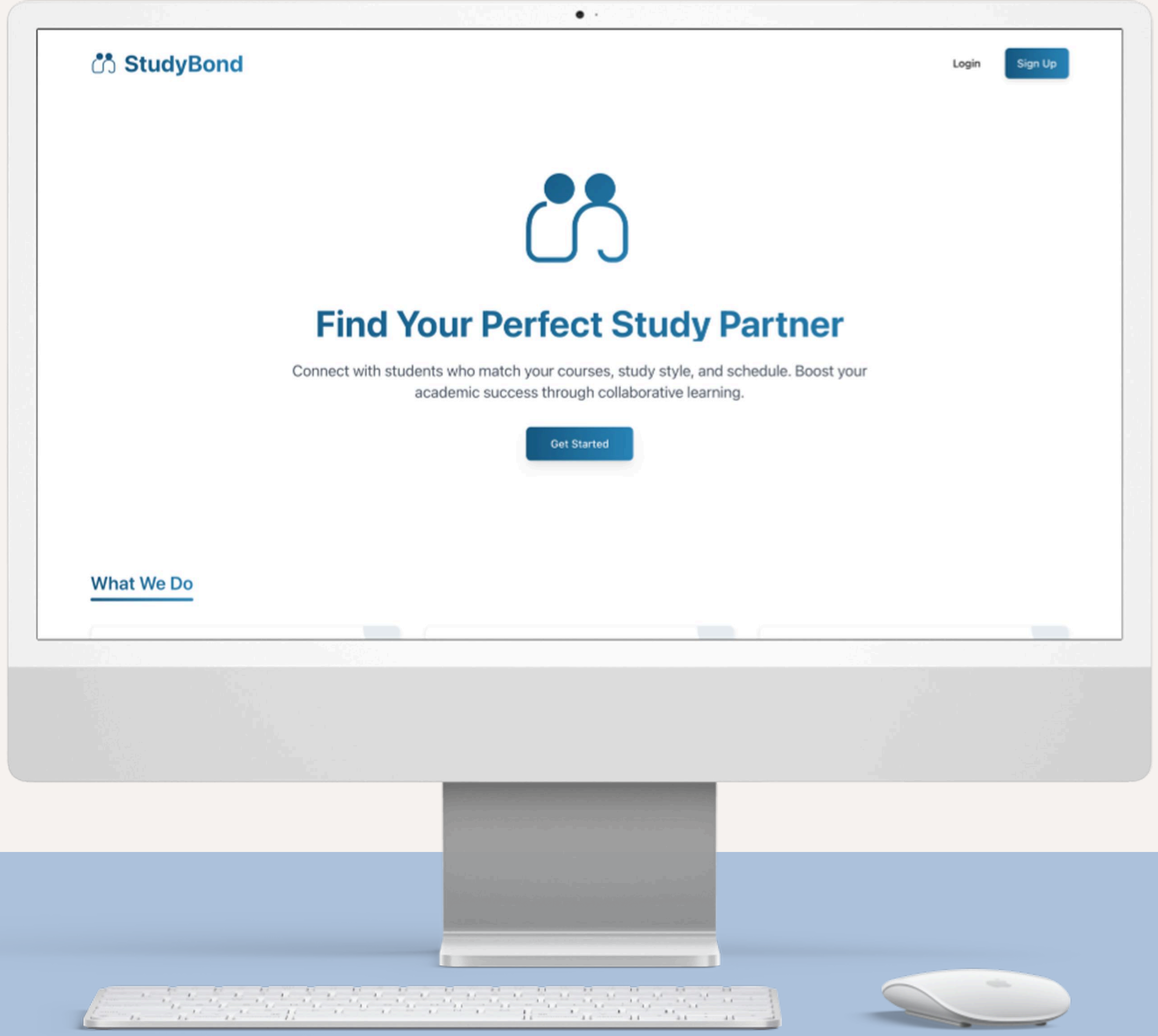
IT 485 Capstone Final Presentation Pitch



Find Your Perfect Study Partner

Connect with students who match your courses, study style, and schedule. Boost your academic success through collaborative learning.

