a matrix input field and add a line break
function clearFeild(inputFeild,brElement){

}

// Function to display a matrix on the canvas
function displayMatrices(matrix,x_axies,y_axies){

}
```

Listing 3.12: Code snippet of the matrix page JavaScript file

Above in the code snippet3.12 is the structure of the matrix calculator.

1. Assign the 'mCalculation' function to the click event of the 'mButton' button when the window loads. This function will be called when the button is clicked.
2. Initializing variables to store the number of rows and columns in the matrix, as well as a counter and an incrementor.
3. Setting initial positions for text and matrix display on the canvas.
4. Getting the canvas element and creating a 2D context object for drawing on it.
5. Initializing an empty matrix array.
6. Setting the font and alignment properties for the canvas context.
7. Defining a function to increment the counter variable.
8. Getting the 'create-f' button element and assigning the 'createFeild' function to its click event. Click event is assigned directly on the HTML button on the HTML file.
9. Defining a function 'mCalculation' to perform matrix calculations. This function will be called when the 'mButton' is clicked.

This code is part of a larger program for matrix calculations and display. The 'mCalculation' function will likely perform the actual matrix calculations using other functions defined within the script and then display the results on the canvas. The 'createFeild' function may create input fields for entering matrix values.

This is not the full code. It's just the code snippet for the whole matrix calculator code. Next, we will break down all the functions starting from matrix addition.

Matrix Addition function (3.12)

```
    function addingMatrices(a,b){
  var result = [];
  for (var i = 0; i < a.length; i++) {
    result[i] = [];
    for (var j = 0; j < a[0].length; j++) {
      result[i][j] = a[i][j] + b[i][j];
    }
  }

  return result;
}
```

Listing 3.13: Code snippet of the matrix page JavaScript file

In the above 3.13 "addingMatrices" function takes two matrices as input and returns a new matrix that is the sum of the two input matrices.

Here is how it works:

- First, an empty array called result is created to store the resulting matrix.
- The function then loops through each row of the matrices using a for loop with a loop variable i.
- For each row, the function creates another empty array to store the row of the resulting matrix, using result[i] = [].
- The function then loops through each column of the matrices using a for loop with a loop variable j.
- For each element of the matrices, the function adds the corresponding elements of the input matrices (i.e., a[i][j] and b[i][j]) and stores the result in the corresponding element of the result matrix (i.e., result[i][j] = a[i][j] + b[i][j]).
- After all the elements have been added, the resulting matrix result is returned.

This function calculates the element-wise sum of two matrices by looping through the rows and columns of the matrices and adding the corresponding elements together. The resulting matrix has the same number of rows and columns as the input matrices, and each element is the sum of the corresponding elements of the input matrices.

Matrix Multiplication function (3.13)

```
function multiplyMatrices(a, b) {
  // create an empty matrix to hold the result
  var result = [];

  // iterate over each row in matrix a
  for (var i = 0; i < a.length; i++) {
    // create an empty row in the result matrix
    result[i] = [];

    // iterate over each column in matrix b
    for (var j = 0; j < b[0].length; j++) {
      // initialize a variable to hold the sum
      var sum = 0;

      // iterate over each element in the current row of matrix a
      for (var k = 0; k < a[0].length; k++) {
        // multiply the current element in a with the current element in b
    ,
        // and add the result to the sum variable
        sum += a[i][k] * b[k][j];
      }

      // set the value of the current cell in the result matrix to the sum
      result[i][j] = sum;
    }
  }

  // return the resulting matrix
  return result;
}
```

Listing 3.14: Code snippet of the matrix page JavaScript file

“multiplyMatrices” takes two matrices (‘a’ and ‘b‘) as input parameters and returns their product as another matrix.

Initializes an empty matrix called “result” that will store the product of the two input matrices. It then uses a nested for loop to iterate each row and column of the output matrix.

The outermost loop iterates over each row of a matrix “a”. Within this loop, another loop iterates over each matrix “b” column. For each cell in the output matrix, the function calculates the dot product of the corresponding row in “a” and column in “b”.

The innermost loop iterates over each current row element in “a”, multiplying each element by the corresponding element in the current column of “b” and summing the products together. This sum is then stored as the value in the current cell of the output matrix.

Finally, the resulting matrix “result” is returned by the function.

Matrix Subtraction Function (3.14)

```
// function for calculating matrix subtraction
function subMatrices(a, b) {
  //empty array to hold the resulting matrix
  var result = [];

  // Loop through each row of the matrix a
  for (var i = 0; i < a.length; i++) {
    // Create a new array within the result to represent a new row
    result[i] = [];

    // Loop through each column of matrix a
    for (var j = 0; j < a[0].length; j++) {
      // Subtract the corresponding values in a and b and store the result
     in result[i][j]
      result[i][j] = a[i][j] - b[i][j];
    }
  }

  // Return the resulting matrix
  return result;
}
```

Listing 3.15: Code snippet of the matrix page JavaScript for matrix subtraction function

The subMatrices function takes two matrices, a and b, as arguments and returns a new matrix resulting from subtracting b from an element. The function accomplishes this by iterating over each element of a and b and subtracting them to create a new matrix result.

The resulting array is initialized as an empty array at the beginning of the function. Then for loop is used to iterate over each row of a. For each row, a new row is created within a result. The inner for loop then iterates over each column of a. For each element in a and b with the same indices, the value in b is subtracted from the value in a, and the result is stored in result[i][j].

At the end resulting result matrix is returned.

Matrix Transpose Function (3.15)

```
  export function matrice_transpose(matrice) {
  // Get the rows and columns in the matrix
  const rows = matrice.length;
  const cols = matrice[0].length;

  // new matrix to hold the transpose
  const transpose = new Array(cols);
  for (let i = 0; i < cols; i++) {
    transpose[i] = new Array(rows);
  }

  // calculating the transpose of the matrix
  for (let i = 0; i < rows; i++) {
    for (let j = 0; j < cols; j++) {
      transpose[j][i] = matrice[i][j];
    }
  }

  return transpose;
}
```

Listing 3.16: Code snippet of the matrix page JavaScript for matrix transpose function

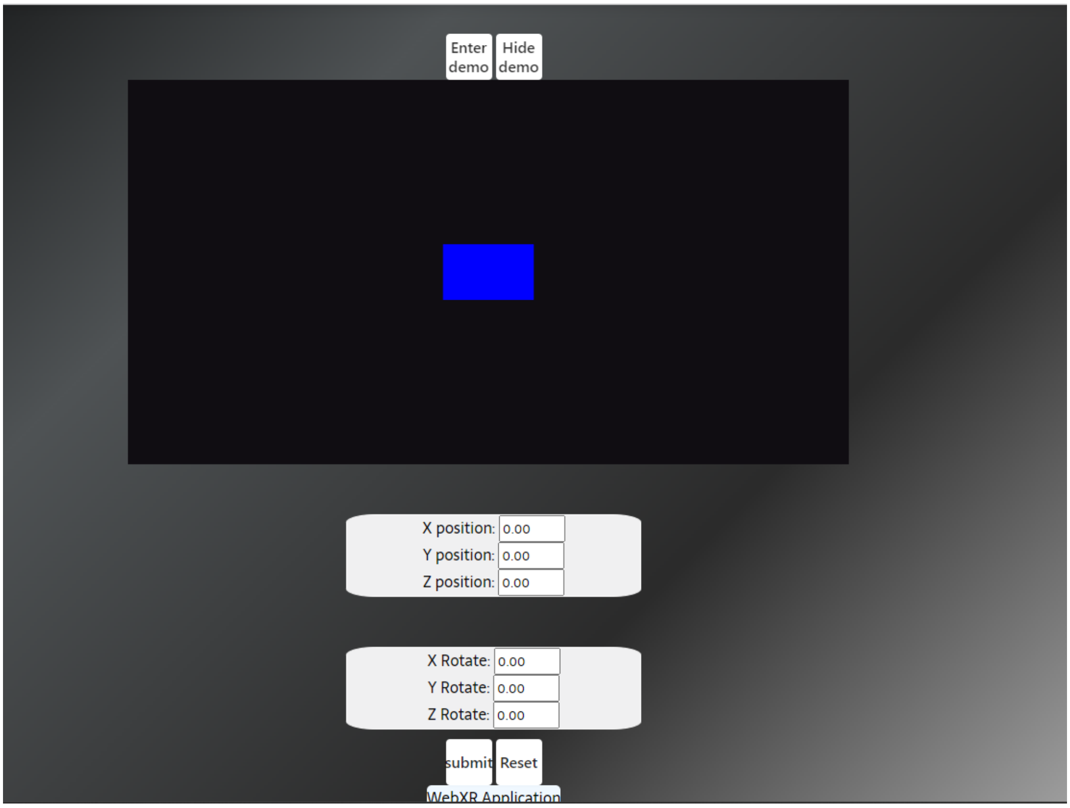
This JavaScript function called ”matrice_transpose” takes a 2D matrix as an argument and returns its transpose.

Here is how it works:

- The function first gets the number of rows and columns in the input matrix using the length property of arrays in JavaScript. It assumes that all rows have the same number of columns (which is why it only checks the length of the first row).
- It then creates a new 2D array called transpose with dimensions cols by rows. This is where the transpose of the input matrix will be stored.
- The function then loops over each element in the input matrix and copies it to the appropriate position in the transpose matrix. This is done using two nested loops.
- Then outer loop iterates over the rows of the input matrix, and the inner loop iterates over the columns of the input matrix.
- The function uses its row and column indices for each element in the input matrix to calculate the corresponding position in the transpose matrix (by swapping the row and column indices). It then copies the element to that position in the transpose matrix.

- Once all elements have been copied over, the function returns the transpose matrix.

Virtual Reality page (3.16)



X position: 0.00
Y position: 0.00
Z position: 0.00

X Rotate: 0.00
Y Rotate: 0.00
Z Rotate: 0.00

Figure 3.11: virtual reality page screenshot.

Note: The CSS and HTML are similar to the matrix page, that's why it did not explain here.

JavaScript for VR.php page (3.17)

JavaScript code that uses the Three.js library to create a 3D scene with a cube and a blue line.

