# Login page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the login page.

There are several challenges/difficulties that may arise during the development of a login page. Some of them include:

**Security**: One of the most critical challenges is ensuring the security of the login page. The login page is an essential entry point to the application, and if it is not secure, it can expose sensitive user data to attackers. Developers need to implement proper security measures, such as encryption, secure storage of passwords, and protection against attacks such as SQL injection and cross-site scripting.

**User experience**: The login page should be easy to use and navigate, as users may become frustrated if the process is overly complicated. The page should also be optimized for different devices and screen sizes.

Compatibility: The login page needs to work on different browsers and operating systems, which can be a challenge for developers. They need to test the page thoroughly to ensure that it works correctly across different platforms.

**Performance**: The login page should load quickly to prevent users from becoming impatient and abandoning the process. Developers need to optimize the page's performance to ensure it loads quickly, even on slower internet connections.

**Accessibility**: Developers should ensure that the login page is accessible to all users, including those with disabilities. They should adhere to accessibility standards such as WCAG 2.1 to ensure that the page is easy to use for everyone.

**User validation**: Developers need to ensure that the user's credentials are valid before granting access to the application. They should use secure and reliable methods to authenticate users, such as two-factor authentication or biometric authentication.

**Maintenance**: The login page may need to be updated over time to address security vulnerabilities or add new features. Developers need to ensure that the code is maintainable and easy to update over time.

Graphical user interface

Description automatically generated with medium confidence

1. The second step is to discuss the **SOLUTION** for the development of the login page.

Regarding the solution for the development of the login page, here are a few suggestions:

**Hash the Password**: Storing passwords in plain text is never recommended as it poses a significant security risk. Therefore, it is recommended to hash the password before storing it in the database. We can use the PHP password\_hash() function for this purpose.

**Use Prepared Statements**: The provided code is susceptible to SQL injection attacks as it directly inserts user input into the SQL query. Therefore, it is recommended to use prepared statements with placeholders to prevent SQL injection attacks.

**Use HTTPS**: It is recommended to use HTTPS instead of HTTP for transmitting sensitive information like passwords. You can obtain an SSL certificate and configure your server to use HTTPS.

Implement Captcha: It is also recommended to implement a captcha system to prevent automated bots from attempting to log in to the system.

**Proper Error Messages**: In the provided code, only one error message is shown for both invalid username and password. It is recommended to provide different error messages for invalid username and invalid password to give the user a clear indication of what went wrong.

**User Feedback**: It is a good practice to provide feedback to users, such as displaying a loading animation after form submission to show that the login process is ongoing and showing a message upon successful login or registration.

**Secure Session Handling**: Session hijacking is a common attack on web applications. Therefore, it is recommended to use secure session handling techniques, such as setting a unique session ID, expiring sessions after a certain period of inactivity, and using HTTPS for transmitting session data.

**Validate User Input**: It is recommended to validate user input before using it in the application. For instance, check whether the entered username and password match the specified criteria (e.g., minimum length, allowed characters, etc.) before using them in the login process.

Text

Description automatically generated

Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the login page.

The following tools and libraries are being used for the login page:

**PHP**: The code is written in PHP, which is a server-side scripting language that is commonly used for web development. It is used to connect to the database, retrieve user information, and handle form submissions.

MySQL: MySQL is a popular relational database management system, and it is being used in this code to store user information, such as usernames and passwords.

**Bootstrap**: Bootstrap is a popular CSS framework that is used to make responsive and mobile-first websites. It provides a set of pre-designed CSS and JavaScript components that can be used to create a consistent and professional-looking interface.

**jQuery**: jQuery is a fast and lightweight JavaScript library that simplifies HTML document traversing, event handling, and Ajax interactions for rapid web development. It is being used in this code to handle form submissions and to load external JavaScript files.

Links of tools used:

<https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css>

<https://code.jquery.com/jquery-3.3.1.slim.min.js>

<https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js>

<https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js>

# Register page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the register page.

Some of the challenges/difficulties that can arise during the development of a registration page include:

**Input validation**: The script needs to validate user input to ensure that the data entered is in the correct format and meets specific requirements, such as minimum password length, valid email address, etc. In this script, input validation is done using regular expressions and the filter\_var() function.

**Handling errors**: When a user submits a form with invalid data, the script needs to display error messages indicating what went wrong. In this script, error messages are displayed using the invalid-feedback class and the $name\_err, $email\_err, etc. variables.

