Set of instructions for group creation and project allocation for FEE-I subject

1. Group Formation

- **Group Size:** Each group should consist of 3 members. Students are encouraged to choose their group members based on complementary skills and mutual interest in the project topics.
- **Group Formation Deadline:** Groups must be formed and submitted by **02.09.2024**. Failure to do so will result in random group assignment.

2. Project Topic Selection

- **Project Topic Options:** Groups may choose their project topic from the provided list or propose their own topic. If proposing a new topic, it must be relevant to the FEE subject and receive approval from the subject teacher before the deadline.
- **Topic Selection:** To ensure diversity and avoid redundancy, a maximum of 2 groups can select the same project topic from the provided list or a similar proposed topic. Topic allocation will be on a first-come, first-served basis. If a group's topic has already been taken by two other groups, they will be required to select a new topic.
- **Topic Approval:** If a group chooses to propose their own topic, they must submit a brief description and objectives for approval by **09.09.2024.**
- **Topic Selection Deadline:** Groups must finalize and submit their chosen or proposed topic by **09.09.2024** to their subject teacher. Topics cannot be changed after this deadline.

3. Project Development and Submission

- **Project Work:** Groups are expected to work collaboratively on their projects, ensuring equal contribution from all members.
- **Continuous Evaluations:** There will be continuous evaluations where groups must submit brief progress reports. These will occur from **CE1: 30.09.2024 - 04.10.2024 and CE2: 04.11.2024 - 08.11.2024.**
- **Final Submission:** The final project report and any deliverables must be submitted by **25.11.2024.**

4. Evaluation

Evaluation Criteria: Projects will be evaluated based on the rubrics for each Continuous Evaluation and Final Project Evaluation.

Ensure that all deadlines are strictly adhered to for a smooth project workflow.

List of Project Topics

Dear Students

This is not just a simple list of projects, but a collection that describes each project in enough detail so that you can develop it from the ground up!

For each project, following **features need to be considered**:

1. A clear and descriptive objective;
2. A list of **bonus features** that not only improve the base project, but also your skills at the same time
3. All the resources and links to help you find what you need to complete the project

1. Welcome Page Design

Objective: Create an attractive welcome page for a website with a clear message and introduction to the content.

Approach:

- Design a simple HTML structure with relevant headings and text.
- Apply CSS styles to enhance the visual appeal, like background colors, fonts, and spacing.
- Add images or icons to complement the message.
- Utilize media queries for responsiveness on different devices.

Bonus Features:

- Implement smooth scrolling to navigate to different sections.
- Add a subscription form for users to sign up for updates.
- Integrate social media share buttons.

2. Blog Page Development

Objective: Build a blog page with organized content, images, and links for an engaging user experience.

Approach:

- Structure the HTML with appropriate headings and sections.
- Use CSS to create an aesthetically pleasing layout and style the blog content.
- Add images to each blog post and optimize them for faster loading.
- Include anchor tags for smooth navigation between blog posts.
- Use semantic tags to enhance accessibility and SEO.

Bonus Features:

- Implement a search bar to allow users to find specific blog posts.
- Add pagination to handle multiple blog posts.
- Enable commenting functionality for users to interact with the blog posts.
- Integrate a "Read More" button for lengthy posts.

3. Responsive Portfolio

Objective: Create a responsive portfolio website showcasing projects and skills.

Approach:

- Design the HTML structure for the portfolio with sections for projects, skills, and about me.
- Utilize CSS media queries to adapt the layout for different screen sizes.
- Apply Flexbox or Grid for arranging project cards.
- Use JavaScript to add interactivity and animations for a more engaging experience.

Bonus Features:

- Implement a filter option to sort projects by category or technology.
- Add a lightbox gallery for displaying project images in a popup.
- Create a progress bar for showcasing skill levels.
- Incorporate a contact form for potential clients or employers.

4. Interactive Quiz Application

Objective: Develop an interactive quiz application with multiple-choice questions and instant feedback.