```
        // Initialize the Three.js library and create a 3D scene
    var scene = new THREE.Scene();
    var camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
    var renderer = new THREE.WebGLRenderer({ antialias: true });
    renderer.setSize(750, 400);
    renderer.setClearColor(0x100d12, 1);
    renderer.xr.enabled = true;
    renderer.shadowMap.enabled = true;
    renderer.shadowMap.type = THREE.PCFSoftShadowMap; // default THREE.PCFShadowMap
    document.body.appendChild(renderer.domElement);
```

Listing 3.17: Code snippet of the VR page JavaScript for Virtual reality

Here, we are initializing the Three.js library and creating a 3D scene with a perspective camera and a WebGL renderer. We set the renderer's size to 750 pixels wide and 400 pixels tall. We also enable antialiasing, set the clear colour of the renderer to a dark blue colour, and enable shadows with the PCFSoftShadowMap type. Finally, we add the renderer's dom element to the body of the HTML document.

```
    //create a blue LineBasicMaterial
    const m = new THREE.LineBasicMaterial( { color: 0x0000ff } );
    const points = [];
    points.push( new THREE.Vector3( -10, 0, 0 ) );
    points.push( new THREE.Vector3( 0, 10, 0 ) );
    points.push( new THREE.Vector3( 10, 0, 0 ) );

    const g = new THREE.BufferGeometry().setFromPoints( points );
    const line = new THREE.Line( g, m );

    scene.add(line);
```

Listing 3.18: Code snippet of the VR page JavaScript for Virtual reality

Here, we create a blue line using the LineBasicMaterial with blue colour. We define the points of the line using the Vector3 class and add them to the points array. We then create a BufferGeometry and set the points of the line as the geometry of the buffer. Finally, we create a Line object with the buffer geometry and the blue material and add it to the scene.

```
    //create a plane material
    const planeGeometry = new THREE.PlaneGeometry( 2000, 2000 );
    planeGeometry.rotateX( - Math.PI / 2 );

    const planeMaterial = new THREE.ShadowMaterial();
    // planeMaterial.opacity = 0.2;

    const plane = new THREE.Mesh( planeGeometry, planeMaterial );
    plane.position.y = -200;
    plane.receiveShadow = true;
    scene.add( plane );
```

Listing 3.19: Code snippet of the VR page JavaScript for Virtual reality

Here, we create a plane geometry with the PlaneGeometry class and rotate it by 90 degrees around the X-axis. We create a ShadowMaterial for the plane and set the opacity to 0.2 if needed. Finally, we create a Mesh object with the plane geometry and the shadow material, set its position and add it to the scene.

```
// Create a cube mesh with a red material
var geometry = new THREE.BoxGeometry();
var material = new THREE.MeshBasicMaterial({ color: 0x0000ff });
/*Right of spawn face*/
console.log(geometry);
console.log(material);
material.lightMapIntensity = 3;
var cube = new THREE.Mesh(geometry, material);
let pos_x = 0.00;
let pos_y = 0.00;
let pos_z = 0.00;
let rot_x = 0.00;
let rot_y = 0.00;
let rot_z = 0.00;
scene.add(cube);
```

Listing 3.20: Code snippet of the VR page JavaScript for Virtual reality

This section of the code creates a cube mesh using the Three.js library. It sets up the cube's geometry using 'THREE.BoxGeometry()', which creates a cube with a default width, height, and depth of 1 unit. Then it sets up the cube's material using 'THREE.MeshBasicMaterial()'.

The 'material.lightMapIntensity' property is set to 3, which controls the lightmap intensity of the cube.

The following lines declare six variables: 'pos_x', 'pos_y', 'pos_z', 'rot_x', 'rot_y', and 'rot_z'. These variables will be used to change the position and rotation of the cube, and they are all initially set to 0.

Finally, the 'scene.add(cube)' line adds the cube to the scene so that it will be rendered.

The code then defines a reset() function, which resets the position and rotation variables for the cube to 0, and a updatePos() function, which retrieves the values from input fields on the HTML page and updates the position and rotation variables for the cube. The camera is positioned five units away from the origin in the z-direction, and a directional light is created and added to the scene. The directional light is positioned at (0,1,0), which means it is pointing downwards.

The renderer is set to use a WebGLRenderer with antialiasing enabled, and the renderer's size is set to 750 x 400 pixels. The clear colour is set to a dark blue, and the renderer is enabled for XR and shadow mapping.

Finally, the animation loop is started using the renderer.setAnimationLoop() function. In the animation loop, the cube's position and rotation are updated based on the values of the position and rotation variables. The scene is then rendered with the updated cube position and rotation.

Two event listeners are added to buttons in the HTML document. One button shows the AR canvas when clicked, and the other hides it.

Tree data structure page (3.18)

Note: The CSS and HTML are similar to the matrix page, that's why it did not explain here.

Javascript for tree-data-structure.php (3.19)

```
const canvas = document.getElementById('canvas');
const ctx = canvas.getContext('2d');
let foundNode = 0;
let delay = 1500;

class Node {
  constructor(value) {
  }
}

class BinaryTree {
  constructor() {}

  add(value) {
  }

  insertNode(node, newNode) {
  }

  find(value) {
  }

  remove(value) {
  }

  //Helper function to empty the binary tree
  empty(){

  }

  // Helper function to get the height of the tree
  getHeight(node) {

  }

  // Helper function to get the x-coordinate of the center of the canvas
  getCenterX() {

  }

  // Helper function to draw a circle representing a node
  drawCircle(x, y, radius, color) {

  }

  // Helper function to draw text inside a circle
  drawText(x, y, text, color) {

  }

  // Recursive function to draw the binary tree
  drawNode(node, level, x, y) {

  }

  // Public function to draw the binary tree onto a canvas element

 draw(canvas) {

}


const output = document.querySelector(’.output’);

document.forms[0].addEventListener(’submit’, function(event) {
  event.preventDefault();
  const value = parseInt(document.querySelector(’#insert-value’).value);
  if(value){
    foundNode = value;
    tree.add(value);
    tree.draw(canvas);
    output.innerHTML = ‘Inserted ${value} into the tree‘;
  }else{
    output.innerHTML = ‘${value} values are allowed‘
  }

  document.querySelector(’#insert-value’).value = ’’;
});
```

```


document.forms[1].addEventListener(’submit’, function(event) {
  event.preventDefault();

  const value = parseInt(document.querySelector(’#find-value’).value);
  if(value){
    const Findnode = tree.find(value);
    tree.draw(canvas);
    if (!Findnode) {
      output.innerHTML = ‘${value} not found in the tree‘;
    } else {
      output.innerHTML = ‘Found ${value} in the tree‘;
    }
  }else{
    output.innerHTML = ‘${value} values are allowed‘
  }

  document.querySelector(’#find-value’).value = ’’;

});

document.querySelector(’#remove-value’).addEventListener(’click’, function(event) {
  const value = parseInt(document.querySelector(’#find-value’).value);
  if(value){
    const Findnode = tree.find(value);
    if(Findnode){
      tree.remove(value);
      tree.draw(canvas);
      output.innerHTML = ‘removed ${value} from the tree‘;
    }else{
      output.innerHTML = ‘${value} not found in the tree‘;
    }