**Security**: It is important to ensure that the registration page is secure and does not allow malicious users to exploit vulnerabilities. One way to enhance security is to use a CAPTCHA or other anti-spam measures. This script uses htmlspecialchars() function to prevent Cross-Site Scripting (XSS) attacks and move\_uploaded\_file() function to move the uploaded files.

**Design**: The registration page should be visually appealing and easy to use. This script uses Bootstrap CSS framework to create a responsive and visually appealing design.

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the register page.

**Keep it simple**: The registration process should be as simple and streamlined as possible. Avoid asking for too much information upfront, and only require the necessary details for creating an account.

**Use clear and concise language**: Make sure the language used on the registration page is clear and easy to understand. Avoid technical jargon or complex terms that could confuse users.

**Use a clean and modern design**: The registration page should be visually appealing and easy to navigate. Use a clean and modern design with clear typography and plenty of whitespace.

**Provide helpful feedback**: When users enter information into the registration form, provide feedback in real-time to let them know if there are any errors or missing information.

**Offer social media integration**: Consider allowing users to register using their social media accounts, such as Facebook or Google. This can make the registration process faster and more convenient for users.

Consider using multi-step forms: If you need to collect a lot of information during registration, consider breaking up the form into multiple steps. This can make the process feel less overwhelming and more manageable for users.

**Provide a clear call-to-action**: Make sure the register button stands out on the page and clearly communicates what will happen when the user clicks it. For example, use a call-to-action like "Create Your Account" or "Join Now"

Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the register page.

**HTML/CSS**: These are the basic building blocks for creating web pages. HTML (Hypertext Markup Language) is used for creating the structure and content of the web page, while CSS (Cascading Style Sheets) is used for styling and formatting the page.

**Bootstrap**: Bootstrap is a popular front-end development framework that provides a set of pre-built UI components and styles for creating responsive and mobile-friendly web pages quickly. It also includes JavaScript plugins for adding interactive features to the page.

**jQuery**: jQuery is a JavaScript library that simplifies the process of working with the Document Object Model (DOM) and handling events. It provides a set of methods for selecting and manipulating HTML elements, as well as an event handling system for responding to user actions.

**Font Awesome**: Font Awesome is a library of scalable vector icons that can be customized and styled using CSS. It provides a wide range of icons for different categories such as social media, business, and web applications.

**Google Fonts**: Google Fonts is a library of free and open-source fonts that can be used on websites. It provides a wide range of fonts with different styles and weights that can be easily integrated into the web page using CSS.

**PHP**: PHP (Hypertext Preprocessor) is a server-side scripting language that can be used for processing form data and interacting with databases. It provides a set of functions for handling data input, validation, and storage.

**Web addresses**:

HTML/CSS: <https://www.w3schools.com/html/> and <https://www.w3schools.com/css/>

Bootstrap: <https://getbootstrap.com/>

jQuery: <https://jquery.com/>

Font Awesome: <https://fontawesome.com/>

Google Fonts: <https://fonts.google.com/>

PHP: <https://www.php.net/>

## User Profile page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the user profile page.

One challenge that can arise during the development of the user profile page is ensuring that only authenticated users can access their own profile page. This can be addressed by implementing a session-based authentication system, as shown in the code snippet provided.

Another challenge could be properly displaying the user's information on the page. In the provided code, this is achieved by fetching the user's information from the database and using PHP to populate the input fields with the corresponding values.

Another potential challenge is ensuring that the user's profile picture is properly displayed. This can be accomplished by storing the user's profile picture on the server and using PHP to retrieve and display the image on the page. In the provided code, the image is displayed using an HTML img tag with the source attribute set to the path of the user's profile picture on the server.

Finally, a potential challenge could be providing a way for the user to edit their profile information. In the provided code, a button is included at the bottom of the page that redirects the user to an edit profile page where they can modify their information. The button uses JavaScript to navigate to the edit profile page when clicked.

Text

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the user profile page.

The solution for the development of the user profile page involved using PHP to connect to a MySQL database, retrieve the logged-in user's information, and display it on the page. The page also included buttons to edit the user's profile or log out of the system. The page was designed with Bootstrap and CSS to make it visually appealing and responsive. The use of sessions and authentication ensured that only authenticated users could access the page.

Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the user profile page.