Approach:

- Create the quiz questions and options in an array or JSON format.
- Use HTML for the quiz layout and radio buttons for options.
- Write JavaScript to handle user selections, validate answers, and calculate scores.
- Display the score at the end of the quiz.

Bonus Features:

- Add a timer to each question, making the quiz more challenging.
- Integrate a progress bar to show the user's progress through the quiz.
- Include a "Hint" button for users who need help with a difficult question.
- Save high scores using local storage.

5. Personal Resume Page

Objective: Build a personal resume page highlighting skills, experience, and achievements.

Approach:

- Design the resume structure using HTML with sections for education, work experience, and skills.
- Apply CSS styles to present the resume in a professional format.
- Use CSS media queries to make the resume responsive on different devices.

Bonus Features:

- Add a timeline to showcase work experience chronologically.
- Incorporate a skills graph to visually represent proficiency levels.
- Implement a PDF download option for users to save the resume.
- Include a contact form for potential employers to get in touch.

6. Image Gallery with Filters

Objective: Create an image gallery with filter options based on categories or tags.

Approach:

- Design the HTML structure for the gallery with image containers and tags.
- Use CSS Grid or Flexbox to arrange the images in a visually appealing manner.
- Write JavaScript to handle filter options and show/hide images accordingly.

Bonus Features:

- Add a lightbox for enlarging images and providing additional details.
- Implement pagination for better user experience with a large number of images.
- Include image upload functionality for users to contribute to the gallery.

7. Weather Forecast App

Objective: Develop a weather forecast application that displays the current weather and a 5-day forecast.

Approach:

- Utilize HTML to structure the app layout with sections for weather information.
- Use CSS to style the app and make it visually appealing.
- Use a weather API to fetch real-time weather data based on the user's location or search.

Bonus Features:

- Display weather icons corresponding to the weather conditions (sunny, cloudy, rainy, etc.).
- Add a search bar for users to enter specific locations and get weather data.
- Provide temperature conversions (e.g., Celsius to Fahrenheit) based on user preferences.

8. To-Do List Manager

Objective: Build a to-do list manager that allows users to add, edit, and delete tasks.

Approach:

- Design the HTML structure for the to-do list with input fields and buttons.
- Use CSS to create a clean and user-friendly interface.
- Write JavaScript to handle adding, editing, and deleting tasks in the list.

Bonus Features:

- Implement local storage to save tasks even after the browser is closed.
- Add priority levels to tasks (e.g., high, medium, low).
- Allow users to mark tasks as completed and view completed tasks separately.
- Provide options to sort tasks by due date or priority.

9. Memory Card Game

Objective: Develop a memory card matching game where players need to find matching pairs.

Approach:

- Create the HTML structure for the card game with face-down cards.
- Use CSS to style the cards and provide visual feedback on card flips.
- Write JavaScript to handle card flipping, matching logic, and game state.

Bonus Features:

- Add a timer to make the game more challenging.
- Include a score counter to keep track of the player's performance.
- Implement different difficulty levels with varying numbers of cards.
- Create a leaderboard to display high scores.

10. Countdown Timer App

Objective: Build a countdown timer app for setting reminders or events.

Approach:

- Design the HTML structure for the countdown timer with input fields for setting the time.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to handle the countdown functionality and update the timer in real-time.

Bonus Features:

- Include sound effects or notifications when the timer reaches zero.
- Allow users to save multiple countdowns with custom labels.
- Implement an option to repeat the countdown after it finishes.
- Provide a visual representation of the countdown progress (e.g., a progress bar).

Certainly! Here are 20 more project ideas for you:

11. Calculator Application

Objective: Create a basic calculator app capable of performing arithmetic operations.

Approach:

- Design the HTML layout with buttons for digits and operators.
- Use CSS to style the calculator and make it visually appealing.
- Write JavaScript to handle button clicks and perform calculations.