 }else{
  output.innerHTML = ‘${value} values are allowed‘
 }
  document.querySelector(’#find-value’).value = ’’;
});

tree.draw(canvas);

document.querySelector(’#generate-value’).addEventListener(’click’, function(event) {
  tree.empty();
  let tempRand = 0;
  let rand = [];
  for (let i = 0; i< 7;i++){
    tempRand = Math.floor(Math.random() * 100);
    tree.add(tempRand+i);
    console.log(tempRand+i);
    rand[i] = tempRand+i;
  }
  output.innerHTML = ‘${rand}‘
  delay = 1500;
  tree.draw(canvas);
});
```

Listing 3.21: Code snippet of the Tree data structure page JavaScript for binary tree

This code implements a binary tree data structure and provides methods to add, remove, and find nodes in the tree. It also provides a way to draw a tree on an HTML canvas.

The 'Node' class represents a single node in the tree, with a 'value' property and 'left' and 'right' pointers to its children. The 'BinaryTree' class represents the tree, with a 'root' property pointing to the node. The 'add' method adds a new node to the tree, while the 'insertNode' method recursively finds the correct place to insert the node based on its value. The 'find' method searches the tree for a node with a given value and returns the node if found or false if not found. The 'remove' method removes a node with a given value from the tree.

The 'getHeight' method is a helper function that recursively calculates the height of a node in the tree. The 'getCenterX' method is a helper function that returns the x-coordinate of the centre of the canvas. The 'drawCircle' and 'drawText' methods are helper functions that draw a circle and text inside the circle, respectively. The 'drawNode' method recursively draws the tree on the canvas, starting from the root node and working down the tree.

The 'delay' variable and the 'setTimeout' calls in the 'drawNode' method provide an animation effect when drawing the tree, with a delay between drawing each tree level. The 'found node' variable highlights the node with a given value when found using the 'find' method.

This code provides a useful implementation of a binary tree data structure, with the bonus of visualising the tree on an HTML canvas.

Back-end Implementation:

PHP and Sql programming language is used for the website's back end. The database used for this project website is MySql, a relational database. The database was needed to store user login credentials when registering with the website and for verification purposes when a user tries to login into the system.

Five different PHP code files for the back-end implementation, and some PHP code is directly implemented along with HTML using the "PHP" tag.

Here is the list of files:

- dbconnect.php - this is used for establishing a connection with the database.
- errorlog.php - this is to log errors into the text file
- logout.php - this contains the script for logging out the user from the website
- members.php - this is to check if the user is logged in and give a success message on successful login.
- registration page - for checking if a user is eligible for registration.

PHP for database connection (3.20)

```
    <?php
//This variables are for database connection//
$host = '192.168.0.12:3306'; //this is to store the host name
$UserName = 'root'; //this is to store the username of the server
$Password = '6Jh[r(CfAKw!FA1q'; // this is to store password of the server
$dbName = 'loginsystem'; //this is to storing database name which I have create for this assignment

//This variables are for error log//
$date = date("d/m/y"); //this is to store the current data for error log
$time = date("h:i:sa"); //for storing the current time for error log
$ip = $_SERVER['REMOTE_ADDR']; // for storing user IP address for error log
$errorFile = "text_files/error_log.txt"; //this is for storing the error log text file location
$err_message = ""; // assigning an error message variable to store error messages to write the errors in the error log file

//making a connection to the database using mysqli connect function, also by checking the hostname, username and password match with the database or not
$connect = mysqli_connect ($host, $UserName, $Password);

//checking for the error that connection is made or not
if(!$connect)
{
  echo $db_connect_err = "Database fail to connect: ". mysqli_error($connect)."\r\n"; //error message is displays if the connection is fail
  $err_message = $date." ".$time." ".$ip." ".$db_connect_err; // storing a error message in the variable along with the date, time and ip for error log
  $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); // oping a error log file using a fopen fucntion and storing a message in local varibale
  fwrite($f_open,$err_message); // writing a file into the error log text file
  fclose($f_open); //closing  the error log text
  exit(); // this will stop the script so, futher code is not exicute

}

//selcting a database which I have created inside a database using the fucntion called mysqli select db
$identify_db = mysqli_select_db($connect, $dbName);

//cheking for errors that that database is selected or not
if (!$identify_db)
{
  echo $db_select_err = "Database fail to select: ". mysqli_error($connect)."\r\n";//error message is displays if the connection is fail
  $err_message = $date." ".$time." ".$ip." ".$db_select_err; //storing a error message in the variable along with the date, time and ip for error log
  $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); // oping a error log file using a fopen fucntion and storing a message in local varibale
  fwrite($f_open,$err_message); // writing a file into the error log text file
  fclose($f_open); //closing  the error log text
  exit(); // this will stop the script so, futher code is not exicute
}
?>
```

Listing 3.22: Code snippet for database connection

PHP code for connecting to a MySQL database. The code initializes several variables for the database connection, including the hostname, username, password, and database name. It also creates variables for error logging, including the date, time, user IP address, and location of the error log text file.

The mysqli_connect function is used for establishing a connection to the database, with the host, username, and password provided as arguments. If the connection is denied, an error message is displayed, the error message is stored in a variable, and the script is stopped using the exit() function. If the connection is successful, the mysqli_select_db function selects the database identified by the $dbName variable.

If the selection is unsuccessful, an error message is displayed, the error message is stored in a variable, and the script is stopped using the exit() function.

In both cases, the error message is also written to the error log text file, specified in the $errorFile variable. The fwrite function is used to write the error message to the file, and the fclose function is used for closing the file after writing the message.

PHP for errorlog.php (3.21)

```
<?php
    if (’’ == filesize(’text_files/error_log.txt’) )//this to check if the file is empty or not
    {
        echo "no errors at the moment!";//display the message if the file is empty
    }else
    {
        $errorFile = "text_files/error_log.txt";//storing the text file location in the variable
        $f_open = fopen($errorFile ,"r") or die("unable to open a file!"); // opens the file for reading purpose
        $error = fread($f_open,filesize($errorFile)); // reading the file and storing the message on the variable
        fclose($f_open); // this will close the open file
        echo nl2br(nl2br($error)); // displaying the error messages and adding the break line
    }
?>
```

Listing 3.23: Code snippet for logging errors

This PHP script checks whether the specified error log file is empty or not. If the file is empty, it displays a message saying, "No errors at the moment". If the file is not empty, it opens the file for reading, reads the content of the file using the 'fread()' function, closes the file using the 'fclose()' function, and then displays the error messages using the 'echo' statement along with the 'nl2br()' function, which adds a line break after each new line character in the error message to display them on separate lines.

PHP for logout.php (3.22)

```
<?php
    //this is to start the session so, session can be executed
    session_start();

    //this will delete the username variable info
    unset($_SESSION[’name’]);

    //displaying a success message of logged out
    $_SESSION['success'] = "You are now logged out!!";

    //this will redirect the user to login page
    header("Location: /index.php");
?>
```

Listing 3.24: Code snippet for logout functionality

This is a PHP code snippet that performs the following actions:

- Starts a session using the session_start() function.
- Removes the value of the 'name' key from the session using the unset() function.
- Sets a success message in the 'success' key of the session using the $_SESSION superglobal variable.
- Redirects the user to the login page using the header() function and the Location header.

This code logs out a user by removing their session data and redirecting them to the login page with a success message.

PHP for members.php (3.23)

```
    <?php
  //this is to start the session so, session can be executed
  session_start();

  if(!isset($_SESSION['name'])) //this will redirect the user to login page if the user is not log in
  {
  //no session name found no redirect back to login page
    $_SESSION['error_login'] = "please login to enter the Members page";

    //this will redirect the user to login page
    header("Location: index.php");
  }
  //otherwise login OK so allow access to members page
  else
  {

        $success_msg = "You are now logged in!!"; //assigning an success message for the user
        $stat_file = "text_files/members_stat.txt"; //this is for storing the error log text file location
        $f_open = fopen($stat_file,"w") or die("unable to open a file!"); // opens a error log text file for writing a file

        //session for hit counter also checking if the session is set or not
        if(isset($_SESSION['members']))
        {
          $_SESSION['members'] = $_SESSION['members'] + 1; // here session is set and increment by 1 when the user visits this page
          $members_stat_count = "Members Page: ".$_SESSION['members']; // storing the hit counter digit and message in the local variable
```

```
     fwrite($f_open,$members_stat_count); // writing the hit counter message into the text file
     fclose($f_open); // closing the text file
    }
    else
    {
     $_SESSION['members'] = 1;// this is setting the members session for the hit counter by one so, when user first enters into the page so, it won't give 'undefined index error'
     $members_stat_count = "Members Page: ".$_SESSION['members'];// storing the hit counter digit and message in the local variable
     fwrite($f_open,$members_stat_count);// writing the hit counter message into the text file
     fclose($f_open); // closing the text file
    }
    echo "<p>"."Hello ". $_SESSION['name']."</p>";  //displays the welcome message by displaying user's email address
    echo "<p class='success'>".$success_msg."</p>"; //this is to display the log in success message
 }
?>
```

Listing 3.25: Code snippet for members page

This PHP code snippet checks if a user is signed in and has a valid session. The user can access the member's page if the session is valid. The code also includes a hit counter that counts how often a user has visited the member's page.

Here is a breakdown of the code:

- The first line starts the PHP session by calling the session_start() function.
- The if statement checks whether the session variable 'name' is set. If the session variable is not set, the user is not logged in, and the code redirects the user to the login page.
- If the session variable 'name' is set, the else block is executed, and the user can access the member's page.
- The code assigns a success message to the $success_msg variable to display when the user is logged in.
- The code opens a text file named 'members_stat.txt' and assigns it to the $stat_file variable for storing the hit counter data.
- The open () function opens the text file in write mode. The code will display an error message and terminate if the file cannot be opened.
- The code then checks whether the session variable 'members' is set. If the variable is set, the code increments it by 1 and writes the new hit counter value to the text file. If the variable is not set, it initializes it to 1 and writes the value to the text file.
- The code displays a welcome message to the user with their email address.
- The success message is displayed below the welcome message.
- The code also displays a logout button, which the user can click to log out of the member's page.

PHP for user registration (3.24)

