

Name: Raj Koyani

Reg no: 21MIS1017

DA:2

Code:

Create-workout:

```
const exerciseForm = document.querySelector("#exercise-form");
const exerciseSearch = document.querySelector("#exercise-search");
const exerciseSuggestions = document.querySelector("#exercise-suggestions");
const selectedExercisesContainer = document.querySelector(
  "#selected-exercises"
);
const exercisesCheckboxes = document.querySelectorAll(".exercise-checkbox");

const submitButton = document.querySelector("#submit-button");

submitButton.addEventListener("click", (e) => {
  e.preventDefault();
  const workoutName = document.querySelector("#workout-name").value;
  const selectedExercises =
    Array.from(selectedExercisesContainer.children).map(
      (exerciseElement) => {
        const exerciseName = exerciseElement.textContent;
        const setsInput = exerciseElement.querySelector(".sets-input");
        const weightInput = exerciseElement.querySelector(".weight-input");
        const repsInput = exerciseElement.querySelector(".reps-input");
        const sets = setsInput.value || 0;
        const weight = weightInput.value || 0;
        const reps = repsInput.value || 0;
        return { name: exerciseName, sets, reps, weight };
      }
    );

  if (workoutName && selectedExercises.length > 0) {
    const workout = {
      name: workoutName,
      exercises: selectedExercises,
    };
  }
});
```

```

    let workouts = JSON.parse(localStorage.getItem("workouts")) || [];
    workouts.push(workout);
    localStorage.setItem("workouts", JSON.stringify(workouts));

    window.location.href = "workouts.html";
  } else {
    alert("Please enter a workout name and select at least one exercise.");
  }
});

exerciseSearch.addEventListener("input", (e) => {
  const searchQuery = e.target.value.trim();
  const suggestions = getExerciseSuggestions(searchQuery);
  displayExerciseSuggestions(suggestions);
});

exerciseSuggestions.addEventListener("click", (e) => {
  if (e.target.tagName === "LI") {
    const exerciseName = e.target.textContent;
    addSelectedExercise(exerciseName);
  }
});

selectedExercisesContainer.addEventListener("click", (e) => {
  if (e.target.classList.contains("delete-button")) {
    const exerciseElement = e.target.parentNode;
    exerciseElement.parentNode.removeChild(exerciseElement);
  }
});

function getExerciseSuggestions(searchQuery) {
  const hardcodedSuggestions = [
    "Bench Press",
    "Squat",
    "Deadlift",
    "Pull-ups",
    "Push-ups",
    "Barbell Rows",
    "Shoulder Press",
    "Lunges",
    "Dumbbell Curls",
    "Tricep Dips",
    "Plank",
    "Russian Twists",
    "Leg Press",
    "Calf Raises",
    "Lat Pulldowns",
  ];

```

```

    "Dumbbell Flyes",
    "Hamstring Curls",
    "Incline Bench Press",
    "Front Squats",
    "Chest Press",
    "Overhead Tricep Extension",
    "Step-ups",
    "Hamstring Curls",
    "Front Squats",
    "Chest Dips",
    "Military Press",
    "Lateral Raises",
    "Hip Thrusts",
    "Preacher Curls",
    "Skull Crushers",
    "Burpees",
    "Barbell Lunges",
    "Leg Press Machine",
    "Machine Shoulder Press",
    "Good Mornings",
    "Close Grip Cable Pulldowns",
    "Dumbbell Flyes",
    "Overhead Cable Tricep Extension",
    "Box Step-ups",
    "Bent-Over Reverse Flyes",
    "Seated Leg Press",
    "EZ Bar Curls",
    "Tricep Cable Pushdowns",
    "Bent-Over Dumbbell Rows",
    "Seated Overhead Dumbbell Press",

];

return hardcodedSuggestions.filter((suggestion) =>
    suggestion.toLowerCase().includes(searchQuery.toLowerCase())
);
}

function displayExerciseSuggestions(suggestions) {
    if (exerciseSearch.value.trim() === "") {
        exerciseSuggestions.innerHTML = "";
        return;
    }

    exerciseSuggestions.innerHTML = "";
    suggestions.forEach((suggestion) => {
        const suggestionElement = document.createElement("li");
        suggestionElement.textContent = suggestion;
    });
}

