

Task Given

Due date: (3hr)

1) Develop a online JS quiz web - application for user to test their JS skills

This include:

- **Login page** - Get name and email from user and start the quiz. (Validation required for username and email)
- **Quiz page** - Questions and options related to Javascript, submit button. Users can select only one correct answer.
- **Result page** - Display all questions, their correct answer, user's answer and view ranking button
- **Dashboard page** - Name, email, score, rank of each user. (Highest score first);

Note: If a user reattempts the quiz , update their score only if the new score is higher than the previous one.

```
const questions = [
  {
    question: "Which method is used to remove the last element from an array?",
    options: [
      { answer: "pop()", isCorrect: true },
      { answer: "shift()", isCorrect: false },
      { answer: "push()", isCorrect: false },
      { answer: "unshift()", isCorrect: false }
    ]
  },
  {
    question: "Which method is used to join all elements of an array into a string?",
    options: [
      { answer: "join()", isCorrect: true },
      { answer: "concat()", isCorrect: false },
      { answer: "slice()", isCorrect: false },
      { answer: "splice()", isCorrect: false }
    ]
  },
  {
    question: "Which method creates a new array with all elements that pass a test?",
    options: [
      { answer: "filter()", isCorrect: true },
      { answer: "map()", isCorrect: false },
      { answer: "reduce()", isCorrect: false },
      { answer: "forEach()", isCorrect: false }
    ]
  },
  {
    question: "Which of the following is not a valid JavaScript data type?",
```

```

options: [
  { answer: "Number", isCorrect: false },
  { answer: "String", isCorrect: false },
  { answer: "Float", isCorrect: true },
  { answer: "Boolean", isCorrect: false }
]
},
{
  question: "What will the following code output: `console.log(3 + '3')`?",
  options: [
    { answer: "33", isCorrect: true },
    { answer: "6", isCorrect: false },
    { answer: "NaN", isCorrect: false },
    { answer: "Error", isCorrect: false }
  ]
}
];

```

Due Date - (5hr)

2) BUILD A CAR RENTAL APPLICATION

1. Home Page

Design a home page that displays available cars for rent.

Features:

- Display a list of cars with details such as car model, price per day, and a "Book Now" button for each car.
- Dynamically populate the car list using JavaScript.
- Implement a "Book Now" button that stores the car information in localStorage and navigates to the booking page.

2. Booking Page

Create a booking page where users can select rental dates and confirm their booking.

Features:

- Create a form to select rental start and end dates.
- Validate that the end date is after the start date and the start date is not in the past.
- Calculate the total cost of the rental based on the car's price per day and the rental duration.
- Display a confirmation message with the rental details after submission.
- Store the booking details in local storage.

3. Admin Panel

Create an admin panel to add and manage cars in the rental system.

Features:

- Create a form to add new cars, including fields for car model and price per day.
- Display a list of all cars, with an option to remove cars from the list.
- Use local storage to store and persist car data across page reloads.

3) Develop a Web-Based Event Manager Application

Create a Web-Based Event Manager that allows users to manage personal events. The application should allow users to add, edit, delete, and filter events based on their status (upcoming or past). The application should have the following features:

Features:

1. Add Event:

Provide a form to input an event title and event date.

Upon submission, the event should be added to a list with the status automatically set as "Upcoming" if the date is in the future or "Past" if the date is in the past.

2. Edit Event:

Allow users to edit the title and date of an existing event.

Recalculate and update the event's status after editing based on the new date.

3. Delete Event:

Provide an option to delete an event from the list.

4. Display Events:

List all events with their title, date, and status (Upcoming or Past).

Dynamically render events on the page using JavaScript.

5. Filter Events:

Implement two filter options:

1. Upcoming Events: Only events that are yet to occur.
2. Past Events: Only events that have already occurred.

6. Event Status:

Automatically classify events as "Upcoming" or "Past" based on the event's date compared to the current date.

Use JavaScript's Date() object to check whether the event is in the future or past.