**PHP**: A server-side scripting language used for building dynamic web pages and applications

**MySQL**: An open-source relational database management system used to store and retrieve data.

**Bootstrap**: A popular CSS framework used for creating responsive and mobile-first websites. The version used in this code snippet is 4.3.1, and the CSS file is included from a CDN: <https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css>

**jQuery**: A fast and concise JavaScript library used for simplifying HTML document traversal and manipulation, event handling, and Ajax interactions. The version used in this code snippet is 3.3.1, and the JavaScript file is included from a CDN: <https://code.jquery.com/jquery-3.3.1.slim.min.js>

**Popper.js**: A library used for positioning tooltips and popovers. The version used in this code snippet is 1.14.7, and the JavaScript file is included from a CDN: <https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js>

# Calendar Page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the calendar page.

There are several challenges that can arise during the development of a calendar page. Here are some possible difficulties that the developer might have encountered:

**Event Data Management**: Storing, retrieving and updating the event data can be complex, especially when the events can be edited dynamically. The code uses localStorage to store event data, but this approach can lead to performance issues if there are a lot of events.

**UI/UX Design**: Creating an intuitive and user-friendly interface for a calendar can be a challenge. The developer has to ensure that the users can easily add, edit and delete events without confusion.

**Compatibility**: Compatibility issues can arise between different browsers and devices. The developer has to ensure that the calendar page works seamlessly across different platforms.

**Performance**: The calendar page can become slow and unresponsive if there are too many events on the calendar. The developer has to optimize the code to ensure that the page remains performant even with a large number of events.

**Time Zone Management**: Handling time zones can be challenging, especially when dealing with events across different time zones. The developer has to ensure that the events are displayed in the correct time zone, and that the user can add events in their own time zone.

**Accessibility**: The calendar page must be accessible to all users, including those with disabilities. The developer has to ensure that the page meets accessibility standards, such as providing alt text for images and using semantic HTML.

**Security**: The calendar page can be vulnerable to security threats such as cross-site scripting (XSS) and SQL injection attacks. The developer has to ensure that the code is secure and that user input is validated and sanitized.

**Localization**: The calendar page may need to be localized for users in different countries. The developer had ensure that the page supports different languages and date formats.

Graphical user interface

Description automatically generated with low confidence

1. The second step is to discuss the **SOLUTION** for the development of the calendar page.

**Define the scope and requirements of the calendar page**: This involves identifying the purpose of the calendar page, the target audience, and the necessary features and functionalities that need to be implemented.

**Choose a suitable framework or tool**: There are many frameworks and tools available for developing a calendar page such as Google Calendar API, FullCalendar, or Bootstrap Calendar. The choice of framework or tool depends on the requirements and the platform on which the calendar page will be deployed.

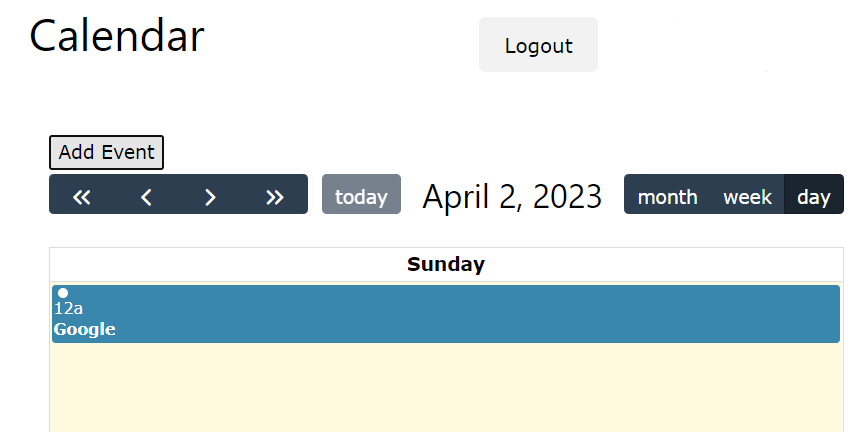
Design the user interface: This involves creating a layout and user interface design that is intuitive and easy to use. It's important to ensure that the design is responsive and mobile-friendly to allow users to access the calendar page on any device.

**Implement the functionalities**: This involves implementing the necessary functionalities such as adding events, displaying events in a calendar view, and allowing users to search and filter events based on various parameters.

**Test and deploy**: Once the calendar page is developed, it should be thoroughly tested to ensure that it functions as intended. After testing, the page can be deployed to the relevant platform, and any necessary updates or bug fixes can be made as needed.

Graphical user interface

Description automatically generated with medium confidence



1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the calendar page.

The calendar page uses FullCalendar, a JavaScript library for displaying and interacting with calendar data.

The version used in this code is 6.1.5, and the web address for this version is <https://cdn.jsdelivr.net/npm/fullcalendar@6.1.5/index.global.min.js>.