```

```

        exerciseSuggestions.appendChild(suggestionElement);
    });
}

function addSelectedExercise(exerciseName) {
    const exerciseElement = document.createElement("li");
    const exerciseNameElement = document.createElement("span");
    exerciseNameElement.textContent = exerciseName;
    exerciseElement.appendChild(exerciseNameElement);

    const repsInput = document.createElement("input");
    repsInput.type = "number";
    repsInput.classList.add("reps-input");
    repsInput.placeholder = "Reps";
    exerciseElement.appendChild(repsInput);

    const setsInput = document.createElement("input");
    setsInput.type = "number";
    setsInput.classList.add("sets-input");
    setsInput.placeholder = "Sets";
    exerciseElement.appendChild(setsInput);

    const weightInput = document.createElement("input");
    weightInput.type = "number";
    weightInput.classList.add("weight-input");
    weightInput.placeholder = "Weight";
    exerciseElement.appendChild(weightInput);

    const deleteButton = document.createElement("button");
    deleteButton.classList.add("delete-button");
    exerciseElement.appendChild(deleteButton);

    selectedExercisesContainer.addEventListener("click", (e) => {
        if (e.target.classList.contains("delete-button")) {
            const exerciseElement = e.target.parentNode;
            exerciseElement.parentNode.removeChild(exerciseElement);
        }
    });

    selectedExercisesContainer.appendChild(exerciseElement);
}

```

## Start-workout.js

```
const urlParams = new URLSearchParams(window.location.search);
```

```

const workoutIndex = urlParams.get("workout");
const workouts = JSON.parse(localStorage.getItem("workouts")) || [];
let currentWorkout = workouts[workoutIndex] || {};

const workoutNameElement = document.querySelector("#workout-name");
const exercisesListElement = document.querySelector("#exercises-list");
const timerElement = document.querySelector("#timer");
const finishButton = document.querySelector("#finish-button");

let timerInterval;
let startTime;
let elapsedTime = 0;

const renderTimer = () => {
  const seconds = Math.floor(elapsedTime / 1000) % 60;
  const minutes = Math.floor(elapsedTime / 1000 / 60) % 60;
  const hours = Math.floor(elapsedTime / 1000 / 60 / 60);

  const formattedTime = `${hours.toString().padStart(2, "0")}:${minutes
    .toString()
    .padStart(2, "0")}:${seconds.toString().padStart(2, "0")}`;
  timerElement.textContent = formattedTime;
};

const startTimer = () => {
  startTime = Date.now() - elapsedTime;
  timerInterval = setInterval(() => {
    elapsedTime = Date.now() - startTime;
    renderTimer();
  }, 1000);
};

const finishWorkout = () => {
  clearInterval(timerInterval);

  const updatedExercises = Array.from(
    document.querySelectorAll(".exercise-item")
  ).map((exerciseItem) => {
    const exerciseName =
      exerciseItem.querySelector(".exercise-name").textContent;
    const setsInput = exerciseItem.querySelector(".sets-input");
    const repsInput = exerciseItem.querySelector(".reps-input");
    const weightInput = exerciseItem.querySelector(".weight-input");

    const sets = parseInt(setsInput.value, 10) || 0;
    const reps = parseInt(repsInput.value, 10) || 0;
    const weight = parseInt(weightInput.value, 10) || 0;
  });
};

```

```

    return {
      name: exerciseName,
      sets,
      reps,
      weight,
    };
  });

  currentWorkout = {
    ...currentWorkout,
    exercises: updatedExercises,
    completedAt: new Date().toISOString(),
  };

  workouts[workoutIndex] = currentWorkout;
  localStorage.setItem("workouts", JSON.stringify(workouts));
  const endTime = Date.now();
  const trainingDuration = Math.floor((endTime - startTime) / 1000);
  const previousWorkoutIndex = workoutIndex - 1;
  const previousWorkout = workouts[previousWorkoutIndex];

  const trainingInfo = {
    duration: trainingDuration,
    exercises: updatedExercises,
    previousWorkout,
  };

  const trainingInfoWindow = window.open("", "_blank",
    "width=600,height=400");
  trainingInfoWindow.document.write(`
    <html>
    <head>
    <title>Training Information</title>
    </head>
    <body>
    <h1>Training Information</h1>
    <p>Duration: ${formatDuration(trainingDuration)}</p>
    <h2>Exercises:</h2>
    <ul>
      ${updatedExercises
        .map(
          (exercise) =>
            `<li>${exercise.name} - Sets: ${exercise.sets}, Reps:
${exercise.reps}, Weight: ${exercise.weight} lbs</li>`
        )
        .join("")}
    </ul>
    ${