StudyBond

A more efficient way to study & bond with UMB students

Meet the team

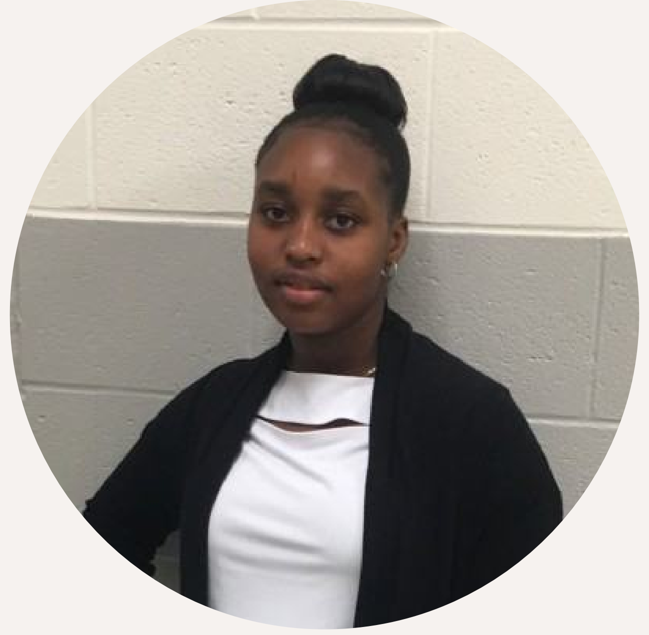
Benson Li
Product Manager

Mohamed ElDamaty
Software Developer

Swarnim Pradhan
UX Designer

Michael Agbesi
UX Designer

Kareem Brathwaite
Marketing Analyst