Bonus Features:

- Add keyboard support for users to input calculations.
- Implement scientific calculator functions like square root and exponentiation.
- Include a history feature to display past calculations.

12. Random Quote Generator

Objective: Build an app that displays random quotes or inspirational messages.

Approach:

- Design the HTML structure to display the quote and an option to fetch a new one.

- Use CSS to style the app and make it visually appealing.

- Write JavaScript to fetch random quotes from an API or a predefined list.

Bonus Features:

- Allow users to share quotes on social media platforms.

- Implement a feature to favorite or bookmark quotes.

- Add animations to enhance the quote transition.

13. BMI Calculator

Objective: Develop a Body Mass Index (BMI) calculator that determines a person's health status based on their weight and height.

Approach:

- Design the HTML structure with input fields for weight and height.

- Use CSS to create a simple and user-friendly interface.

- Write JavaScript to calculate the BMI and display the results.

Bonus Features:

- Add a visual representation of different BMI ranges using color codes.

- Include a feature to save and track BMI over time.

- Provide health tips based on the calculated BMI.

14. Countdown to a Special Event

Objective: Build an app that counts down to a specific date, like a birthday or holiday.

Approach:

- Design the HTML layout with an input field for the target date.

- Use CSS to style the app and make it visually appealing.

- Write JavaScript to calculate the time remaining and update the countdown.

Bonus Features:

- Add different themes or background images for various events.

- Include animations to celebrate when the countdown reaches zero.

- Allow users to set reminders for the upcoming event.

15. Virtual Keyboard

Objective: Create an on-screen virtual keyboard that allows users to type using their mouse or touchscreen.

Approach:

- Design the HTML structure with buttons for each letter and character.
- Use CSS to style the virtual keyboard and make it visually appealing.
- Write JavaScript to handle button clicks and simulate typing.

Bonus Features:

- Implement support for different keyboard layouts (e.g., QWERTY, AZERTY).
- Add an option for users to change the keyboard's appearance (color, size, etc.).
- Provide sound feedback for key presses.

16. Currency Converter

Objective: Develop a currency converter app that converts amounts between different currencies.

Approach:

- Design the HTML layout with input fields for the amount and currency selection.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to fetch currency exchange rates from an API and perform conversions.

Bonus Features:

- Allow users to select multiple currencies and perform conversions simultaneously.
- Implement a history feature to display past currency conversions.
- Add an option to save favorite currency pairs.

17. Recipe Finder

Objective: Build an app that allows users to search for recipes based on ingredients they have at hand.

Approach:

- Design the HTML structure with a search bar and result display area.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to fetch recipe data from a recipe API based on user input.

Bonus Features:

- Include options for filtering recipes based on dietary preferences (e.g., vegetarian, gluten-free).
- Allow users to create accounts and save favorite recipes.
- Implement a feature to generate shopping lists based on selected recipes.

18. Music Player

Objective: Create a simple music player that allows users to play audio files.

Approach:

- Design the HTML layout with play, pause, and volume control buttons.
- Use CSS to style the music player and make it visually appealing.
- Write JavaScript to handle audio playback and control.

Bonus Features:

- Implement a playlist feature to queue and play multiple audio tracks.
- Add support for displaying album art and track information.
- Include an equalizer to adjust audio settings.

19. Interactive Quiz Application (Advanced)

Objective: Enhance the previous quiz application with more complex questions, a scoring system, and user accounts.

Approach:

- Expand the question data to include various question types (multiple-choice, true/false, etc.).
- Use a database to store user accounts, quiz scores, and progress.
- Implement user authentication and login functionality.

Bonus Features:

- Add a timer to each question, with bonus points for answering quickly.
- Create leaderboards to showcase top scores among users.
- Allow users to create and share their quizzes.

20. Online Bookstore

Objective: Build an online bookstore where users can browse, search, and purchase books.

Approach:

- Design the HTML layout with sections for book categories and a search bar.
- Use CSS to style the bookstore and make it visually appealing.
- Write JavaScript to handle book search, filtering, and adding books to the cart.