```

```

    previousWorkout
      ? `

## ${exercise.name} - You lifted ${weightDifference} lbs more than the previous workout</li>`; } return ""; }) .join("")} </ul>` : "" } </body> </html> `); window.location.href = "index.html"; }; const formatDuration = (duration) => { const seconds = duration % 60; const minutes = Math.floor((duration / 60) % 60); const hours = Math.floor(duration / 3600); const formattedTime = `${hours.toString().padStart(2, "0")}:${minutes .toString() .padStart(2, "0")}:${seconds.toString().padStart(2, "0")}`; return formattedTime; }; workoutNameElement.textContent = currentWorkout.name; currentWorkout.exercises.forEach((exercise) => { const exerciseItem = document.createElement("div"); exerciseItem.classList.add("exercise-item"); exerciseItem.innerHTML = ` <div class="exercise-name">${exercise.name}</div> <div> <label for="sets-${exercise.name}">Sets:</label> <input id="sets-${exercise.name}" class="sets-input" type="number" min="0" value="${exercise.sets}" />


```

```

        </div>
        <div>
          <label for="reps-${exercise.name}">Reps:</label>
          <input id="reps-${exercise.name}" class="reps-input" type="number"
min="0" value="${exercise.reps}" />
        </div>
        <div>
          <label for="weight-${exercise.name}">Weight:</label>
          <input id="weight-${exercise.name}" class="weight-input" type="number"
min="0" value="${exercise.weight}" />
        </div>
      </div>`;
      exercisesListElement.appendChild(exerciseItem);
    });

finishButton.addEventListener("click", () => {
  finishWorkout();
});

startTimer();

```

## Workouts.js

```

const workoutsList = document.querySelector("#workouts-list");

let workouts = JSON.parse(localStorage.getItem("workouts")) || [];
workouts.forEach((workout, index) => {
  const workoutItem = document.createElement("li");
  workoutItem.innerHTML = `
    <div class="workout-container">
      <span class="workout-name">${workout.name}</span>
      <div class="exercises">
        ${workout.exercises
          .map(
            (exercise, exerciseIndex) => `
              <div class="exercise">
                <span class="exercise-name">${exercise.name}</span>
                <div class="exercise-details">
                  <span>${exercise.sets} sets, ${exercise.reps}
reps, ${exercise.weight} weight</span>
                  <button class="delete-exercise" data-workout="${index}" data-
exercise="${exerciseIndex}">

```



```

        <div class="sign">X</div>

        <div class="d-text">Delete</div>
    </button>        </div>
</div>
,
)
.join("")]

workoutsList.appendChild(workoutItem);
});

const deleteButtons = document.querySelectorAll(".delete-workout");
deleteButtons.forEach((button) => {
    button.addEventListener("click", (e) => {
        const index = e.target.dataset.index;
        workouts.splice(index, 1);
        localStorage.setItem("workouts", JSON.stringify(workouts));
        location.reload();
    });
});

const deleteExerciseButtons = document.querySelectorAll(".delete-exercise");
deleteExerciseButtons.forEach((button) => {
    button.addEventListener("click", (e) => {
        const workoutIndex = e.target.dataset.workout;
        const exerciseIndex = e.target.dataset.exercise;

        if (
            workouts[workoutIndex] &&
            workouts[workoutIndex].exercises[exerciseIndex]
        ) {
            workouts[workoutIndex].exercises.splice(exerciseIndex, 1);

            if (workouts[workoutIndex].exercises.length === 0) {
                workouts.splice(workoutIndex, 1);
            }

            localStorage.setItem("workouts", JSON.stringify(workouts));
            location.reload();
        }
    });
});
});

```

## Database.php

```

<?php

$name = $_POST["name"] ?? "";
$address = $_POST["address"] ?? "";
$dob = $_POST["dob"] ?? "";
$mobile = $_POST["mobile"] ?? "";
$email = $_POST["email"] ?? "";
$username = $_POST["username"] ?? "";
$password = $_POST["password"] ?? "";

if (empty($name) || empty($address) || empty($dob) || empty($mobile) ||
empty($email) || empty($username) || empty($password)) {
    die("Error: All input fields are required");
}

if (!preg_match('/^[A-Za-z\s]+$/ ', $name)) {
    die("Error: Name must consist of only alphabets and white spaces");
}

if (!preg_match('/^[0-9]{10}$/', $mobile)) {
    die("Error: Mobile number must consist of 10 digits only");
}

if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    die("Error: Invalid email address");
}

if (strlen($password) < 8 || strlen($password) > 20) {
    die("Error: Password length should be between 8 and 20");
}

error_reporting(E_ALL);
ini_set('display_errors', 1);

$hostname = 'localhost';
$username = 'root';
$password = '';
$database = 'signup';

$connection = mysqli_connect($hostname, $username, $password, $database);
if (!$connection) {
    die("Database connection failed: " . mysqli_connect_error());
}

// Step 2: Retrieve form data
$name = $_POST['name'];
$address = $_POST['address'];
$dob = $_POST['dob'];

