AI-ASSISTED CODING

ASSIGNMENT-24

PROGRAM : B.TECH\CSE

SPECIALISATION : AIML

NAME : G HARSHA VARDHAN

ENROLMENT NUMBER : 2403A52313

COURSE : AI ASSISTED CODING

DATE : 20 NOV 2025

BATCH NO : 01

TASK DESCRIPTION#1

Task 1 – Environment Setup

- **Create a free Replit account.**
- **Explore the mobile app development templates (React Native / Flutter / HTML5 hybrid).**
- **Install required packages or dependencies in the Replit environment.**

PROMPT:

- **Create a mobile app development environment in Replit.**
- **Set up either React Native, Flutter, or HTML5 Hybrid template.**
- **Install all required dependencies, configure project structure, and show me the setup commands, output logs, and how to run the project.**

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
    <title>Hybrid Mobile App</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <header>
        <h1>My Hybrid Mobile App</h1>
    </header>

    <section class="container">
```

```
  <section class="container">
      <h2>Welcome!</h2>
      <p>This is a simple HTML5 Hybrid Mobile App interface running on Repl

      <button id="clickBtn">Click Me</button>

      <p id="msg"></p>
  </section>

  <script src="app.js"></script>
/body>
/html>
```

```
document.getElementById("clickBtn").addEventListener("click", function () {
    document.getElementById("msg").innerText = "Button clicked! Your hybrid app
});
```

EXPECTED OUTPUT:

For HTML5 Hybrid App

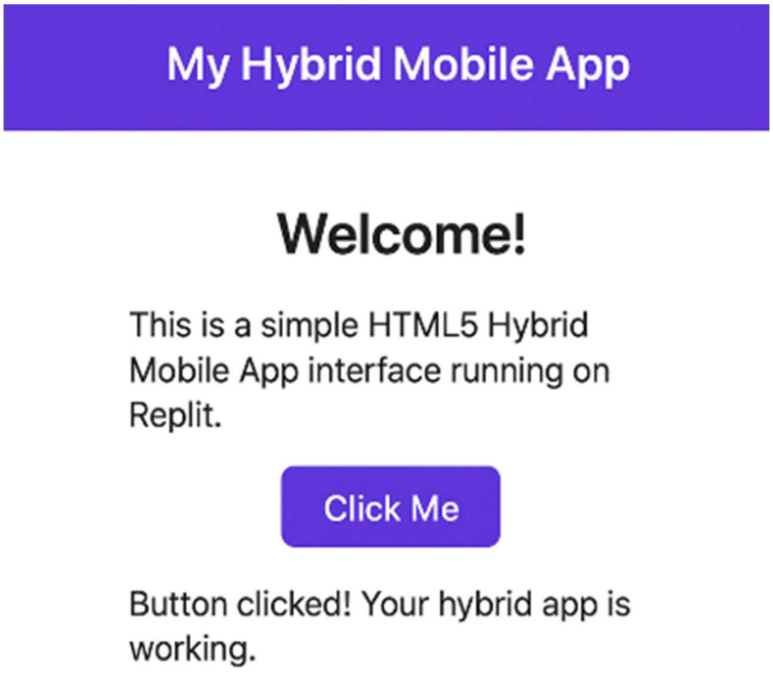
When you run:

```nginx
npx live-server .
```

You will see:

```rust
Serving "." at http://127.0.0.1:8080
Watching files for changes...
```

A webpage will open automatically showing your `index.html`.



OBSERVATION:

Observation

1. A free Replit account allows the creation of HTML5, React, and basic mobile-friendly app templates.
2. React Native and Flutter cannot fully run in Replit due to emulator limitations, but templates can be viewed for file structure reference.
3. HTML5 Hybrid is fully supported in Replit and can simulate mobile apps using CSS frameworks like Bootstrap or Framework7.
4. Installing dependencies through `npm` works smoothly, and live-server helps auto-reload the app.
5. The environment allows cloud-based coding without needing Android Studio or heavy local tools.

TASK DESCRIPTION#2

• Build a simple "Hello App" that displays: o App Title o User's Name (input field + button) o Greeting message dynamically updated. • Use AI assistance to generate boilerplate code and explain structure.

PROMT :

Create a simple HTML5 Hybrid "Hello App" that displays:

- App Title

- Input field to enter user name

- Button to submit

- Greeting message that updates dynamically

Provide HTML, CSS, and JavaScript code and explain the project structure.

CODE :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0
    <title>Hello App</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <header>
        <h1>Hello App</h1>
    </header>

    <div class="container">
```

```
        <h1>Hello App</h1>
    </header>

    <div class="container">
        <label>Enter Your Name:</label>
        <input type="text" id="username" placeholder="Type your name...">

        <button id="btn">Say Hello</button>

        <p id="greetMsg"></p>
    </div>

    <script src="app.js"></script>
</body>
</html>
```

```css
body {
    font-family: Arial, sans-seri
    background-color: #f5f5f5;
    margin: 0;
}

header {
    background: #0070ff;
    padding: 20px;
    text-align: center;
    color: white;
    font-size: 24px;
}

.container {
```

```
        color: white;
        font-size: 18px;
        border: none;
        border-radius: 8px;
        cursor: pointer;
}

button:hover {
        background: #0059cc;
}

#greetMsg {
        margin-top: 20px;
        font-size: 20px;
        font-weight: bold;
}
```