Bonus Features:

- Implement a shopping cart feature to handle book purchases.
- Add user authentication and account creation for personalized shopping experiences.
- Include book recommendations based on user preferences and previous purchases.

21. Recipe Sharing Platform

Objective: Create a platform where users can share their favorite recipes, rate recipes, and leave comments.

Approach:

- Design the HTML layout with sections for recipe cards and user interactions.
- Use CSS to style the platform and make it visually appealing.
- Set up a backend with a database to store recipe data and user comments.

Bonus Features:

- Add a "Favorite" button for users to save recipes to their personal collection.
- Implement a recipe rating system with stars or thumbs up/down.
- Include social media sharing options for users to share recipes.

22. Online Learning Platform

Objective: Build an e-learning platform where users can access educational courses and track their progress.

Approach:

- Design the HTML layout with sections for course listings and user profiles.
- Use CSS to style the platform and make it visually appealing.
- Create a backend with user authentication and a database to store course data.

Bonus Features:

- Implement progress tracking and completion certificates for completed courses.
- Allow instructors to create and upload their courses to the platform.
- Add a review and rating system for courses.

23. Expense Tracker

Objective: Develop an app that helps users track their expenses and manage budgets.

Approach:

- Design the HTML layout with input fields for expenses and budget categories.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to handle expense calculations and display summaries.

Bonus Features:

- Include data visualization with charts to show spending patterns.
- Implement a budget planner feature to set monthly spending limits.
- Add local storage support for data persistence.

24. Social Media Dashboard

Objective: Create a dashboard that aggregates data from multiple social media platforms.

Approach:

- Design the HTML layout with sections for different social media metrics.
- Use CSS to style the dashboard and make it visually appealing.
- Fetch data from various social media APIs and display it on the dashboard.

Bonus Features:

- Allow users to connect their social media accounts and customize the dashboard.
- Add data filtering and sorting options for better analysis.
- Implement real-time updates for social media activity.

25. Language Learning App

Objective: Build an app that helps users learn new languages through interactive exercises.

Approach:

- Design the HTML layout with sections for language exercises and quizzes.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to handle language exercises and check user responses.

Bonus Features:

- Implement text-to-speech functionality to help with pronunciation.
- Include gamification elements like rewards and achievements for progress.
- Add multiple language support for learners studying different languages.

26. Fitness Tracker

Objective: Develop an app that tracks users' fitness activities and provides workout routines.

Approach:

- Design the HTML layout with sections for tracking fitness data and workout routines.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to handle user input and display workout recommendations.

Bonus Features:

- Implement a calendar view to track daily fitness activities.
- Add support for syncing with fitness wearables or smartphone health apps.
- Include video demonstrations for workout routines.

27. Job Board Platform

Objective: Create a platform for employers to post job listings and for job seekers to search and apply for jobs.

Approach:

- Design the HTML layout with sections for job listings and user profiles.
- Use CSS to style the platform and make it visually appealing.
- Create a backend with user authentication and a database to store job data.

Bonus Features:

- Implement a resume upload feature for job seekers.
- Add a notification system to alert job seekers of new job listings.
- Include a chat feature for employers and job seekers to communicate.

28. Online Auction Platform

Objective: Build an online auction platform where users can bid on items and win auctions.

Approach:

- Design the HTML layout with sections for auction listings and user bidding.
- Use CSS to style the platform and make it visually appealing.
- Create a backend with a database to handle auction data and user bids.

Bonus Features:

- Implement a countdown timer for auctions to create excitement.
- Allow users to set auto-bids to automatically bid up to a certain amount.
- Add a seller rating system based on successful auctions.

29. Personal Finance Manager

Objective: Develop an app that helps users manage their personal finances, budget, and track expenses.

Approach:

- Design the HTML layout with sections for expense tracking and budget planning.
- Use CSS to style the app and make it visually appealing.
- Write JavaScript to handle financial calculations and display summaries.