```

```

$mobile = $_POST['mobile'];
$email = $_POST['email'];
$username = $_POST['username'];
$password = $_POST['password'];

$name = mysqli_real_escape_string($connection, $name);
$address = mysqli_real_escape_string($connection, $address);

$query = "INSERT INTO details (name, address, dob, mobile, email, username,
password)
VALUES ('$name', '$address', '$dob', '$mobile', '$email',
'$username', '$password')";

$result = mysqli_query($connection, $query);
if ($result) {
    echo "Data successfully stored in the database.";
} else {
    echo "Error: " . mysqli_error($connection);
}

mysqli_close($connection);
?>

```

## Singup.html

```

<!DOCTYPE html>
<html>
  <head>
    <title>Survey Form</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        margin: 0;
        padding: 0;
        background: linear-gradient(135deg,rgb(77, 77, 77), rgb(10, 7, 7));
      }

      .container {
        max-width: 500px;
        margin: 50px auto;
        padding: 20px;
        background-color: rgba(255, 255, 255, 0.907);
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
        border-radius: 8px;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h1>Singup</h1>
    </div>
  </body>
</html>

```

```
}

h1 {
  text-align: center;
  color: #000;
}

form {
  margin-top: 30px;
}

label {
  display: block;
  margin-bottom: 5px;
  color: #000;
}

input[type="text"],
input[type="date"],
input[type="email"],
input[type="password"] {
  width: 95%;
  padding: 10px;
  border: none;
  border-radius: 4px;
  margin-bottom: 10px;
  background-color: grey;
  color: #000;
}

input[type="text"]::placeholder,
input[type="date"]::placeholder,
input[type="email"]::placeholder,
input[type="password"]::placeholder {
  color: black;
}