The page also uses FullCalendar's CSS file for styling the calendar, and the web address for the CSS file version used in this code is <https://cdnjs.cloudflare.com/ajax/libs/fullcalendar/3.10.2/fullcalendar.min.css>.

In addition to FullCalendar, the code also includes a PHP script (conn.php) for connecting to the database. The contents of the PHP script are not included in the code provided, so it is unclear what database management system is being used.

# Study Tracking/Study Diary/Web Development page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the Web development page.

Some challenges that may have been encountered during the development of this web development page are:

**Design**: Designing a good user interface that is easy to navigate and visually appealing can be a significant challenge. The page may have required multiple iterations before settling on a design that met the requirements.

**Responsiveness**: The page needs to work on different screen sizes and resolutions, including mobile devices. Achieving a responsive design that adjusts to different screen sizes and still looks great can be challenging.

Cross-browser compatibility: Different browsers may render web pages differently, which can lead to inconsistencies in the appearance and functionality of the page. Ensuring cross-browser compatibility can be a significant challenge.

**Database connectivity**: Connecting to the database to store and retrieve data can be challenging, especially when using different database management systems.

**Security**: Security concerns must be addressed, such as preventing unauthorized access to the system and protecting user data from hacking attempts.

**Debugging**: Debugging is an important aspect of web development. Finding and fixing errors in code can be challenging, especially when working with complex code.

**Performance**: Ensuring the page loads quickly and efficiently is essential to provide a good user experience. Optimizing the performance of the page can be a challenging task.

Graphical user interface

Description automatically generated with medium confidence

1. The second step is to discuss the **SOLUTION** for the development of the web development page.

The solution is written in PHP and HTML. The PHP code is used to include the database connection file, while the HTML code is used to create the structure and layout of the page.

The JavaScript code is used to add new tasks, courses, and save the task data to localStorage.

Text

Description automatically generated

Graphical user interface, application

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the web development page.

The following tools and libraries were used in the web development page:

PHP - a server-side scripting language used to develop dynamic web pages. It was used to include the 'conn.php' file in the page. PHP is a free and open-source software available at <https://www.php.net/>.

W3.CSS - a CSS framework that provides a collection of CSS styles and components for building responsive and modern web pages. It was used to style the navigation bar and some elements on the page. W3.CSS is free to use and available at <https://www.w3schools.com/w3css/4/w3.css>.

JavaScript - a client-side scripting language that allows creating dynamic and interactive web pages. It was used to add functionality to the page, such as adding a new sub-subsector and adding new tasks dynamically. JavaScript is a free and open-source software available at <https://www.javascript.com/>.

No additional libraries or frameworks were used in the web development page.

# Study Tracking/My Grades page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the my grades page.

Some potential challenges or difficulties that could be encountered during the development of the my grades page include:

**Database connectivity issues**: The page seems to include a reference to a conn.php file, which suggests that it is interacting with a database. Connectivity issues with the database could result in errors or incomplete data being displayed on the page.

**User authentication and authorization**: The page appears to be part of a larger student portal, which likely requires user authentication and authorization. Ensuring that only authorized users can access the page and view their own grades could be a challenge.

**Data processing and presentation**: Depending on the complexity of the grading system and the amount of data to be displayed, processing and presenting the information in a clear and user-friendly way could be a significant challenge. Careful consideration of the layout and design of the page will be important to ensure that users can easily understand and interpret their grades.

**Handling user input**: The page includes a button to add a new year, which presumably requires user input. Validating and processing user input can be a source of errors and security vulnerabilities, so careful attention must be paid to this feature.

**Accessibility**: Ensuring that the page is accessible to users with disabilities or impairments may require additional effort, particularly if complex data visualization or interactive features are included.

**Compatibility**: Ensuring that the page is compatible with a range of devices and browsers may require additional testing and development effort.

Graphical user interface, application

Description automatically generatedTable

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the my grades page.

Added a function that allows the user to add Years to the subsection of side navigation panel.

Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the my grades page.

The code provided is a PHP script that generates the HTML markup for a web page that allows the user to select a year and view their grades for that year. The following tools and libraries are used in the code:

**PHP**: The code is written in PHP, a popular server-side scripting language used for web development.

**MySQL**: The code uses a MySQL database to store the user's grades and other information. The database connection is established using the 'conn.php' file, which likely contains the necessary credentials and configuration settings.