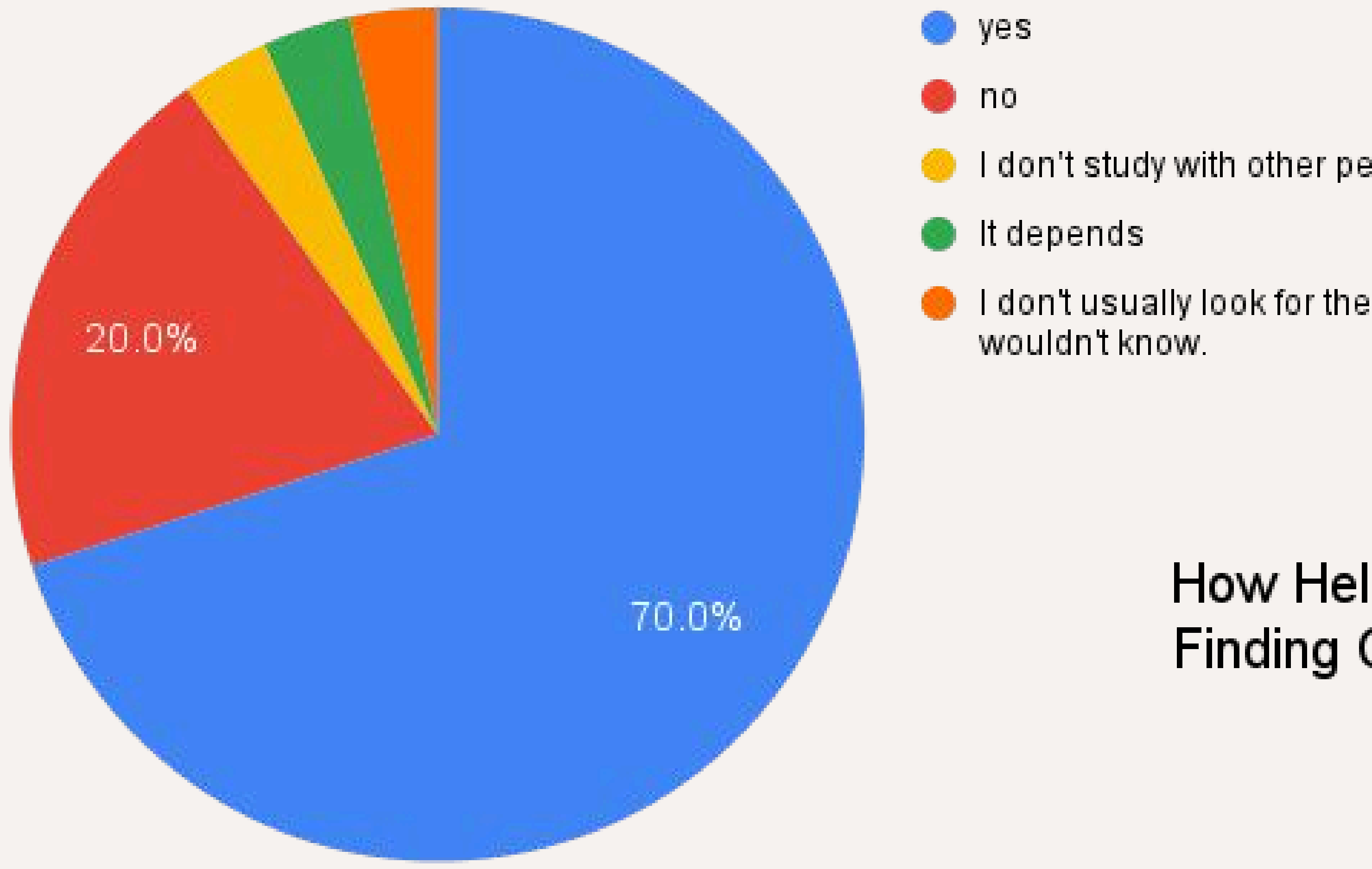


Devinne Wilson
User Researcher

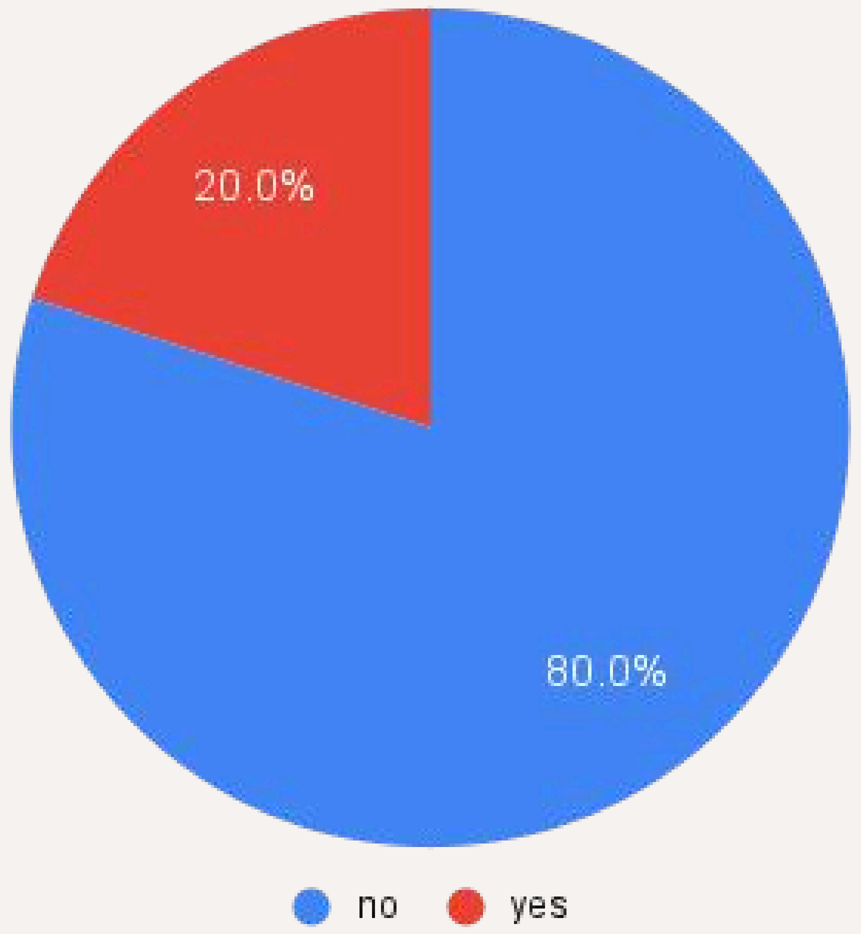
The Problem

We started with one question: why is it so hard to find a study partner at UMB?

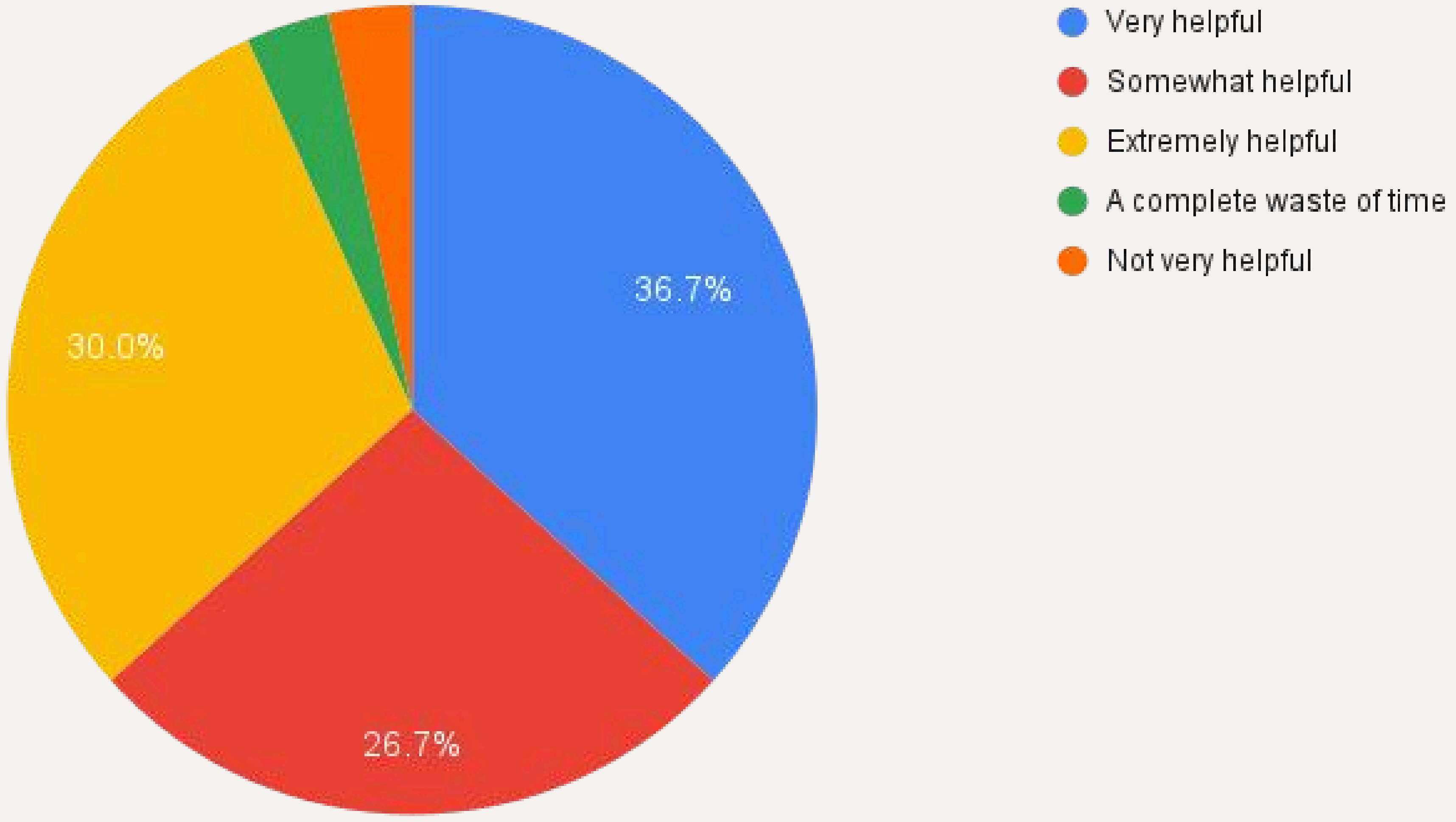
Do you Find it Challenging to Find Study Partners?