```
    <?php
//this is to check if the form method is set to post
if(isset($_POST['submit']))
{
  //variables for the error log
  $date = date("d/m/y"); //storing a date into the variable for error log
  $time = date("h:i:sa"); // storing time into the variable for error log
  $ip = $_SERVER['REMOTE_ADDR'];// this is to store the ip address of the current user
  $errorFile = "/text_files/error_log.txt"; // this is to store the error log text file location
  $err_message = ""; // assigning an error message variable to store error messages to write the errors in the error log file

  //including database connection file to make a connection with the database
  include 'dbconnect.php';

    //variables for the registration form
    $firstName = mysqli_real_escape_string($connect,$_POST['fn']);//this to store first name from the form using gobal post variable
    $lastName = mysqli_real_escape_string($connect,$_POST['ln']);//this to store last name from the form using gobal post variable
    $userName = mysqli_real_escape_string($connect,$_POST['user']);//this to store user name from the form using gobal post variable
    $email = mysqli_real_escape_string($connect,$_POST['email']);//this to store email from the form using gobal post variable
    $password = mysqli_real_escape_string($connect,$_POST['pass']); //this to store password from the form using gobal post variable
    $cfm_pass = mysqli_real_escape_string($connect,$_POST['cfmpass']);// this to store confirm password from the form using gobal post variable
    $email_vali = "/^[a-z\d\._-]+@([a-z\d-]+\.)+[a-z]{2,6}$/i"; // this is for storing regular expression for email validation
    $pass_len = strlen($password); // this will store the password length using string length function

    //this will check if the fields are empty or not using empty fucntion, this will also prevent "undifine index error"
    if(empty($firstName) || empty($lastName) || empty($userName) || empty($email) || empty($password) || empty($cfm_pass))
    {
      echo $empty_fielderr_msg = "please fill in the fields"."\r\n";// error message for the empty feild
      $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$empty_fielderr_msg; //storing error message along with ip, date and time in the variable
      $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); //opening a error log text file
      fwrite($f_open,$err_message); // writing error message into the error log text file
      fclose($f_open); // closing the opened error log text file
      exit(); // exiting the script so, no futher script is excuted

    }else if(!preg_match($email_vali, $email)) //this will check the user mail is valid or not according to reguler expersion and with the use of preg match fucntion
    {
      echo $email_vali_err = "this "."<strong> $email</strong>". " is not valid!" ."</br>". "please enter a valid email!"."\r\n";//error message is displays if the user email is not valid according to reguler experision
      $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$email_vali_err; // storing a error message in the variable along with the date, time and ip for error log
      $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); //oping a error log file using a fopen fucntion and storing a message in local varibale
      fwrite($f_open,$err_message); // writing a file into the error log text file using fwrite function
      fclose($f_open); //closing the opened text file
      exit(); // this will stop the script so, futher code is not exicute
    }else if($password !== $cfm_pass) //checking if the two passwords are match or not
    {
      echo $cfm_pass_err = "Your two password dosen't match!!"."\r\n"; // error message is displayed if two passwords are not match
      $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$cfm_pass_err; //error message is store in the varibale with the date, ip and time for error log
      $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); //error log text file is open
      fwrite($f_open,$err_message);// writing error log file with error message
      fclose($f_open);// closing the error log file
      exit();  // exiting the script so, no futher script is excuted

    }else if ($pass_len < 5)//checking for the password length is less 5 or not and if so, give an error message
    {
      echo $pass_len_err = "your password is too weak!!" ."</br>". "please enter a stronger password, length of more than 5"."\r\n"; // giving an error message if the password length is less than 5
      $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$pass_len_err; //error message is store in the varibale with the date, ip and time for error log
      $f_open = fopen($errorFile,"a+") or die("unable to open a file!");// error log text file is open
      fwrite($f_open,$err_message);// writing error log file with error message
      fclose($f_open);// closing the error log file
      exit();// exiting the script so, no futher script is excuted
    }else
    {
      //variables for the sql query and password hashing
      $salt = "rksnfdh";// this is the encrytion key which will be use for encrypting the password
      $hashed_pass = md5($salt.md5($password.$salt)); // this is the variable which will store hashed password which is hashed using md5 fucntion
      $user_check_query = "SELECT * FROM users WHERE user_name = '$userName' OR email = '$email' OR password = '$password'"; // this a sql query to check if the user is exists in the databse or not
      $run_query = mysqli_query($connect, $user_check_query);// this will execute the query which I created above using a fucntion called mysqli query
      $result = mysqli_num_rows($run_query); // this will store the result of the query in rows, using function called 'mysqli_num_rows'

      //this will check if the query is exicuted or not
      if(!$run_query)
      {
        echo $db_conn_err = "Database connection failed :" . mysqli_error($connect)."\r\n"; // error message is given if the query is connection is fail
        $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$db_conn_err; //error message is store in the varibale with the date, ip and time for error log
        $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); //error log text file is open
        fwrite($f_open,$err_message); // writing error log file with error message
        fclose($f_open); // closing the error log file
        exit(); // exiting the script so, no futher script is excuted
      }else // if the above statement is false than below code will be exicuted
      {
        // this will check for the user is already exists in the database or (for dupilcation purpose)
        if($result == 1)
        {
          echo $user_exist_err = "This user is already exist."."\r\n"; // display error message if user already exists
          $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$user_exist_err;//error message is store in the varibale with the date, ip and time for error log
          $f_open = fopen($errorFile,"a+") or die("unable to open a file!"); //error log text file is open
          fwrite($f_open,$err_message);// writing error log file with error message
          fclose($f_open); // closing the error log file
          exit();  // exiting the script so, no futher script is excuted
        }else
        {
          //this is to insert the user data into the database if there are no errors
          $sql = "insert into users(first_name, last_name, user_name, email, password)
          values('$firstName','$lastName','$userName','$email','$hashed_pass')";//creating a query for inserting data into the database and storing into the variable

          $rs = mysqli_query($connect,$sql); // exicuting the sql query using mysqli query fucntion and storing the result in the variable

          if(!$rs)//checking if the query is run properly for not
          {
            echo $db_conn_err =  "Database connection failed :" . mysqli_error($connect)."\r\n"; // displaying an error message if the query is not run
            $err_message = "[ ".$date." ".$time." ".$ip." "."] ".$db_conn_err;//error message is store in the varibale with the date, ip and time for error log
            $f_open = fopen($errorFile,"a+") or die("unable to open a file!");  //error log text file is open
            fwrite($f_open,$err_message); // writing error log file with error message
            fclose($f_open); // closing the error log file
            exit(); // exiting the script so, no futher script is excuted
          }
          else
          {
            // this will create a session success message which will be display when the user succefully register and redirected to the log in page
            $_SESSION[’success’] = $userName." Registration sucessfully! please log in to visit members page!";
            // user wil be redirected to the login page if the user is succefully register
            header(’Location: /index.php’);

          }

        }


        mysqli_close($connect); // closing the connection
      }
    }
}

?>
</p> <!--cloding p tag-->
```

Listing 3.26: Code snippet for user registration

This is a PHP script for user registration. It first checks whether the hit counter session is set or not. If set, it increments the counter and writes the hit counter message to a text file. If not, it sets the session to 1 and writes the hit counter message to the text file. However, the session is commented out, so this part will not work.

Next, it checks if the form method is set to post. If set, it includes the database connection file and initializes variables for the registration form.

It then checks if any of the fields in the form are empty or not using the empty function. If any fields are empty, it displays an error message and writes it along with the date, time, and IP to an error log file. If the email is not valid according to a regular expression, it also displays an error message and writes it along with the date, time, and IP to the error log file.

Otherwise, it escapes special characters in the form fields using the mysqli_real_escape_string function, gets the length of the password using the strlen function, and inserts the user's information into the database. It then displays a success message. However, there are no error checks for inserting the user's information into the database. So, if the insertion fails, the script will continue to execute and display a success message.

Testing:

For testing the website,Loadninja (2019) website is used, which is an online website testing tool. This allows the tester to create automated scripts for testing the websites, one example screenshot can be seen in A.1.

LoadNinja helps simulate real-world user behaviour and measure web application performance under load. This enables the tester to identify and fix performance issues before they impact users.

Here is how LoadNinja works:

- Recording: LoadNinja allows the tester to record user interactions with web applications using a browser extension. During the recording, LoadNinja captures all the user actions, including clicks, typing, scrolling, and more. Which can be seen on the screen A.1

- Scripting: LoadNinja generates a script that simulates the recorded user interactions once the recording is complete. Then modify the script as needed to create a more realistic load test. 3 scripts where created for this project, one for each page, which you can see in the screenshotA.2

- Load Testing: LoadNinja allows the simulation of thousands of virtual users accessing web applications simultaneously. The load can be customised by setting the number of virtual users, the duration of the test, and other parameters. As seen in the given screenshot A.3 for this project, three scripts are added to the test scenario, which will be tested in the testing process. There will be three visual users for each test script for this specific test.

- Analysis: LoadNinja provides detailed reports that help identify performance bottlenecks and other issues that may affect the user experience. The reports include response time, throughput, error rate, and more. which can be seen on this screenshot **??** for this project. More detailed information is provided in this report on results and discussion chapter 4

Deployment:

To deploy this E-learning website, a Homemade server is used. To create this Homemade server Linux server operating system was installed. Docker was used for hosting the Apache server, and the MySQL database and PhpMyAdmin database management tool were used for maintaining the website hosting.

1. Apache server A.9
2. MySQL Database A.10
3. MyPhpAdmin A.11