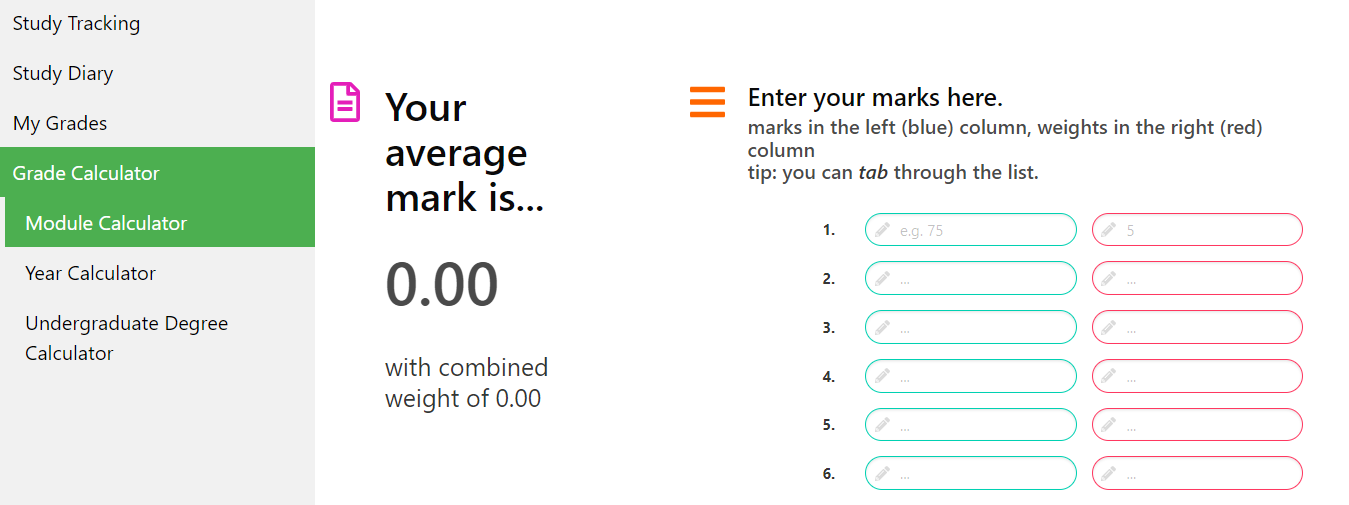
**W3.CSS**: The code uses the W3.CSS framework to style the web page. W3.CSS is a lightweight and responsive CSS framework developed by W3Schools.

**JavaScript**: The code uses JavaScript to add a new year option to the year selection menu when the user clicks the '**+ Add Year'** button. The function **'addSubSubSector**()' achieves this functionality.

# Study Tracking/Grade Calculator/Module Calculator page

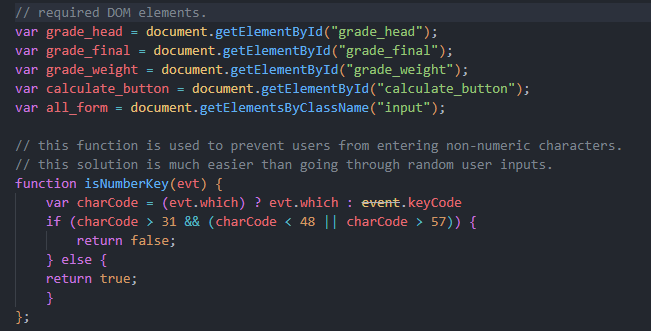
1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the module calculator page.

Calculation and UI/UX was difficult to develop.



1. The second step is to discuss the **SOLUTION** for the development of the module calculator page.

Hardcoded for the table and take marks. Then used separated script file to define many functions to calculate.



Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the module calculator page.

The module calculator page uses the following tools and libraries:

**W3.CSS** (<https://www.w3schools.com/w3css/>): a CSS framework that provides a set of styles, components, and JavaScript plugins to build responsive web pages.

**Bulma** (<https://bulma.io/>): a modern CSS framework based on Flexbox that provides a comprehensive set of customizable UI components and utilities to build responsive web interfaces.

**Font Awesome** (<https://fontawesome.com/>): a popular icon set and toolkit that provides a collection of scalable vector icons and CSS tools to add icons to web projects.

**gradecalc.js** (custom script): a JavaScript file that contains custom code to implement the functionality of the module calculator page, such as calculating the module grade based on the input data.

# Study Tracking/Grade Calculator/Year Calculator page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the year calculator page.

Some possible challenges/difficulties that might have been faced during the development of this year calculator page are:

**Validating user input**: The form used for calculating the overall weighted grade requires input from the user, such as the module name, grade, and credit. The form needs to be validated to ensure that the user enters valid input, and the system does not process incorrect or incomplete data.