Do you Have a Specific Approach to Find Study Partners?



How Helpful Do You Think a Platform That Assists UMB Students With Finding Compatible Study Partners Based On Factors Like Personality,



We surveyed 50 UMB Students

The Problem

- **70%** of UMB students think it’s difficult to find a study partner
- **80%** aren’t even sure how to go about finding one

The Commuter Challenge

Many students leave campus right after class, missing out on chances to form study connections.

Scheduling Struggles

Class times and personal schedules rarely align, making it difficult to find study partners with overlapping availability.

The Connection Gap

Students often don’t know where to start when looking for reliable, like-minded study partners.

Our Solution

StudyBond helps UMass Boston students connect with study partners who fit their learning styles and schedules, making study sessions more productive.

By matching students based on shared courses, availability, and study preferences, it removes the friction of finding reliable partners.

Competitive Audit

There are so many social media platforms

But none are *perfect* for finding a study partner at UMass Boston.

Platforms like Instagram, Discord, and GroupMe aren’t designed for academic matching

How it works

- Exclusively for UMass Boston students — sign in with your @umb.edu email
- Smart matching based on your courses, schedule, and study style
- Seamless scheduling to set up focused 1:1 sessions without the group chat noise

1

Create an account with UMB email

2

Enter your account and preferences

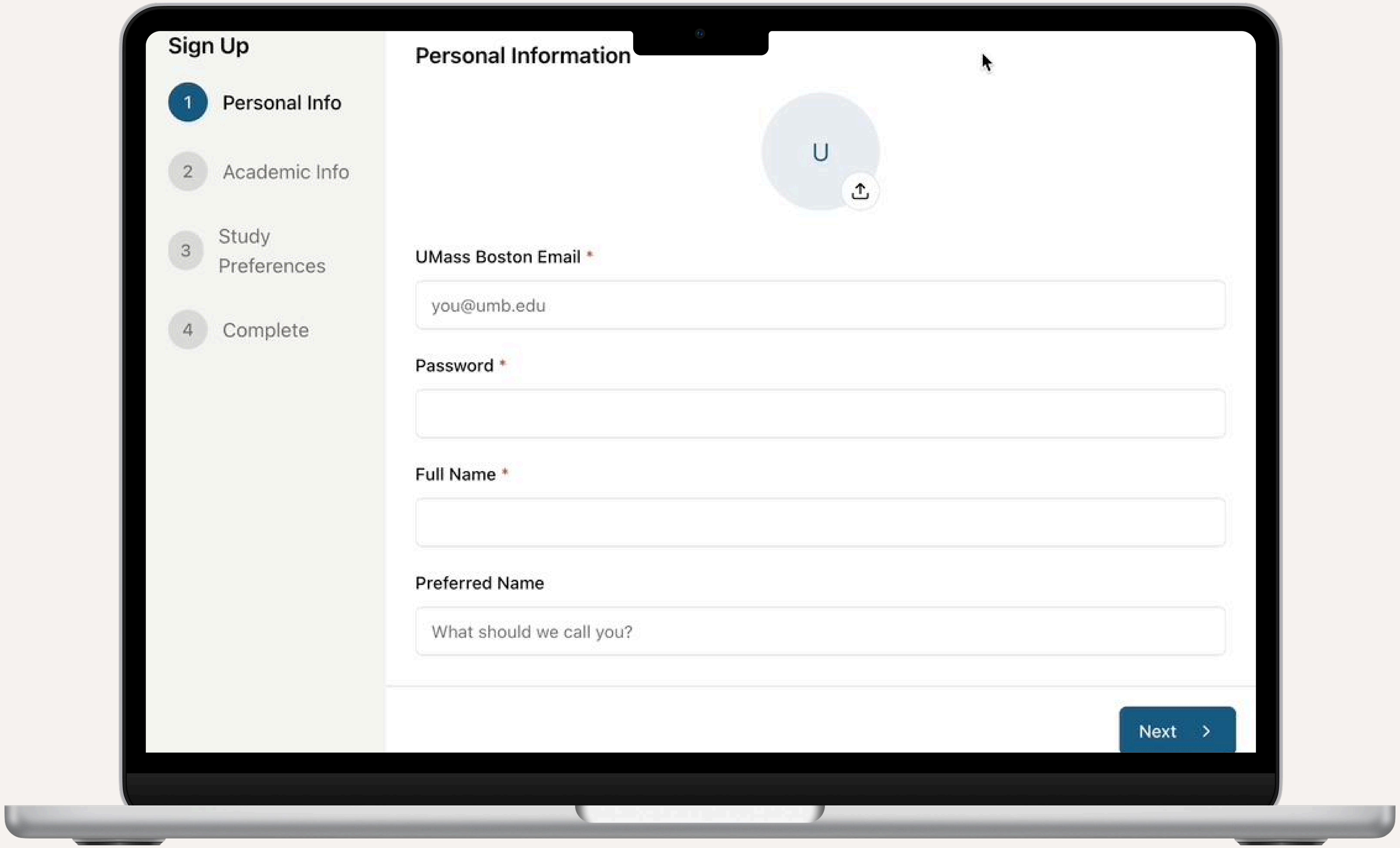
3

Match with compatible study partners

4

Schedule and manage Study Sessions

SIGN UP



Sign Up

1 Personal Info

2 Academic Info

3 Study Preferences

4 Complete

Personal Information

U

UMass Boston Email *

you@umb.edu

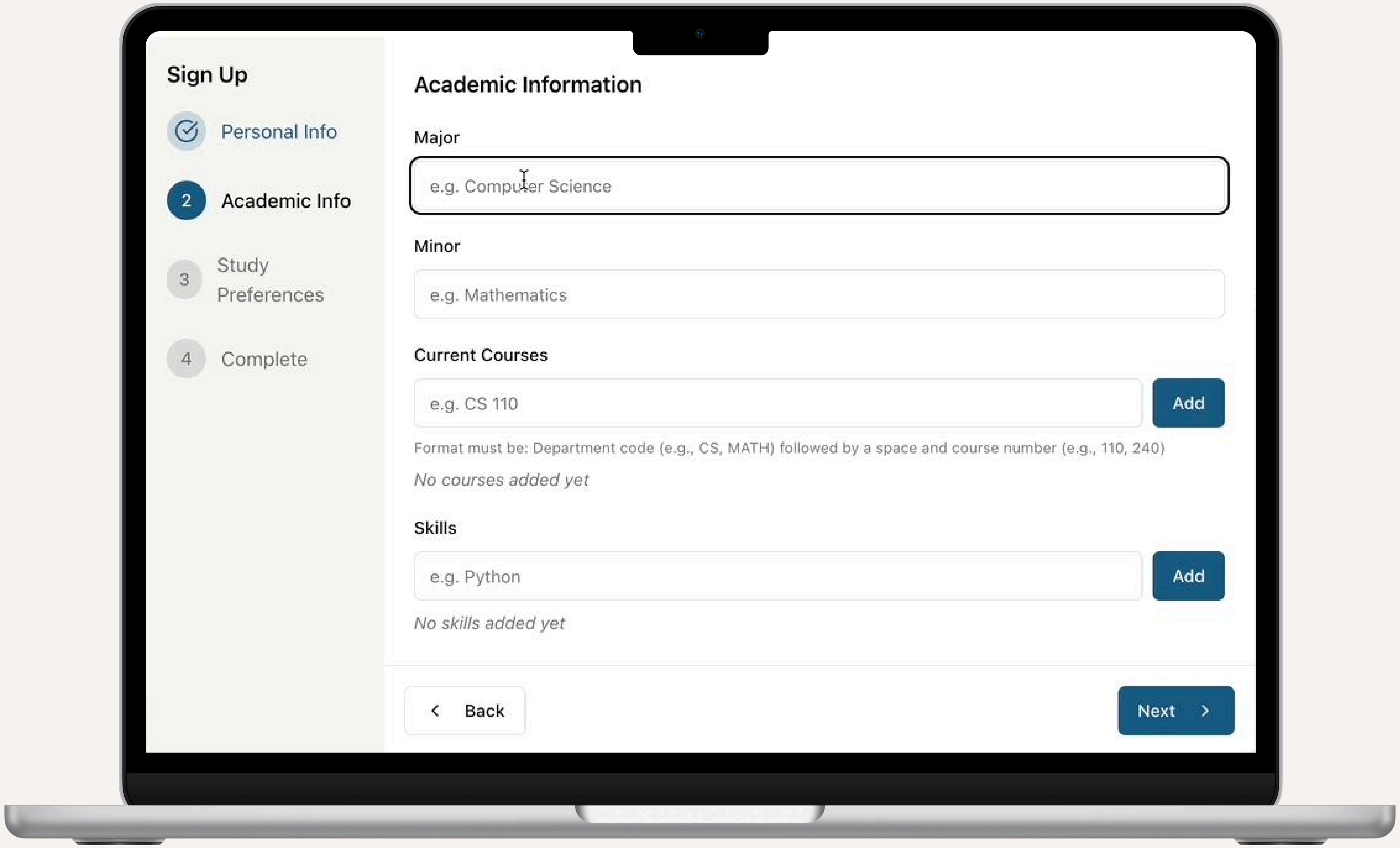
Password *

Full Name *

Preferred Name

What should we call you?