```
    version: "3"

services:
    webserver:
        build: ./bin/webserver
        ports:
            - "8008:80"
        volumes:
            - ./www:/var/www/html/
        networks:
            - default
        environment:
            - PUID=1000
            - PGID=1000

    db:
        build: ./bin/mysql
        container_name: 'png-mysql-test'
        restart: 'always'
        ports:
            - "3306:3306"
        volumes:
            - ${MYSQL_DATA_DIR-./data/mysql}:/var/lib/mysql
            - ${MYSQL_LOG_DIR-./logs/mysql}:/var/log/mysql
        environment:
            MYSQL_ROOT_PASSWORD: password
        networks:
            - default

    phpmyadmin:
        build: ./bin/phpmyadmin
        links:
            - db
        environment:
            PMA_HOST: db
            PMA_PORT: 3306
        ports:
            - ’8010:80’
        volumes:
            - /sessions
        networks:
            - default
```

Listing 3.27: Code snippet for user registration

For setting up all of these docker images, a Docker compose file was used, which you can see in this code snippet 3.27

Chapter 4

Results and Discussion

4.1 Results and Discussion

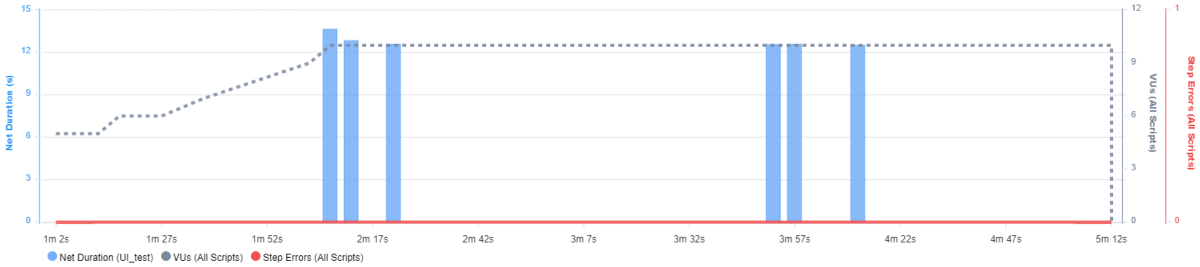
As mentioned in the last chapter 3.1.4, LoadNinja is used for this project website's performance testing.

Performance is tested using the test scripts and scenario features of the LoadNinja. Three test scripts are created, one for each topic page. These scripts are recorded using the AI model and identify the website's clicks, scrolling, and other user interaction. After creating these scripts, they are used in the scenario feature of the LoadNinja to apply those scripts using the servers and visual users.

Using Scenario functionality, those scripts are executed by ten visual users (servers) on the first performance test. The process can be seen in figure A.4

$$\textbf{Test No.1} \tag{4.1}$$

Figure 4.1: Test one summary chart



This table is a performance report for a software application's front-end processing time, with information about the time for different scripts/steps to execute.

The table contains three sections, each representing a different test scenario: "registration-page-test", "tree-data-structure", and "UI_test".

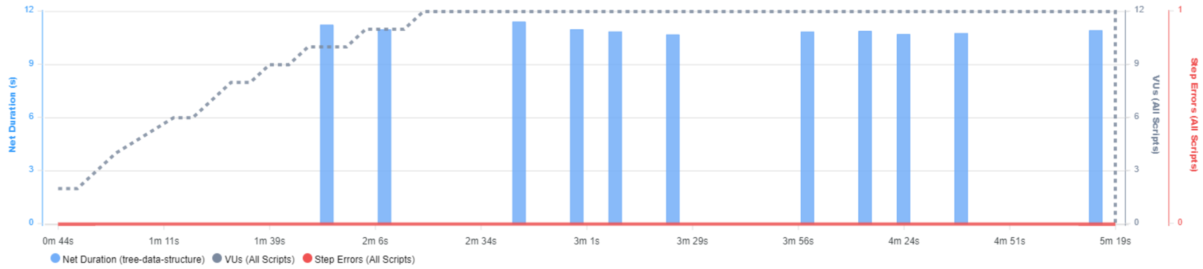
Each section includes the average, minimum, and maximum time taken for each step to execute, the total number of iterations, the total number of steps, and any errors encountered during the test.

Based on the data in the table, it appears that the "registration-page-test" script took the least amount of time to execute (with an average processing time of 16.32 seconds), while the "UI_test" script took the most amount of time (with an average processing time of 11.09 seconds).

The website performed reasonably well on the test, with no significant issues or errors encountered during the tests.

Test No.2 (4.2)

Figure 4.2: Test two summary chart



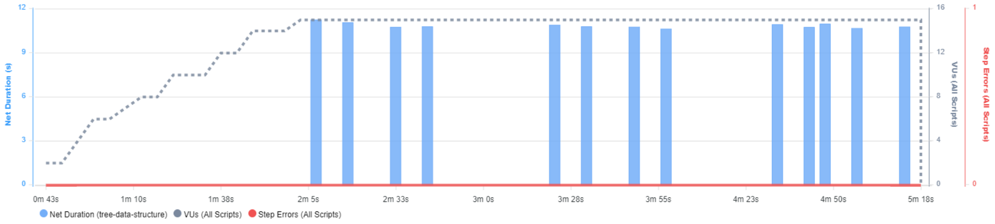
For the second test, visual users are increased to 12 to add more stress to the website and increase server from 1 to 2.

Compared to the first set of results, there are some differences in the processing times for each script/step:

- For the "registration-page-test" script/step, the average processing time increased slightly from 16.32 to 16.34 seconds, but the minimum and maximum processing times remained similar.
- For the "tree-data-structure" script/step, the average processing time decreased slightly from 9.68 to 9.57 seconds, with a lower minimum and maximum processing time.
- For the "matrix_page" script/step, there was a new set of results with an average processing time of 10.93 seconds, a minimum of 10.79 seconds, and a maximum of 11.16 seconds.

Test No.3 (4.3)

Figure 4.3: Test three summary chart



And for the last test, visual users are increased to 15 to add more stress to the website, and the server will remain to 2.

Based on the results, there are improvements in the front-end processing time for some of the tests.

For example, in the "matrix_page" test, the average processing time has decreased from 10.93 seconds to 11.04 seconds, indicating that the script is running slightly faster.

In the "registration-page-test" test, there has been no change in the average processing time, but there have been fewer errors, suggesting that the script has become more stable and reliable.