Bonus Features:

- Implement data visualization with charts to show spending patterns over time.
- Add expense categorization for better financial analysis.
- Include a savings goal feature to help users save for specific targets.

30. Polling and Survey Application

Objective: Create an app for conducting polls and surveys with user responses and results.

Approach:

- Design the HTML layout with sections for creating and submitting polls/surveys.

- Use CSS to style the app and make it visually appealing.

- Create a backend with a database to store poll/survey data and user responses.

Bonus Features:

- Implement real-time updates for poll/survey results.

- Add the ability for users to create their polls and share them.

- Include data analysis tools to visualize survey responses.

31. Address book

In this project, you have to build an application that can search for particular entries in your address book by filtering the attributes you specify.

32. Countdown timer

This is a countdown timer or clock. For this project, you just need to create a simple webpage that can update the time every second. With JavaScript as its foundation, you can make the page more appealing by including start, stop, and pause buttons on the page.

33. Word Counter

This is a pretty simple project which requires you to build an application that can parse texts and show the number of words and characters contained in a write-up. You can also include additional functionality in the word counter to provide more advanced information such as the number of passive sentences in a block of text.

34. Toast notifications

In this project, you will be required to design a toast notification tool. Using your JavaScript skills and knowledge, you've to create a functional toast notification tool that can respond to events on the page and notify the users as and when an event has been completed successfully. You could also use the setTimeout function to represent the delay in loading or saving data.

35. Social share buttons

In this project, you will take up the challenge of writing JavaScript code that will allow you to add social share buttons to static sites. While you can do this by incorporating HTML elements or images in the site's template, using JavaScript allows you to add the share buttons dynamically.

36. Note log

This project will be much like the to-do list project. The aim here is to design and build a notes app that can take multiple entries per note. It should allow users to select a note when they launch the app. When they choose a note, a new entry will be automatically tagged along with the current date, time, and location. Users can also sort and filter their entries based on this metadata. This is a great web app for tracking events and resolving messy calendar problems.

37. Exit the plugin

In this project, you will design an exit widget or plugin. When you visit a website or a webpage, you must have seen the tiny pop-ups that show on the screen when you wish to exit the site/page. Companies usually use exit plugins to show exciting offers to keep a user on the page. This is precisely what you have to design.

38. Survey form

Lots of companies use survey forms as a means of collecting relevant data about their target audience. In this project, you will have to design a full-fledged survey form that includes relevant questions like name, age, email, address, contact number, and other questions, depending on the type of company or organization you are shaping the form. This project will put to the test your webpage structuring skills

39. Tribute page
A tribute page is a webpage dedicated in honor of someone you love, admire, or respect. It can be a person or a beloved pet.
A tribute page is a perfect project for sharpening your HTML and CSS skills and knowledge. In this project, you will make a webpage where you can write and dedicate a tribute to someone and publish the same. Apart from the write-up for the tribute, you need to add relevant images, links, etc., on the page.

40. Google Search Engine lookalike

Another interesting JavaScript project on our list, this project requires you to build a webpage that resembles Google's home page. Keep in mind that you have to build a replica of Google home page along with the Google logo, search icons, text box, Gmail, and image buttons – basically, everything that you see on Google's home page. While building this project, you must ensure that the webpage can display at least ten search results (just like Google). Also, you must include the navigation arrow at the bottom of the webpage so that users can switch to the next page.

41. A drawing Tool
This JavaScript-based project uses JavaScript as a drawing tool to bring to life HTML and CSS elements on a web browser. The best thing about this project is that you can take advantage of JavaScript's supercool drawing libraries like oCanvas, Canviz, Raphael, etc.
By working on this project, you can learn how to use and implement JavaScript's drawing capabilities. This skill will come in handy for enhancing the appeal of static pages by adding graphical elements to them.