Next >



Sign Up

- ✓ Personal Info
- 2 Academic Info
- 3 Study Preferences
- 4 Complete

Academic Information

Major

e.g. Computer Science

Minor

e.g. Mathematics

Current Courses

e.g. CS 110 | Add

Format must be: Department code (e.g., CS, MATH) followed by a space and course number (e.g., 110, 240)

No courses added yet

Skills

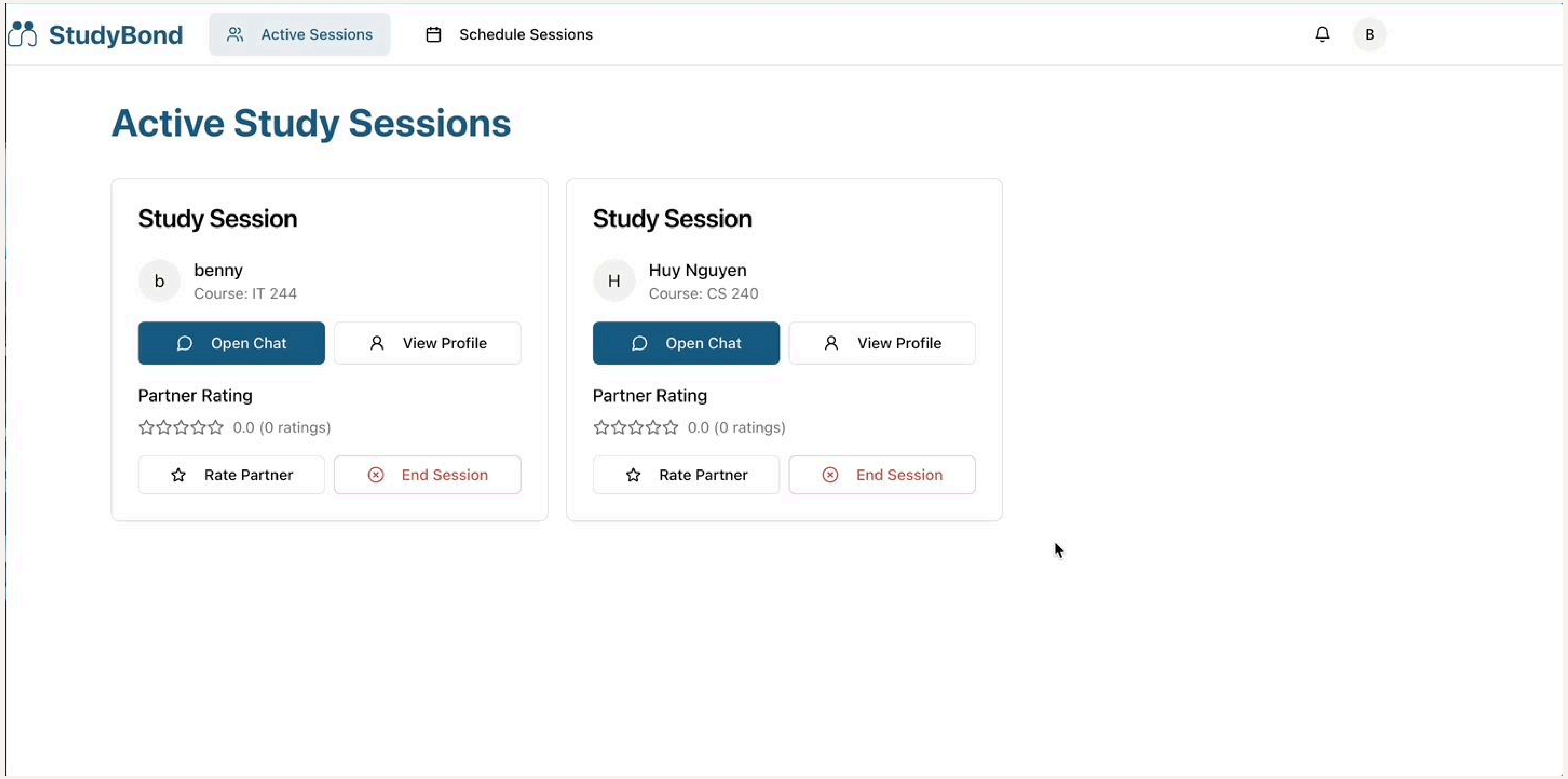
e.g. Python | Add

No skills added yet

< Back | Next >

SIGN UP

MATCH



Active Study Sessions

Study Session

b benny

Course: IT 244

Partner Rating

☆☆☆☆☆ 0.0 (0 ratings)