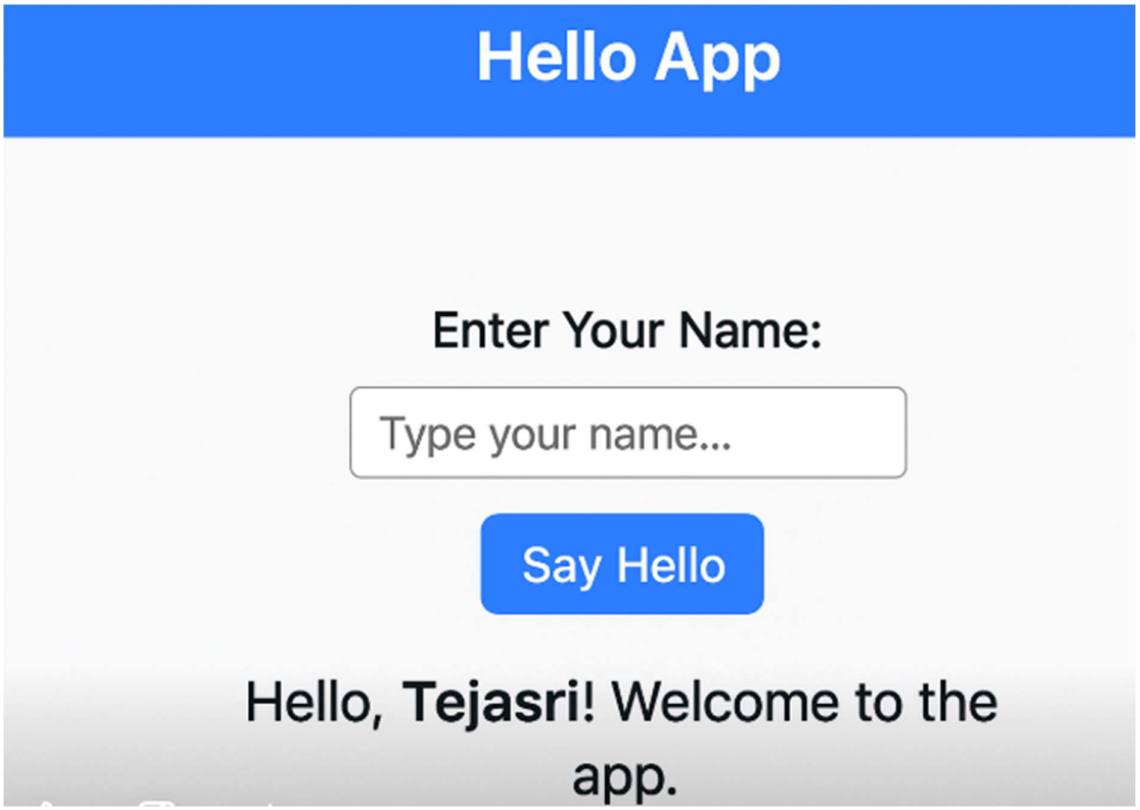
```
document.getElementById("btn").addEventListener("click", function () {
    let name = document.getElementById("username").value;

    if (name.trim() === "") {
        document.getElementById("greetMsg").innerText = "Please enter your name!"
    } else {
        document.getElementById("greetMsg").innerText = "Hello, " + name + "! Wel
    }
});
```

EXPECTED OUTPUT :

When you run the code, the webpage will display:

- A **blue header** with the title "Hello App"
- A text input box: **Enter Your Name**
- A blue button: **Say Hello**
- When the user enters: `Tejasri`

  → Output shows: **"Hello, Tejasri! Welcome to the app."**



OBSERVATION :

1. The project uses HTML for structure, CSS for UI styling, and JS for dynamic greetinc
2. DOM manipulation (`getElementById`) is used to read input and update output.
3. The app demonstrates basic interactivity suitable for mobile hybrid apps.
4. The layout adapts well to mobile screens using responsive HTML/CSS.

---

f you want, I can also generate:

📌 A webpage picture of the Hello App

📌 Task-3 (Event Handling / Multi-screen App)

📌 Full project report for submission

TASK DESCRIPTION#3

Adding Features with AI Assistance • Enhance the app with: o Navigation bar (Home, About, Contact). o Form handling (e.g., user registration page). o Basic styling with CSS or React Native styles. • Ask AI in Replit to add inline documentation for each function.