In the "tree-data-structure" test, the average processing time has decreased from 9.57 seconds to 9.51 seconds, indicating that the script is running slightly faster.

These improvements suggest that the test scripts are becoming more efficient and reliable, which can help ensure that the website being tested is high quality and meets user needs.

Manual Testing (4.4)

Login page was tested for manual testing. This test was conducted to ensure the login page's input field are working as intended. Making sure they are free from SQL infections by escaping the special characters and giving errors message to users if they do anything wrong such as entering the wrong password or trying to submit the form with empty input Fields.

It should also not allow users to access the website with incorrect credentials or unregistered users; it should display an error message if the user enters an incorrect credential.

login page Testing table			
Test Name	expected output	Actual output	Test status
Empty field	error message: "Fields can't be empty! Please fill in the username and password to log in!"	It displays the correct message	Test pass because it displayed the correct error message, which can be seen in this screenshotA.5
SQL Injection test	it should not give out the database table to the screen	an error message: "This email is not registered with us!!"	test pass because it didn't output the tables from the database. Here is the screenshots (1.testA.6) (2.output(A.7
Wrong password: checking if the user has entered the correct credentials; if not, the error message should be displayed	"password is wrong"	"password is wrong"	test pass because an error message was displayed as expected on the wrong password. Here is the screenshot A.8

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In conclusion, this project aimed to provide interactive and engaging web pages to teach computer science concepts to students. By providing these web pages, the aimed to enable students to understand these topics faster and better than they would through traditional classroom-style teaching. The web pages covered three computer science topics, namely matrices, virtual reality, and graph data structures, and included quizzes for students to test their knowledge. We also included a secure login system, which limited users to ensure that the website would not exceed the bandwidth of our homemade server.

The project was implemented using HTML, CSS, and JavaScript technologies for implaementation of the Front-end coding and PHP, SQL and yaml was user for the back-end implamentation. We followed a systematic approach to the design and development of the website, including requirement gathering, analysis, design, implementation, and testing. During the implementation phase, we used various techniques such as responsive design, multimedia elements such as the text to speach feature where users can listen to cotain rather than reading it, and interactive features to enhance the user experience such as Matrix calculator on the matrix page to test and understand their knowleadge on matrices.

Overall, the project has achieved its aims and objectives by providing a unique learning experience for computer science students. The web pages have been designed to be user-friendly, and the quizzes have been designed to test students' knowledge. This project has contributed to the field of e-learning and has the potential to be extended to cover more computer science topics in the future.

5.2 Future work

Although the current implementation of the e-learning website has met the project requirements and objectives, there is still room for improvement and future work. Some potential future works include:

- Adding more interactive and engaging features: While the current implementation provides an interactive way of learning, more features could be added to engage the students further. For instance, virtual labs could be added to allow students to conduct practical experiments or social media integration to allow students to collaborate with their peers.

Also, add some challenging learning games that can be played with their classmate or friends to gain knowledge on specific topics.

- Expanding the topics covered: Currently, the e-learning website covers only a few topics in computer science. However, the website could be expanded to cover more topics, including programming languages, AI (Artificial intelligence), cybersecurity etc.

