Internship Report – Week 5, Day 6
Topic: Asynchronous JavaScript – setTimeout, setInterval, and Promises
Name: Hamza Rafique

Work Done Today:

Today I learned about asynchronous programming in JavaScript, focusing on three important concepts:

1. **setTimeout()**
    - Executes a function once after a specific time delay.
    - Useful for delaying actions like showing messages or animations.
    - Example:

javascript

CopyEdit

```
setTimeout(() => console.log("Hello after 2s"), 2000);
```

2. **setInterval()**
    - Runs a function repeatedly at fixed intervals until stopped.
    - Useful for periodic tasks like timers or updating UI.
    - Example:

javascript

CopyEdit

```
const id = setInterval(() => console.log("Tick"), 1000);
```

3. **Promises**
    - A cleaner way to handle asynchronous operations without nested callbacks.
    - Has three states: Pending, Fulfilled, Rejected.
    - Example:

javascript

CopyEdit

```
new Promise((resolve) => resolve("Done"))
```

```
  .then(msg => console.log(msg));
```

Key Learnings:

- Asynchronous JS prevents blocking the main thread.
- setTimeout is for one-time delayed execution, setInterval is for repeated execution.
- Promises improve readability and maintainability in async code.

Challenges Faced:

- Understanding the difference between synchronous and asynchronous execution order.
- Learning how to stop setInterval and cancel setTimeout.

Next Steps:

- Practice with async/await for handling promises.
- Build a small project using timers and promises together.