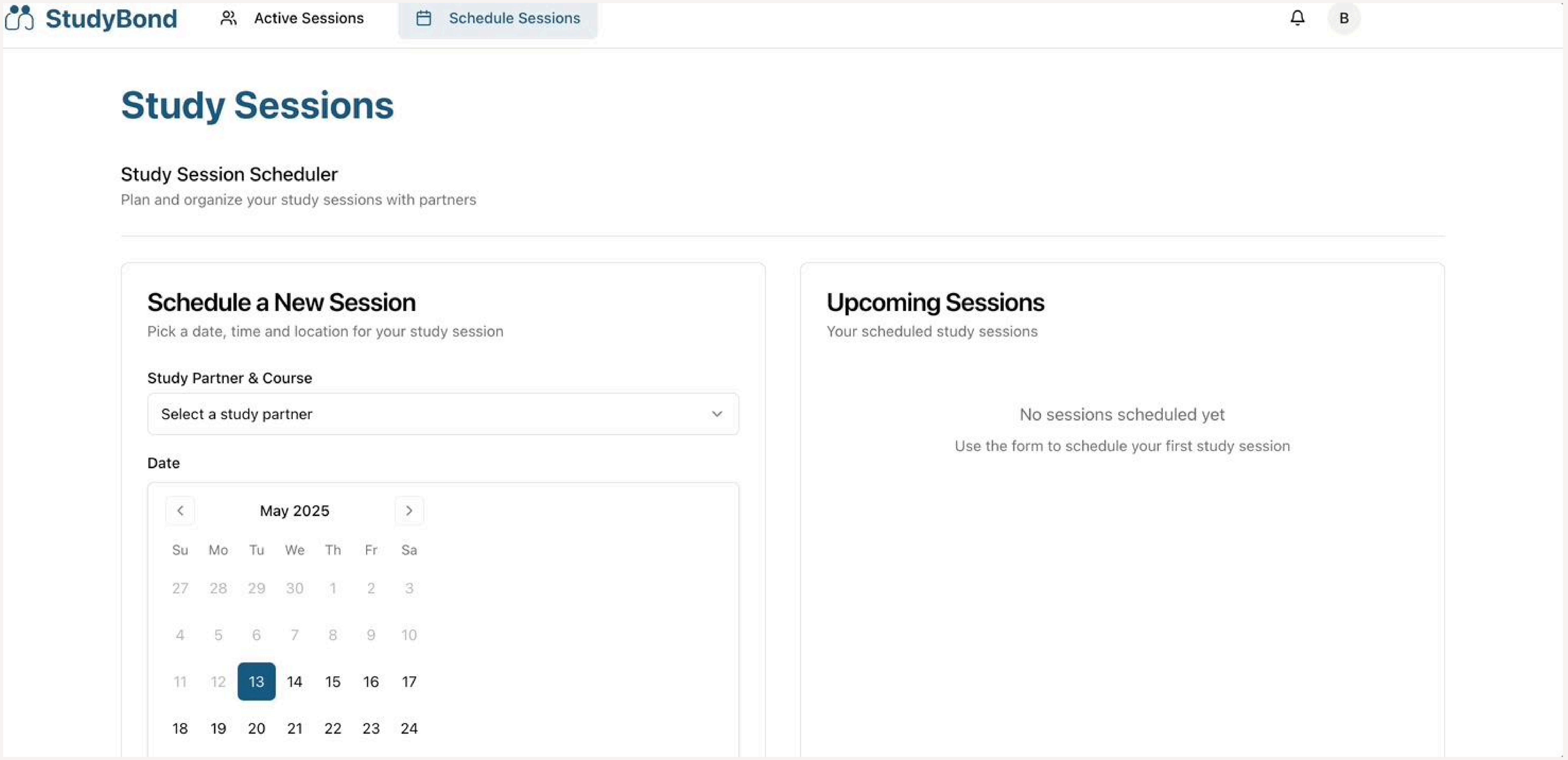
Study Session

H Huy Nguyen

Course: CS 240

Partner Rating

☆☆☆☆☆ 0.0 (0 ratings)



Study Sessions

Study Session Scheduler

Plan and organize your study sessions with partners

Schedule a New Session

Pick a date, time and location for your study session

Study Partner & Course

Select a study partner

Date

May 2025						
Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24

Upcoming Sessions

Your scheduled study sessions

No sessions scheduled yet

Use the form to schedule your first study session

SCHEDULE

Marketing Opportunity

Post-COVID academic gap:
Students need stronger peer-to-peer support as classes return in-person.

Lack of tailored tools:
No existing study platforms are built specifically for UMB's structure or student body.

Demand for connection:
Students are actively seeking simpler, low-stakes ways to find study partners.

Loneliness epidemic:
According to Active Minds and TimelyCare's 2024 national student mental health survey, Nearly two-thirds (64.7%) of college students report feeling lonely.