- Personalization and adaptive learning: Personalization and adaptive learning are becoming increasingly popular in e-learning. The website could be enhanced to provide personalized learning experiences that adapt to each student's learning style and pace. More information can be found on(ELM Learning (2022), 2022).

- Integration with Learning Management Systems (LMS): The website could be integrated with popular LMS platforms such as Blackboard, Canvas, and Moodle to allow instructors to manage and track their students' progress. By integrating the e-learning website with these LMS platforms, instructors can easily manage their courses, create assignments and quizzes, and track their student's progress from a single platform.

- Adding a better server: Currently, the server that is used for the project is Homemade, which is not ideal for website hosting because of the bandwidth issue. In future, this could be shifted to a better server, such as a cloud-based server provided by companies like GoDaddy, Hostinger, BlueHost etc., for better bandwidth and overall ease and performance of the website.

In conclusion, the e-learning website could revolutionize how computer science is taught. The project has successfully implemented a set of web pages that provide an interactive and engaging way of learning, along with online quizzes and a secure login system. The future works outlined above provide a road map for further enhancing the website and making it more effective in teaching computer science concepts to students.

Chapter 6

Reflection

Throughout this project, I have gained significant knowledge and experience in web development, particularly in designing and implementing e-learning websites. This project required me to research various theories and practices in the field of e-learning and explore various programming languages and tools that I had not used before. I sometimes found working with new languages such as PHP and Javascript challenging, but I overcame these challenges through perseverance and self-directed learning. W3School (W3Schools, no date)was a big help in learning these languages because if I had any errors or tried to add new features, this was handy.

One of the key takeaways from this project is the importance of user-centred design in web development. Creating an engaging and user-friendly website requires a profound understanding of the user's needs and preferences. I also learned the importance of incorporating user feedback throughout the development process to ensure that the website meets their needs.

Another essential aspect of this project was using Docker images to host the website on my homemade server. I learned the benefits and challenges of using containerization to deploy web applications. I gained a deeper understanding of how containerization can help to simplify the deployment process and improve application security.

Another new thing got to learn was the Overleaf LATEXeditor, which is suitable for writing professional-looking reports. It was also challenging because it had its own language to learn to write a report. The best thing is that it allows more freedom compared to other text editors like Microsoft Word.

Overall, this project gave me a valuable learning experience and allowed me to develop new web development and e-learning design skills. I am excited to continue exploring this field and applying my newfound knowledge and skills in future projects.

References

ELM Learning (2022) (2022), ‘Elm learning (2022) adaptive learning vs. personalized learning: A guide to both, elm learning’. (May 1, 2023).
URL: *https://elmlearning.com/blog/personalized-learning-vs-adaptive-learning/*

Kaurav, Rahul, Sneha Rajput, and Ruturaj Baber (2019), ‘Factors affecting the acceptance of e-learning by students: A study of e-learning programs in gwalior, india.’. (accessed January 04, 2023).
URL: *https://www.proquest.com/scholarly-journals/factors-affecting-acceptance-e-learning-students/docview/2251594754/se-2*

Loadninja (2019), ‘Loadninja (no date) loadninja’. (April 30, 2023).
URL: *https://app.loadninja.com/*

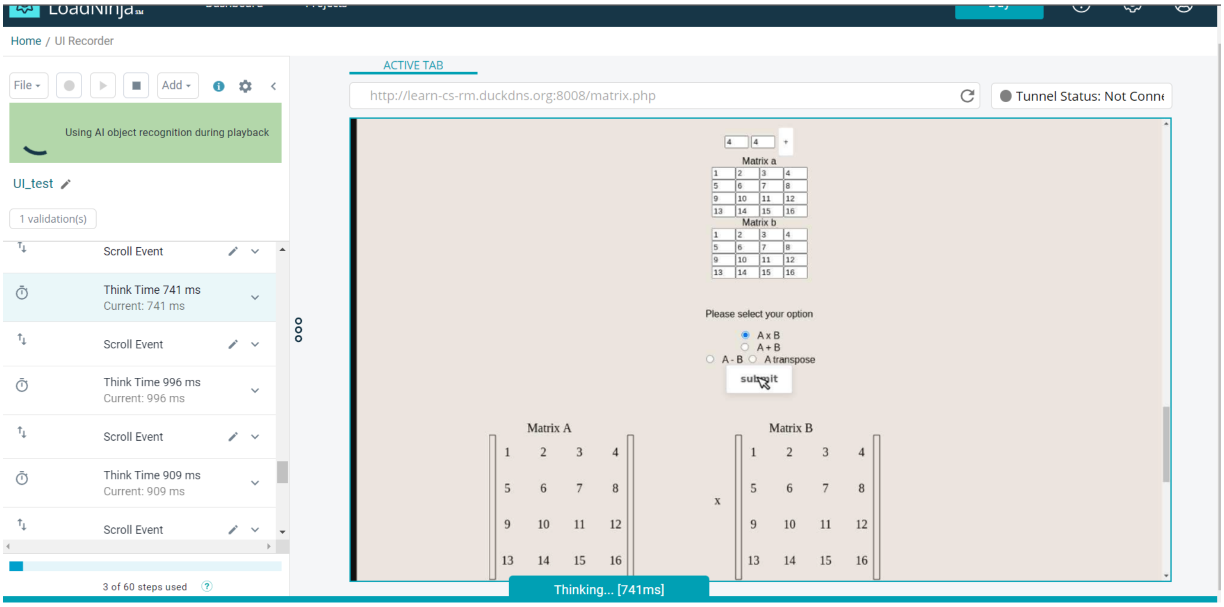
W3Schools (no date), ‘W3schools free online web tutorials (no date) w3schools online web tutorials’. (April 3 , 2023).
URL: *https://www.w3schools.com/*

Appendix A

An Appendix Chapter (Optional)

A.1 Testing screenshots