**Calculation logic**: The calculation logic for computing the overall weighted grade and degree classification might be complex and require several conditions and mathematical formulas. Ensuring that the calculations are accurate and error-free can be a challenging task.

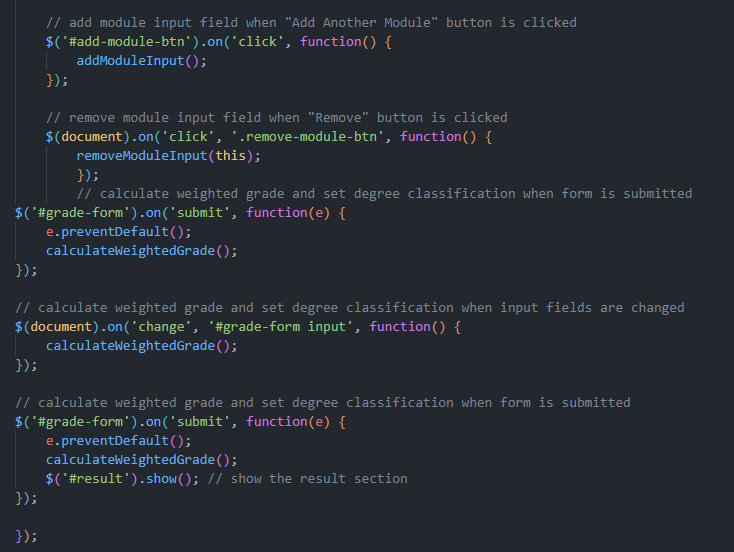
**Integration with other modules**: The year calculator page is part of a larger student portal system that includes other modules such as study tracking and fitness tracking. Integrating this page with other modules and ensuring that the system functions smoothly can be a complex task.

**User interface design**: The user interface design of the year calculator page must be intuitive and easy to use. Designing an interface that is both aesthetically pleasing and functional can be a challenge.

**Debugging and testing**: Debugging and testing the year calculator page to ensure that it is working correctly can be a challenging task, especially when there are many variables and conditions involved in the calculation logic.

Text

Description automatically generated

Text

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the year calculator page.

Here are some points regarding the code:

The code uses jQuery, a popular JavaScript library, to manipulate the HTML and handle events.

The code defines three functions: addModuleInput(), removeModuleInput(), and calculateWeightedGrade(), to add input fields, remove input fields, and calculate the weighted grade respectively.

The code also defines a setDegreeClassification() function to set the degree classification based on the weighted grade.

The code binds event listeners to the "Add Another Module" button, "Remove" button, and input fields, to trigger the appropriate functions when certain events occur.

The code prevents the form from submitting when the submit button is clicked, by using e.preventDefault().

Graphical user interface, text, application

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the year calculator page.

The tools and libraries used in the year calculator page are:

**HTML**: Hypertext Markup Language is used for creating the structure and content of the webpage.

**CSS**: Cascading Style Sheets are used for styling the webpage.

**PHP**: Hypertext Preprocessor is used as a server-side programming language to handle form data and interact with the database.

**jQuery**: A fast and feature-rich JavaScript library is used for DOM manipulation, event handling and AJAX requests.

**Bootstrap**: A popular front-end framework is used for responsive design and UI components.

**MySQL**: A relational database management system is used for storing and retrieving data.

**Font Awesome**: A font and icon toolkit is used for adding scalable icons to the webpage.

The page also includes custom JavaScript and CSS files named **'yr.js'** and **'yr.css'**, respectively. The 'yr.js' file is used for adding dynamic behaviour to the page, while 'yr.css' is used for custom styling.

# Study Tracking/Grade Calculator/Undergraduate Degree Calculator page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the undergraduate degree calculator page.

Some of the challenges and difficulties that may have been encountered during the development of this undergraduate degree calculator page could include:

**Input validation, Calculations, User interface design, Database integration, Security**

Text

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the undergraduate degree calculator page.

The solution provided is a JavaScript code that performs two main functions. The first function is to calculate the overall weighted grade of an undergraduate degree based on the grades and weightings of each year of the degree program. The second function is to add another year input to the form.

The code first defines a function called calculateWeightedGrade(), which calculates the overall weighted grade of the degree program. It does this by first selecting all the input fields with class name "year-input" using querySelectorAll(). It then loops through each input field and retrieves the year grade and year weighting values. These values are used to calculate the total weighting and weighted grade sum, which are then used to calculate the overall weighted grade of the degree program.

The code then updates the UI by setting the inner text of an element with ID "weighted-grade" to the calculated overall weighted grade. It also calculates and updates the degree classification by setting the inner text of an element with ID "degree-classification" based on the overall weighted grade calculated.

