# **טכנולוגיות אינטרנט מתקדמות - 61776 (WEB)**

משימת בית מס' 1

Group 13

StudyBuddy

System Engineer: Ahmad Tawil

As the appointed System Engineer, I coordinated all tasks for the team during the StudyBuddy project sprints. Below is a breakdown of each member's responsibilities and progress.

| Team Member | Assigned Tasks | Completed Tasks |
| --- | --- | --- |
| Ahmad Tawil | Sprint 1: Setup full structure, layout.js, Navbar/Footer  Sprint 2: Review and support integration  Sprint 3: Debugging, assist AI UI logic  Sprint 4: Final documentation, Git cleanup | All |
| Adam | Sprint 1: Set up ask/page.js with placeholder  Sprint 2: Build AskForm.js with upload  Sprint 3: Connect rephraseQuestion() from AI  Sprint 4: Polish form + error UX | All |
| Ibrahim | Sprint 1: Setup requests/page.js layout  Sprint 2: Create RequestFeed.js using Card.js  Sprint 3: Add filters + loading state  Sprint 4: Review for mobile & accessibility | All |
| Omar | Sprint 1: Setup profile/page.js structure  Sprint 2: Develop ProfileView.js with data  Sprint 3: Add stats, badges display  Sprint 4: Polish styling | All |
| Marwa | Sprint 1: Setup profile/page.js structure  Sprint 2: Develop ProfileView.js with data  Sprint 3: Add stats, badges display  Sprint 4: Polish styling | All |
| Cyrine | Sprint 1: Build firebase.js and authContext.js  Sprint 2: Create Firestore helpers  Sprint 3: Test rules + login logic  Sprint 4: Security finalization | All |

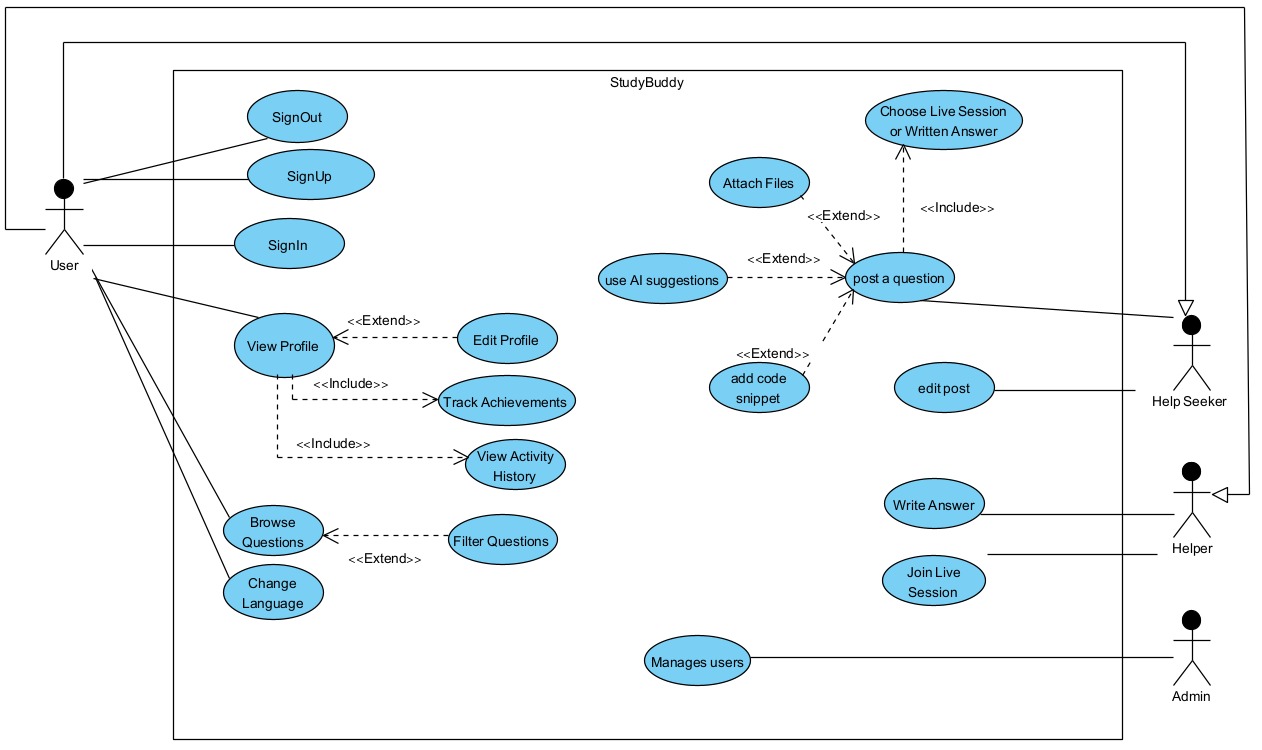
Interaction and Collaboration:

Team members coordinated through daily stand-ups and GitHub pull requests. Most work was divided by page/component ownership, with integration and testing occurring in later sprints. Overlapping tasks ( Ask Page, Profile Page) were resolved through shared commits and pair programming sessions.

git-Link:<https://github.com/AhmadTawil1/study-buddy>

2) Web Architecture

<https://github.com/AhmadTawil1/study-buddy/blob/main/architecture/Editor%20_%20Mermaid%20Chart-2025-06-06-105814.png>

3) Use Case

5) 3 שאלות על החומר הנלמד עד כה:

### **שאלה 1: מה ההבדל בין useState לבין useContext בריאקט?**

**תשובה:**

* useState משמש לניהול מצב פנימי מקומי בתוך קומפוננטה אחת (כמו טופס או כפתור).
* useContext משמש לשיתוף מצב גלובלי בין קומפוננטות שונות, דרך ספק הקשר (כמו authContext, userContext).

### **שאלה 2: מה ההבדל בין ניהול נתיבי גלישה (Routing) ב-React לעומת ב-Next.js?**

**תשובה:**

* ב-React יש להגדיר את הנתיבים ידנית בעזרת react-router-dom.
* ב-Next.js הנתיבים נוצרים באופן אוטומטי לפי שמות הקבצים בתיקיית app/ או pages/.  
   לדוגמה: הקובץ app/profile/page.js מייצג את הנתיב /profile.

### **שאלה 3: למה להשתמש ב-API Routes של Next.js במקום בשרת Backend נפרד?**

**תשובה:**

* API Routes מאפשרים להוסיף לוגיקה בצד השרת בתוך אותו פרויקט Next.js, ללא צורך בשרת חיצוני.
* זה חוסך זמן בפיתוח, מתאים במיוחד לפרויקטים קטנים עד בינוניים, ומקל על פריסה (למשל עם Vercel).