# Semester Project – Full Stack AI-Integrated Web Application (Section 5)

## Project Title:

AI-Powered Peer Tutoring & Study Planner (teams may rename)

## Overview

Build, integrate, and deploy a modern web app that connects students for peer tutoring while helping them plan study time and track progress. The app should include a responsive React front end, a lightweight API layer (mocked or real), and at least one meaningful AI capability (chatbot, content generation, analytics, or accessibility checks).

## Core Features

- Responsive UI for desktop, tablet, and mobile  
- Account & Profiles (mocked or real)  
- Tutor Match & Booking  
- Study Planner with progress dashboard  
- Real or Mock API  
- AI Integration: TutorBot, AI Study Plan Generator, Accessibility Audit Helper, Auto-summaries  
- Compliance & Quality: Accessibility, Cross-browser testing  
- Documentation: README, API docs, user guide, and AI-reflection report

## Tech Stack (recommended)

Front end: HTML5, CSS3 (or Tailwind), React + React Router  
State: React Query/Context  
API: Mock Service Worker / JSON-Server / Node/Express  
Storage: JSON or a free DB  
AI: Open-source LLM endpoint or local rules

### Phase 1 – Planning & Proposal

* Team Lead: Roadmap, scope control, risk log
* AI Researcher: Picks AI use cases; drafts prompt library
* Documentation Lead: Proposal + AI-reflection outline
* UX Designer: Wireframes, user journeys
* Scrum Reporter: Standups/notes, weekly update

### Phase 2 – Front-End Development

* HTML/CSS Developer: Layouts, components, design system
* Accessibility Tester: WCAG checks, keyboard paths
* Component Generator: Uses AI to scaffold boilerplate
* Responsive Designer: Breakpoints and mobile polish
* Quality Analyst: Cross-browser suite + bug triage

### Phase 3 – Backend Simulation & API

* Mock Server Lead: JSON-Server/MSW/Express endpoints
* AJAX Programmer: Data fetching, optimistic updates
* Logger: Error/reporting hooks; network log
* Bug Tracker: Issues → priorities → fixes
* Reviewer: API docs (endpoints, payloads, examples)

### Phase 4 – Final Integration & Deployment

* Deployment Specialist: Hosting (Vercel/Netlify/Render)
* Repo Manager: Commits, PR reviews, tags
* Manual Author: README + user guide
* Presenter: Live demo script and slide deck
* AI-Reflection Writer: AI decisions, prompts, failures, mitigations

## Minimum Feature Checklist

- Search/filter tutors  
- Create/edit tasks; mark complete; progress %  
- Book/cancel a session (mock ok)  
- At least one AI feature in UI  
- Lighthouse performance ≥ 85 desktop  
- No major console errors; clean 404/error states

## Stretch Goals

- Calendar sync (iCal export)  
- Real auth with protected routes  
- Realtime chat  
- Analytics page  
- AI: 'Explain this concept' mode

## Assessment Rubric (100 pts)

Functionality & Stability (25)  
UX & Accessibility (20)  
Code Quality (15)  
AI Integration (20)  
Deployment & Demo (10)  
Team Process (10)

## What to Submit

- Repo link (clean README)  
- Deployed URL  
- /docs folder: proposal, API docs, a11y report, AI-reflection, slides  
- 5–7 minute demo video (optional)