42. SEO-friendly website
Today, SEO is an integral part of website building. Without SEO, your website will not have the visibility to drive traffic from organic searches in SERPs (search engine result pages). While Web Developers are primarily concerned about website functionality, they must have a basic idea of web design and SEO. In this project, you will take up the role of a Digital Marketer and gain in-depth knowledge of SEO. When you are well-versed in SEO, you can build a website having user-friendly URLs and featuring an integrated, responsive design. This will allow the site to load quickly on both desktop and mobile devices, thereby strengthening a brand's social media presence.

43. To-do list
Create a web development project comprising various features essential for daily life. To-do lists contain user-interactive features with basic features like adding or removing tasks, highlighting dates, strikethrough features to indicate completion, and other text decoration components.
You can use JavaScript to build a web app that allows you to make to-do lists for routine tasks. For this project, you must be well-versed in HTML and CSS. JavaScript is the best choice for a to-do project since it allows users to design interactive coding lists where they can add, delete, and also group items.

44. JavaScript quiz game
This web development project aims to create a JavaScript quiz game that can take multiple answers and show the correct result to users. While gaining JavaScript knowledge isn't tricky, applying that knowledge in real-world scenarios is usually challenging. However, you can experiment with your skills by working on a small JavaScript-based quiz game.
While building this project, you will not only deal with complex logic, but you will also learn a lot about data management and DOM manipulation. Depending on your JavaScript skills and ability to handle complex logic, you can make the game as simple or complicated as you want it to be!

45. Product landing page
Being the face of any website, a product landing page has the ability to target customers more than any other aspect through its visuals and various other compelling features. Designing a product landing page is vital for web developers to test practical skills and how convincing they actually are. To develop a product landing page of a website, you must have sound knowledge of HTML and CSS. In this project, you will create columns and align the components of the landing page within the columns. You will have to perform basic editing tasks like cropping and resizing images, using design templates to make the layout more appealing, and so on.

46. Login authentication

Login authentication is a vital process, widely followed by organizations to keep their servers safe by allowing access only to authenticated users. Every website or application demands users to complete the login authentication process to cement their credentials for security and to improve user experience.
In this project, you will design a website's login authentication bar – where users enter their email ID/username and password to log in to the site. Since almost every website now comes with a login authentication feature, learning this skill will come in handy in your future web projects and applications.

47. Music player using JavaScript

The architecture, which is divided into three buckets, will be used: CSS (Cascading Style Sheets (adding styling to each element defined in the HTML file) JavaScript is a programming language that allows you to (adding elements for audio, player buttons, and music information) (when HTML elements are clicked, functionality is added).

Before diving into the code, take some time to plan this out. You will get your hands on HTMLMediaElement Interface to play audio files and control its playback. In this project you will get to work on Sliders, Flex Layout and different functions of JavaScript.

48. Build your own portfolio site

You should be able to organize a webpage with HTML, style its elements with CSS, and make the website interactive with JS.
Task is to create a form containing all the relevant fields to be filled by the person and after submission of the form, a portfolio will be created with appealing designs and styles.

When submit the contact form, you'll get an error message if:

- If any component e.g. Name, Email Address, or Message fields are blank, the message "This field is required" should appear.
- Also "use a valid email address" should appear if the email address is not formatted correctly.

49. Test Management System

A test management portal is to be designed using HTML, CSS and JS. For example, A portal consisting of MCQs based test can be designed and based upon the right/wrong answers a score sheet with detail can be maintained.

50. Big Boss Voting Zone

A Voting platform is to be designed to show the increase/decrease in the votes of particular contestant using HTML, CSS and JS.
The platform should look quite dynamic to show the progress bars of each contestant.

51. Caffe Management System

A Caffe management portal is to be designed using HTML, CSS and JS. For example many components can be added such as Menu list, Price list, Bill generator, Any reward policy if any, Best Customer etc.

52. Food Ordering Site

In this site, a person can choose food items based on various inciting and available menu options. A payment method and bill information can be added. You can also refer to Ufood app running in University for ordering of food.