The code also defines a function called addYearInput(), which adds another year input to the form. It does this by first selecting all the input fields with class name "year-input" using querySelectorAll(). It then creates new input elements for year name, year grade, and year weighting using the createElement() method. It sets the attributes and class names for these elements, and then appends them to a new year input container using the appendChild() method. Finally, it appends the new year input container to the existing year inputs container in the form.

The code adds an event listener to the form using the addEventListener() method, which listens for the "submit" event and prevents the default behavior using event.preventDefault(). It then calls the calculateWeightedGrade() function to calculate and update the overall weighted grade and degree classification initially.

Overall, the code provides a solution for developing an undergraduate degree calculator page that can calculate the overall weighted grade and degree classification of a degree program based on the grades and weightings of each year of the program. It also allows users to add another year input to the form.

Graphical user interface, application

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the undergraduate degree calculator.

The tools and libraries used in the undergraduate degree calculator are:

**PHP**: A server-side scripting language used to develop the dynamic web page and connect with the database. It is used in this code for including the connection file (**conn.php**) and interacting with the database to save the user's input.

**MySQL**: A relational database management system used to store and manage data. It is used in this code for storing the user's input and retrieving it to calculate the degree classification.

**W3.CSS**: A CSS framework used to style the web page. It is used in this code for applying the predefined styles to the web page elements and making the layout responsive.

**JavaScript**: A client-side scripting language used to add interactivity to the web page. It is used in this code for adding event listeners to the buttons and calculating the weighted grade and degree classification based on the user's input. (**deg.js**)

# Fitness Tracking/Exercise Diary/Monday

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the page.

There are several challenges/difficulties that could be identified during the development of the code:

**Incomplete code**: the provided code seems to be incomplete and lacks some essential parts that may not have been included in this code snippet. As a result, some parts of the code may not function correctly.

**Security issues**: the use of PHP\_SELF in the form action is susceptible to cross-site scripting (XSS) attacks. A hacker could use this vulnerability to inject malicious code into the form and steal sensitive data.

**Input validation**: the code does not validate user input, making it vulnerable to SQL injection attacks. A malicious user could use this vulnerability to manipulate the SQL query and access or modify sensitive data.

**Code structure**: the code structure could be improved by using a separate file for the database connection and separating the PHP code from the HTML code. This would improve the readability and maintainability of the code.

**User interface**: the user interface could be improved to make it more user-friendly and visually appealing. This could be achieved by using CSS to style the page and adding more interactive features to the page.

Text

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the page.

The script is intended to be used as part of a larger system or application, it may need to be integrated with other components or functions. Additionally, it may need to be modified or updated to accommodate changes in the application or user requirements.

If the script is meant to stand alone and function as a standalone tool, it may need to be enhanced with additional features or functionality to make it more user-friendly and efficient. For example, it could include error handling to ensure that input data is valid and within appropriate ranges, or it could be modified to store data in a database for future reference.

Table

Description automatically generated with medium confidence

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the Monday page.

The following tools and libraries are used in the code:

**PHP**: This is a popular programming language used for web development, particularly for server-side programming. PHP is used to process the form data entered by the user in the code.

**HTML**: This is the markup language used to create the structure and content of web pages. HTML is used to define the layout and content of the web page in the code.

**CSS**: This is the language used to style the appearance of web pages. CSS is used to define the style of the web page, including the fonts, colors, and layout.

**W3.CSS**: This is a CSS framework developed by W3Schools that provides a set of CSS classes to style web pages quickly and easily. The code uses the W3.CSS framework to style the web page elements.

**MySQL**: This is an open-source relational database management system used to store and retrieve data. The code includes a connection to a MySQL database using the "conn.php" file, which is used to store the user data entered in the form.

# Fitness Tracking/Food Diary/Daily Nutrition Goals page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the daily nutritional goals page.

Regarding the challenges/difficulties during the development of the daily nutritional goals page, there could be several possible issues. Here are some examples:

**Defining nutritional goals**: Before developing the page, it is necessary to define the nutritional goals that the user wants to achieve, such as calorie intake, macronutrient ratio, and micronutrient requirements. Determining these goals may require consulting a nutritionist or using a reliable source of nutritional information.

**Data collection**: To accurately track food intake, the application needs to collect data on the user's meals and snacks. This can be challenging, as users may not have the necessary information (e.g., exact portion sizes, ingredients, etc.) or may forget to log certain foods.