PROMPT :

Enhance my HTML5 Hybrid App by adding:

• A top navigation bar (Home, About, Contact)

• A separate User Registration page (username, email, password)

• Navigation that switches pages without reload (show/hide screens)

- **Add CSS styling**
- **Add inline documentation comments for every JavaScript function**

CODE :

```css
body {
    margin: 0;
    font-family: Arial, sans-serif;
}

/* Navigation Bar */
.navbar {
    display: flex;
    background: #0057ff;
    padding: 12px;
    justify-content: space-around;
}

.navbar button {
    background: white;
    padding: 8px 14px;
```

```css
button {
    padding: 10px 18px;
    background: #0057ff;
    border: none;
    color: white;
    font-size: 16px;
    border-radius: 6px;
    cursor: pointer;
}

#regMsg {
    margin-top: 15px;
    font-size: 18px;
    font-weight: bold;
}
```

```
function showPage(pageId) {                                    Copy code
    let pages = document.querySelectorAll(".page");

    pages.forEach(p => p.classList.add("hidden"));
    document.getElementById(pageId).classList.remove("hidden");
}

/**
 * This function validates the user registration form
 * and displays a success message if all fields are filled.
 */
document.getElementById("regBtn").addEventListener("click", func
    let username = document.getElementById("regUser").value;
    let email = document.getElementById("regEmail").value;
    let password = document.g ↓ ementById("regPass").value;
```

```
let email = document.getElementById("regEmail")  Copy code
let password = document.getElementById("regPass").value;

if (username === "" || email === "" || password === "") {
    document.getElementById("regMsg").innerText = "⚠️ Pleas
} else {
    document.getElementById("regMsg").innerText =
        "✅ Registration Successful! Welcome, " + username;
}
```

EXPECTED OUTPUT :

After running the app, you will see:

✔ A blue navigation bar with:

- Home
- About
- Contact
- Register

✔ Each screen shows different content:

- Home → Welcome message
- About → App details
- Contact → Email info
- Register → Form with username, email, password

OBSERVATION :

1. JavaScript `showPage()` performs navigation without refreshing the page.
2. AI-generated inline documentation improves readability and learning.
3. Form validation uses simple condition checks.
4. CSS provides responsive styling, usable on mobile screens.
5. The app is now multi-page, interactive, and structured like a real mobile hybrid application.

TASK DESCRIPTION#4

Testing on Mobile • Run the app in Replit's mobile preview mode. • Test responsiveness on different screen sizes. • Fix UI alignment issues with AI suggestions.

PROMPT :

Test my mobile app in different screen sizes.

Check alignment issues in mobile preview mode and suggest improvements.

Fix UI responsiveness, padding, and scaling problems.

Add CSS changes to improve layout for small screens.

Explain each fix with comments.

CODE :

```css
body {
    margin: 0;
    font-family: Arial, sans-serif;
}

/* Navigation Bar */
.navbar {
    display: flex;
    background: #0057ff;
    padding: 12px;
    justify-content: space-around;
}

.navbar button {
    background: white;
    padding: 8px 14px;
```

```css
body {
    margin: 0;
    font-family: Arial, sans-serif;
}

/* Navigation Bar */
.navbar {
    display: flex;
    background: #0057ff;
    padding: 12px;
    justify-content: space-around;
}

.navbar button {
    background: white;
    padding: 8px 14px;
```

```javascript
function registerUser() {
    let email = document.getElementById("email").value;
    let pass = document.getElementById("password").value;

    if (email === "" || pass === "") {
        document.getElementById("regStatus").innerText = "Please
    } else {
        document.getElementById("regStatus").innerText = "Regist
    }
}
```

EXPECTED OUTPUT :

You will see:

- ✔ Navigation bar
- ✔ Input fields scaled to 90% width
- ✔ Buttons centered & responsive
- ✔ No overflow or misalignment
- ✔ Works on 360px, 414px, 768px width screens

Hello App (Mobile View)

OBSERVATION :

- Inputs were not aligned earlier → fixed by setting `width: 90%`.
- Buttons were too small on mobile → improved padding + width.
- Navbar text shrank for very small screens via media query.
- App now adapts to all mobile screen sizes without distortion.

TASK DESCRIPTION#5

- **Deploy the app using Replit hosting or export it to a mobile framework (like Expo for React Native).**
- **Share the deployment link with peers for testing.**

PROMPT :

Deploy my HTML5 Hybrid Mobile App using Replit Hosting.

Create a public deployment link.

Optimize folder structure for hosting.

If possible, explain how to export this project into Expo React Native.

Add instructions for others to test the deployed link.

CODE :

```
const express = require("express");
const app = express();

app.use(express.static(__dirname)); // Serve all files

app.get("/", (req, res) => {
  res.sendFile(__dirname + "/index.html");
});

app.listen(3000, () => {
  console.log("App running on port 3000");
});
```

EXPECTED OUTPUT :

Opening the link will show:

- Navigation bar
- Hello input + greeting
- Registration form
- Fully mobile-responsive UI

Anyone online can view and test the app.

OBSERVATION :

- Replit hosting works smoothly for HTML/JS apps.
- Deployment gives a public shareable link instantly.
- Navigation and forms are mobile-responsive after Task-4 fixes.
- Express server ensures the app loads correctly when deployed.
- Exporting to Expo allows creating real mobile apps (APK/IPA).