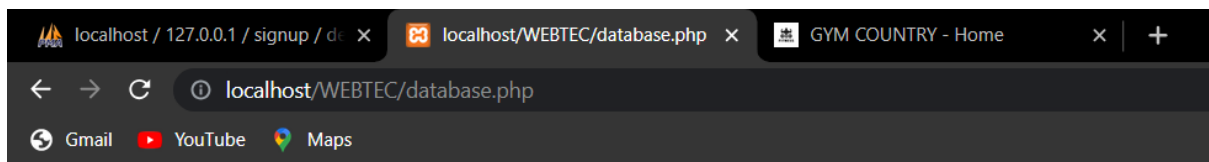
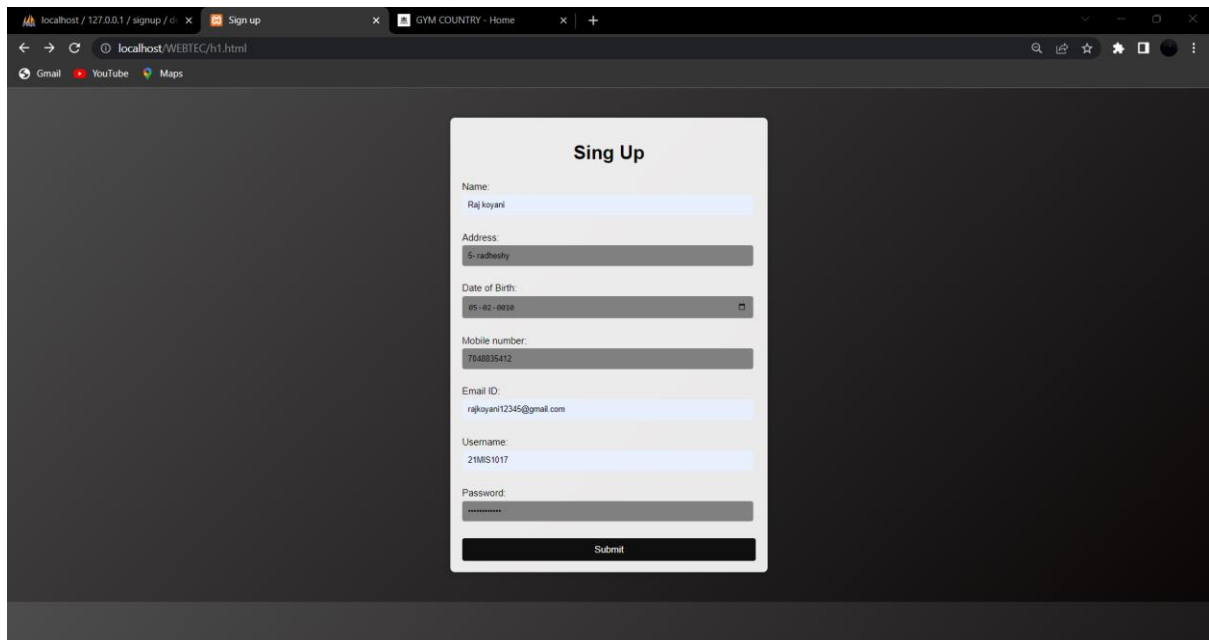
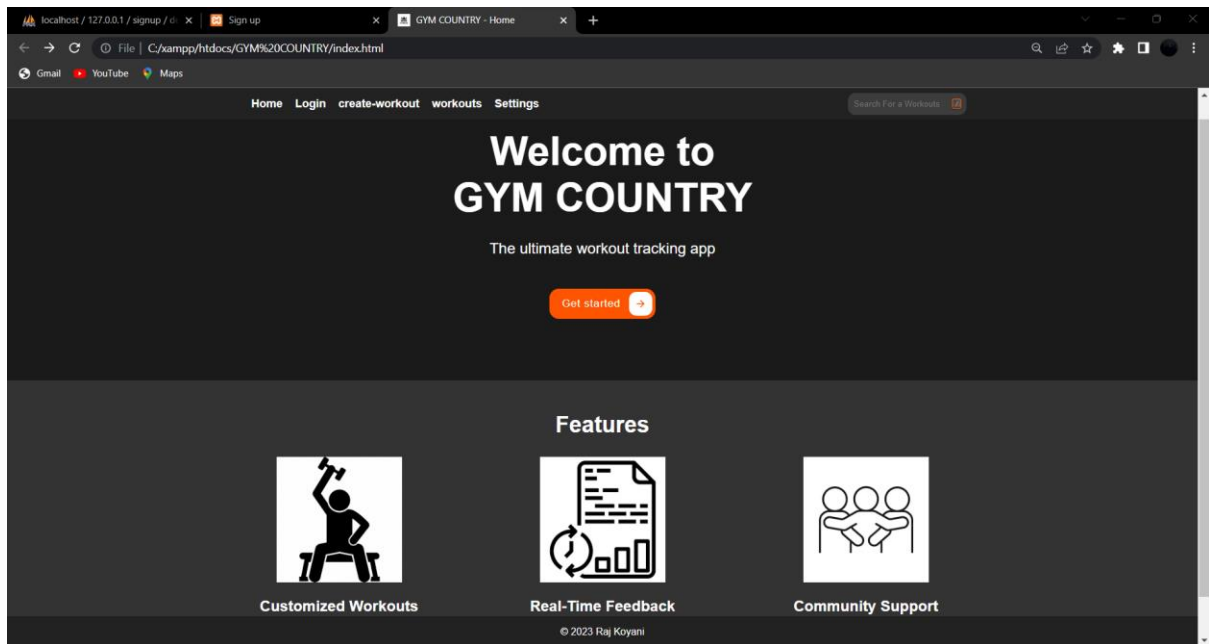
input[type="submit"] {
  width: 100%;
  padding: 10px;
  background-color: #0f0f0f;
  border: none;
  color: #fff;
  cursor: pointer;
  border-radius: 4px;
  font-size: 16px;
}
```

```

        input[type="submit"]:hover {
            background-color: #3a373a;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Sing Up</h1>
        <form name="clientForm" action="database.php" method="post">
            <label for="name">Name:</label>
            <input type="text" name="name" required placeholder="Enter your name"
/><br /><br />
            <label for="address">Address:</label>
            <input type="text" name="address" required placeholder="Enter your
address" /><br /><br />
            <label for="dob">Date of Birth:</label>
            <input type="date" name="dob" required placeholder="Select your date
of birth" /><br /><br />
            <label for="mobile">Mobile number:</label>
            <input type="text" name="mobile" required placeholder="Enter your
mobile number" /><br /><br />
            <label for="email">Email ID:</label>
            <input type="email" name="email" required placeholder="Enter your
email address" /><br /><br />
            <label for="username">Username:</label>
            <input type="text" name="username" required placeholder="Choose a
username" /><br /><br />
            <label for="password">Password:</label>
            <input type="password" name="password" required placeholder="Enter
your password" /><br /><br />
            <input type="submit" value="Submit">
        </form>
    </div>
</body>
</html>

```

Output:



Data successfully stored in the database.