**Database management**: Storing and retrieving user data is a critical component of the application. Therefore, database management is an essential aspect of the development process. Ensuring data security, handling data backups, and optimizing database queries are some of the challenges that developers may encounter.

User interface design: The user interface should be easy to navigate and visually appealing to encourage users to engage with the application. Designing an intuitive interface that provides relevant information to the user can be a challenge.

**Mobile compatibility**: Since users may want to access the application from their smartphones, ensuring that the application is mobile-friendly is essential. This may require additional development efforts, such as optimizing page layout, font sizes, and images for smaller screens.

Timeline

Description automatically generated with medium confidence

1. The second step is to discuss the **SOLUTION** for the development of the daily nutritional goals page.

The code is a PHP file that contains an HTML form for adding food items to a fitness tracking system. The form collects the name, quantity, kcal, notes, and time for a food item, and when the "Add" button is clicked, it saves the data to a JSON file named "storage.json".

The solution for the development of the daily nutritional goals page could involve modifying this code to include additional features such as:

Retrieving the data from "storage.json" and calculating the total kcal consumed for the day.

Comparing the total kcal consumed with a target value and displaying a message indicating whether the target has been met.

Providing suggestions for food items to consume to meet the target, based on the user's preferences and dietary restrictions.

Allowing the user to set their target value and view their progress towards it over time.

Providing visual representations of the user's daily nutritional goals and progress towards them, such as charts or graphs.

Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the daily nutritional goals page.

.

**PHP**: This is a popular programming language used for web development, particularly for server-side programming. PHP is used to process the form data entered by the user in the code.

**HTML**: This is the markup language used to create the structure and content of web pages. HTML is used to define the layout and content of the web page in the code.

**CSS**: This is the language used to style the appearance of web pages. CSS is used to define the style of the web page, including the fonts, colors, and layout.

**W3.CSS**: This is a CSS framework developed by W3Schools that provides a set of CSS classes to style web pages quickly and easily. The code uses the W3.CSS framework to style the web page elements.

**MySQL**: This is an open-source relational database management system used to store and retrieve data. The code includes a connection to a MySQL database using the **"conn.php"** file, which is used to store the user data entered in the form.

# Fitness Tracking/Food Diary/Calories by Meals page

1. The first step is to discuss the **CHALLENGES/DIFFICULTIES** during the development of the calories by meals page.

One potential challenge with the development of the calories by meals page could be the organization and structuring of the data. Since this page is designed to track daily meal intake and nutritional information, it requires proper categorization of food items based on meals, such as breakfast, lunch, dinner, and snacks. This requires the use of data structures that can store and organize the data efficiently.

Another challenge could be the calculation of the consumed data, such as calories, carbohydrates, protein, and fats, based on the user's input. This requires the use of mathematical functions to accurately calculate the total consumed data and display it in an understandable format.

Furthermore, ensuring the security of user data is crucial, as this page deals with sensitive information about the user's health and dietary habits. Thus, appropriate security measures such as input validation, data encryption, and secure storage should be implemented to prevent unauthorized access to user data.

Lastly, designing an intuitive and user-friendly interface can be challenging, as this page needs to display a large amount of data in an organized and easily understandable manner. This requires the use of proper UI/UX design principles and extensive testing to ensure that the page is user-friendly and meets the user's expectations.

A screenshot of a computer

Description automatically generated

1. The second step is to discuss the **SOLUTION** for the development of the calories by meals page.

Save records to a file. And for calculating data first retrieve target data from database which was saved in daily nutrition goals page. Then calculate all the consumed data, then calculate the remaining data.

Text

Description automatically generated

1. The third step is to discuss the **TOOLS & LIBRARIES** used in for the calories by meals system.

The code provided is written in PHP, a server-side scripting language that is used for web development. The system is implemented using the following tools and libraries:

**W3.CSS**: W3.CSS is a CSS framework developed by W3Schools that is used to create responsive and mobile-first websites. It is used in this system to style the HTML components of the web page.

**JSON**: JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate. In this system, JSON is used to store and retrieve meal information such as name, serving size, calories, carbs, protein, and fats.

**File functions**: The PHP file functions such as file\_get\_contents and json\_decode are used to read and parse the JSON files that store the meal information.

**Foreach loop**: A foreach loop is used to iterate over the meal information stored in the JSON files and extract the calorie, carbs, protein, and fats information of each meal. These values are then summed up to calculate the total consumed calories, carbs, protein, and fats.

Overall, these tools and libraries help to create a responsive and interactive user interface and enable the system to store, retrieve, and manipulate meal information in an organized and efficient manner.