# Test Plan – Full Stack Web Application Automation

## 1. Introduction

This test plan outlines the testing strategy for a full-stack web application involving both UI and API automation. The application includes a React frontend and a Node.js/Express backend. The objective is to validate key functionalities like user registration, login, and CRUD operations on goals through automated test scripts.

## 2. Scope of Testing

The scope includes functional validation of the following:

* - UI flows for registration, login, goal creation, goal deletion
* - API endpoints for user and goal management (CRUD operations)

## 3. Test Coverage Areas

UI Tests cover:

* - User Registration
* - Login with valid/invalid credentials
* - Create new goal
* - Delete goal

API Tests cover:

* - Register user via POST /api/users
* - Login via POST /api/users/login
* - Create goal via POST /api/goals
* - Get goals via GET /api/goals
* - Update goal via PUT /api/goals/:id
* - Delete goal via DELETE /api/goals/:id

## 4. Tools and Technologies

- Playwright with Java for UI automation

- REST-assured for API testing

- JUnit 5 as the test framework

- ExtentReports for detailed HTML reporting

- CI/CD integration.

## 5. How to Execute Tests

- Prerequisites: Java 17+, Maven, Chrome, Node server running

- To run all UI tests: `mvn test -Dgroups=ui`

- To run API tests: `mvn test -Dgroups=api`

- Reports are generated in the `/reports` folder

## 6. Assumptions and Limitations

- Tests rely on consistent server responses

- Tests are executed sequentially based on dependencies

# 7.Documentation & Setup

Complete README.md file is included with:  
- App setup  
- API/Frontend launch  
- Test execution commands