Figure A.1: Screenshot for the test script on LoadNinja.



Using AI object recognition during playback

UI_test

1 validation(s)

Scroll Event

Think Time 741 ms
Current: 741 ms

Scroll Event

Think Time 996 ms
Current: 996 ms

Scroll Event

Think Time 909 ms
Current: 909 ms

Scroll Event

3 of 60 steps used

http://learn-cs-rm.duckdns.org:8008/matrix.php

Tunnel Status: Not Conne

4 4

Matrix a

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Matrix b

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Please select your option

● A x B
○ A + B
○ A - B ○ A transpose

submit

Matrix A

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

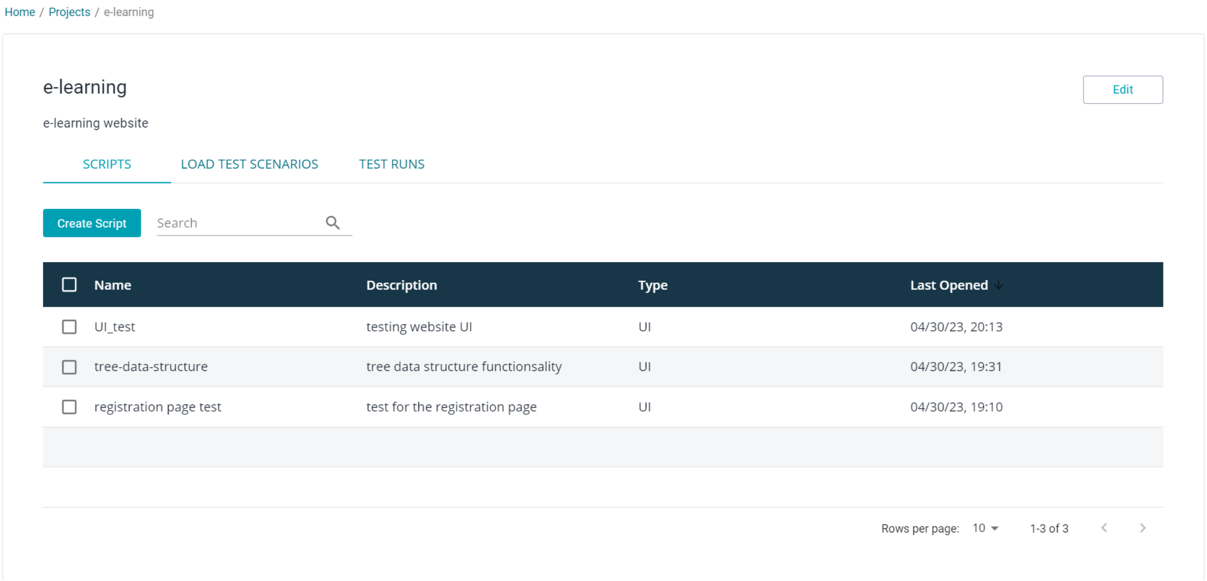
x

Matrix B

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Thinking... [741ms]

Figure A.2: Screenshot of all the test scripts on LoadNinja.



Home / Projects / e-learning

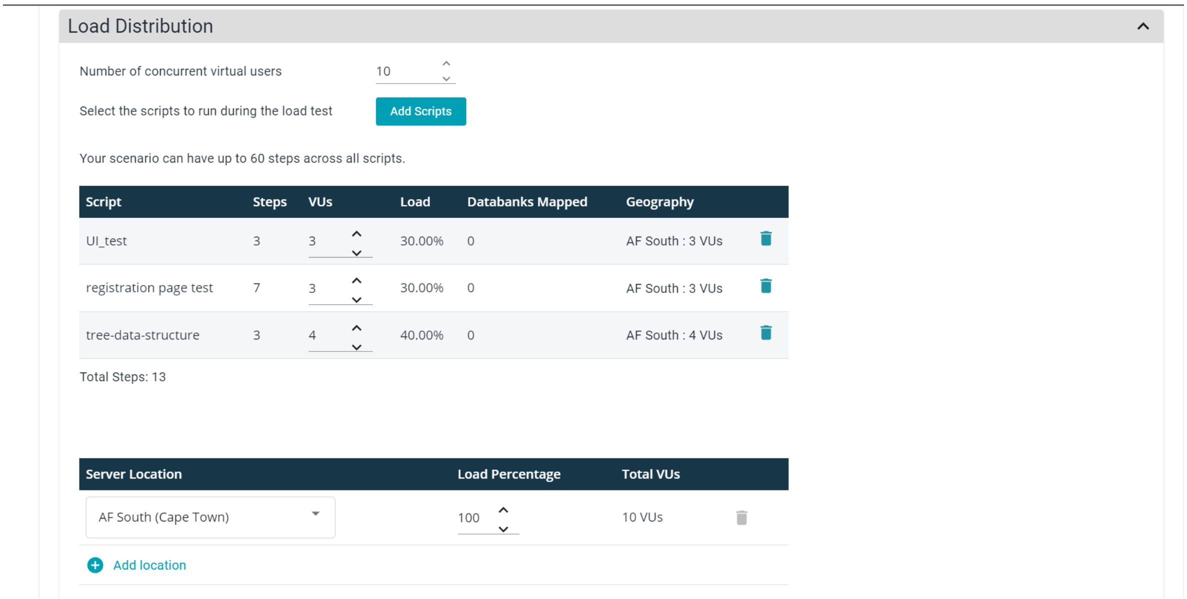
e-learning

e-learning website

□	Name	Description	Type	Last Opened
□	UI_test	testing website UI	UI	04/30/23, 20:13
□	tree-data-structure	tree data structure functionsality	UI	04/30/23, 19:31
□	registration page test	test for the registration page	UI	04/30/23, 19:10

Rows per page: 10 1-3 of 3

Figure A.3: Screenshot of test scenario on LoadNinja.



Load Distribution

Number of concurrent virtual users 10

Select the scripts to run during the load test

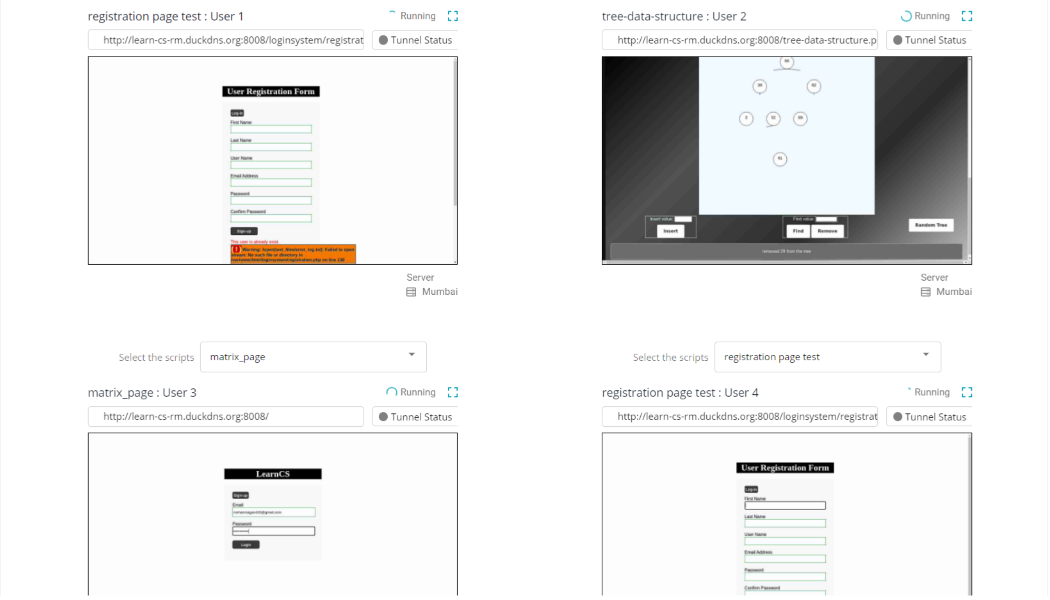
Your scenario can have up to 60 steps across all scripts.

Script	Steps	VUs	Load	Databanks Mapped	Geography
UI_test	3	3	30.00%	0	AF South : 3 VUs
registration page test	7	3	30.00%	0	AF South : 3 VUs
tree-data-structure	3	4	40.00%	0	AF South : 4 VUs

Total Steps: 13

Server Location	Load Percentage	Total VUs
AF South (Cape Town)	100	10 VUs

Figure A.4: Screenshot of test process



registration page test : User 1 — Running

http://learn-cs-rm.duckdns.org:8008/loginsystem/registrat — Tunnel Status

Server: Mumbai

tree-data-structure : User 2 — Running

http://learn-cs-rm.duckdns.org:8008/tree-data-structure.p — Tunnel Status

Server: Mumbai

Select the scripts: matrix_page

Select the scripts: registration page test

matrix_page : User 3 — Running

http://learn-cs-rm.duckdns.org:8008/ — Tunnel Status

registration page test : User 4 — Running

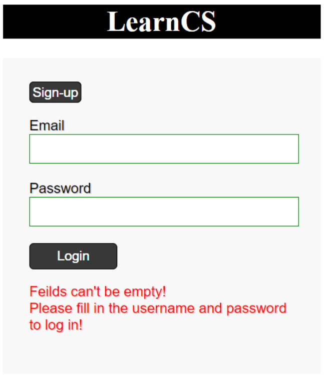
http://learn-cs-rm.duckdns.org:8008/loginsystem/registrat — Tunnel Status

This user is already exist.

Warning: fopen(text_files/error_log.txt): Failed to open stream: No such file or directory in /var/www/html/loginsystem/registration.php on line 236

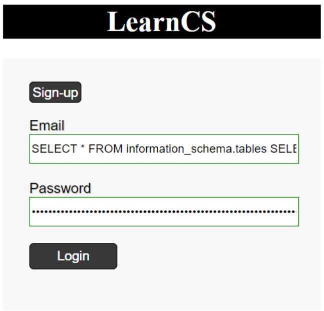
removed 29 from the tree

Figure A.5: Screenshot of testing empty input field



Feilds can't be empty!
Please fill in the username and password to log in!

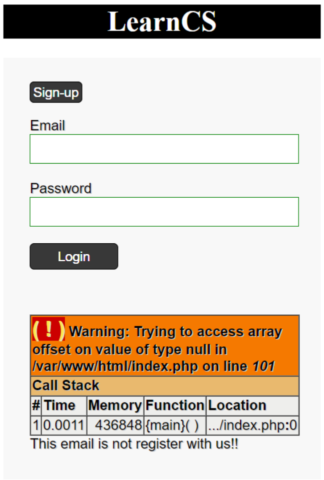
Figure A.6: Screenshot of testing SQL injection



Email: SELECT * FROM information_schema.tables SELE

Password: ••

Figure A.7: Screenshot of output of SQL injection

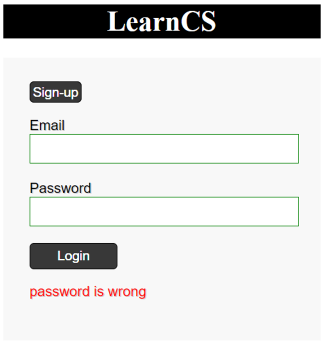


(!) Warning: Trying to access array offset on value of type null in /var/www/html/index.php on line *101*

Call Stack				
#	Time	Memory	Function	Location
1	0.0011	436848	{main}( )	.../index.php:0

This email is not register with us!!

Figure A.8: Screenshot of wrong password test

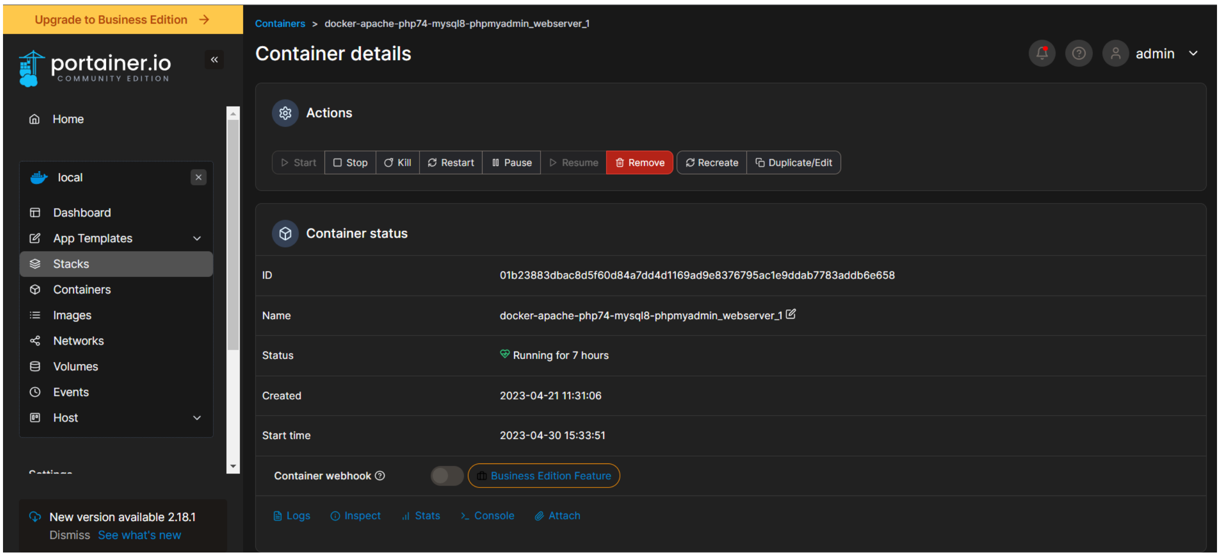


password is wrong

A.2 server setup

$$\textbf{Sever setup screenshots} \tag{A.1}$$

Figure A.9: Screenshot of Apache

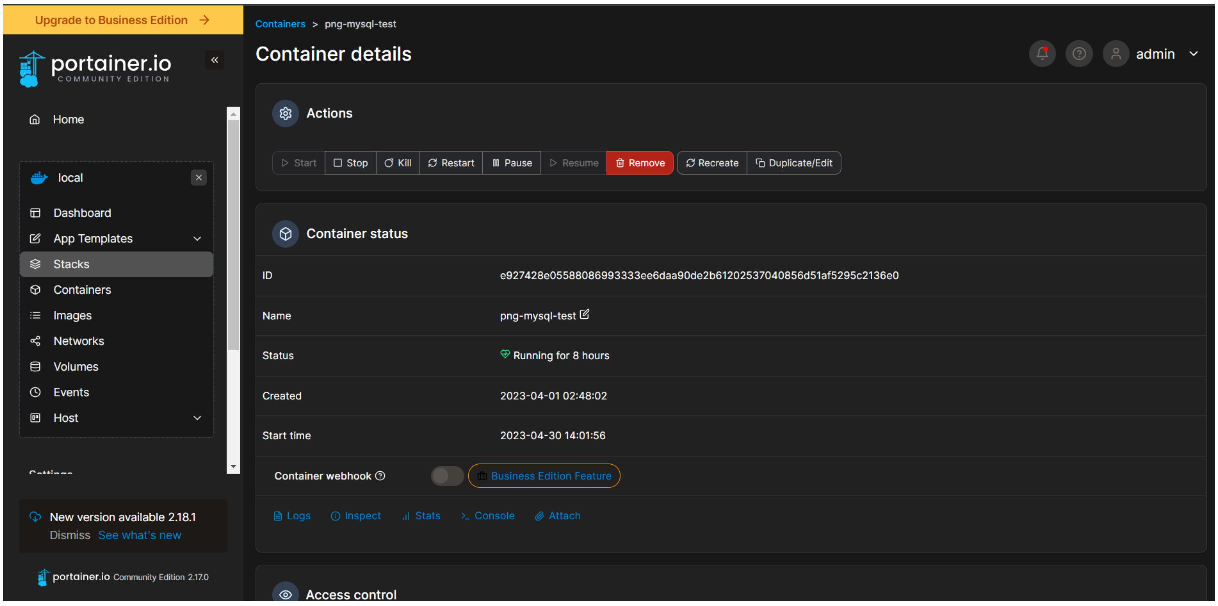


Container details

Container status

ID	01b23883dbac8d5f60d84a7dd4d1169ad9e8376795ac1e9ddab7783addb6e658
Name	docker-apache-php74-mysql8-phpmyadmin_webserver_1
Status	Running for 7 hours
Created	2023-04-21 11:31:06
Start time	2023-04-30 15:33:51
Container webhook	

Figure A.10: Screenshot of MySQL database

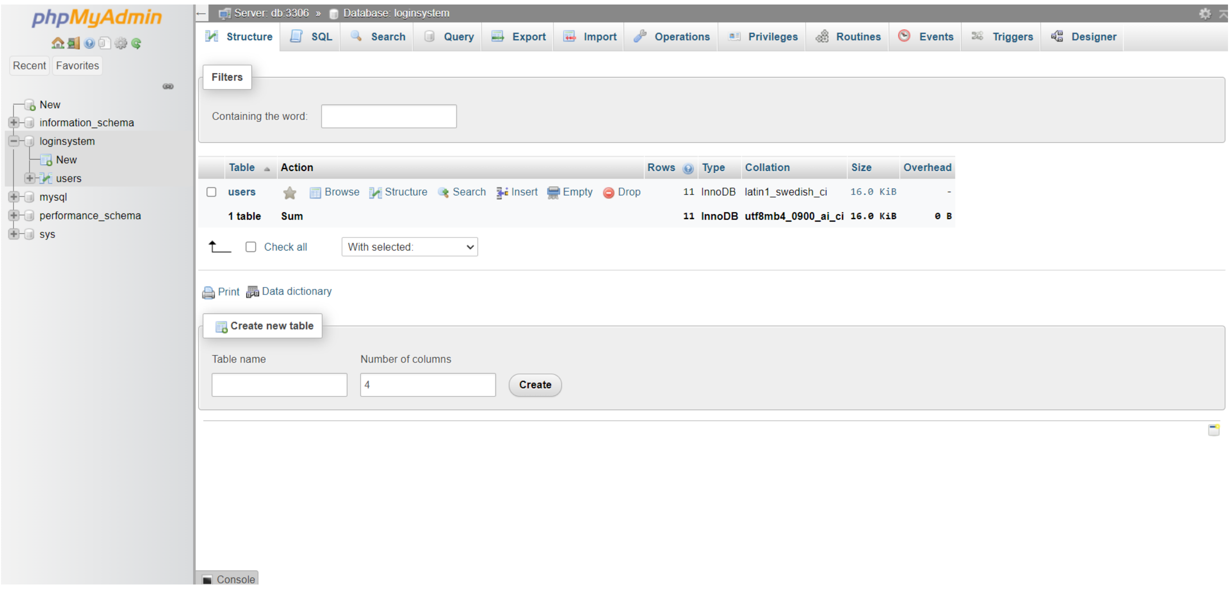


Container details

Container status

ID	e927428e05588086993333ee6daa90de2b61202537040856d51af5295c2136e0
Name	png-mysql-test
Status	Running for 8 hours
Created	2023-04-01 02:48:02
Start time	2023-04-30 14:01:56
Container webhook	

Figure A.11: Screenshot of test MyPhpAdmin



Containing the word:

	Table	Action	Rows	Type	Collation	Size	Overhead
□	users	Browse Structure Search Insert Empty Drop	11	InnoDB	latin1_swedish_ci	16.0 KiB	-
	1 table	Sum	11	InnoDB	utf8mb4_0900_ai_ci	16.0 KiB	0 B

□ Check all With selected:

Print Data dictionary

Table name

Number of columns 4