Revenue Model

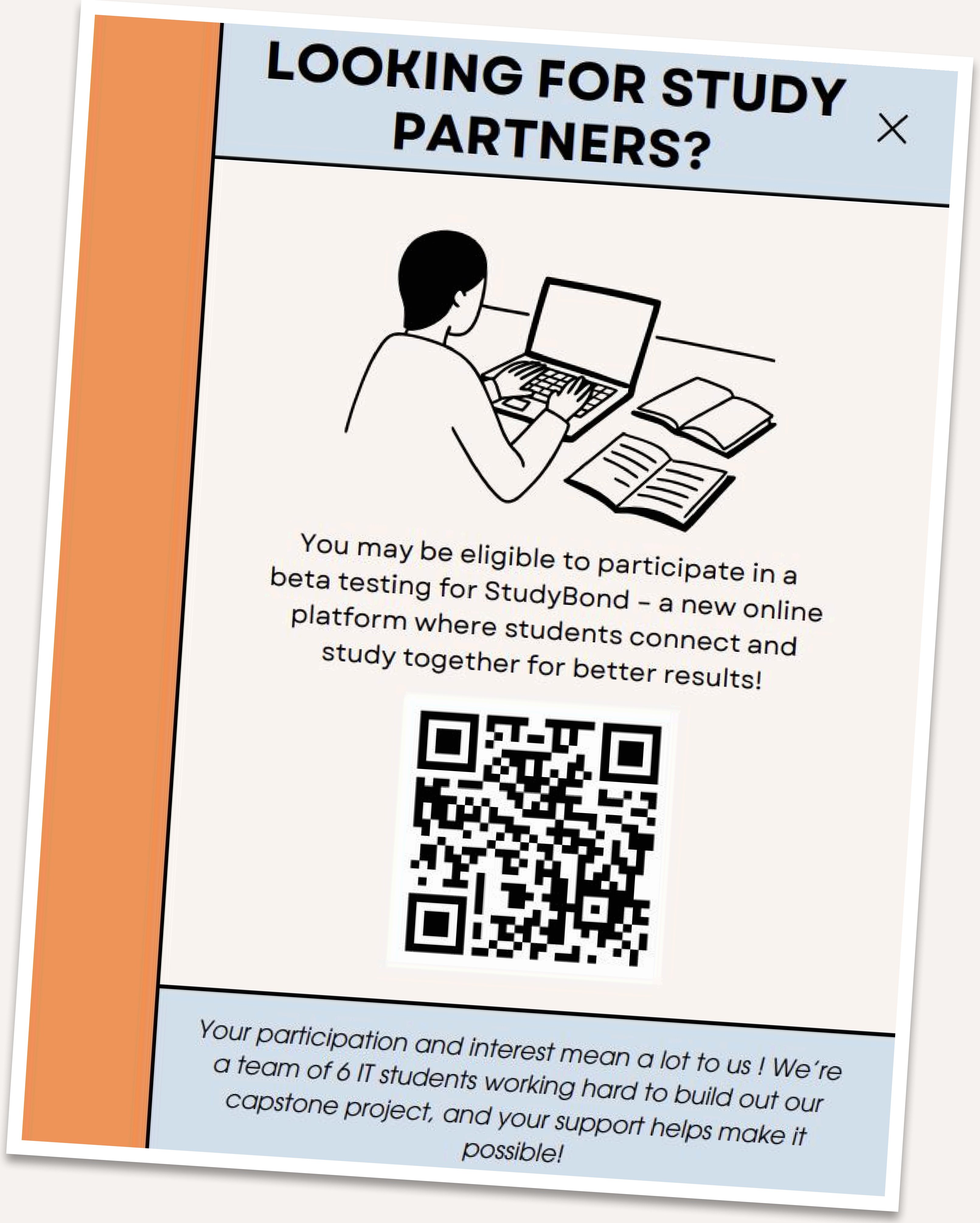
- **Student-first model:** Core features will always be free to ensure every UMB student can access academic support
- **Freemium upgrades:** Unlock smarter matching, advanced scheduling tools, and session analytics with a premium tier
- **Branded educational events:** Sponsored workshops and campus activations with mission-aligned brands

Long-term

- **Ad-supported growth:** Run contextual ads from vetted education and productivity brands
- **Insight-driven partnerships:** Share anonymized usage data with academic partners to inform service design
- **Campus SaaS licensing:** License the platform to other universities seeking scalable, peer-to-peer academic support solutions

Real Demand & Real Feedback

- Launched a closed beta with **20 UMB students** across **5 majors**
- Promoted solely through campus fliers — no email blasts, no paid ads
- Iterated weekly based on real student feedback and behavior
- **3 successful organic matches** within the first week
- Feedback led to **improvements** in onboarding and match relevance



LOOKING FOR STUDY PARTNERS?

You may be eligible to participate in a beta testing for StudyBond - a new online platform where students connect and study together for better results!

Your participation and interest mean a lot to us ! We're a team of 6 IT students working hard to build out our capstone project, and your support helps make it possible!

Buddy: Studybond's Built-in AI

Built for studying, not chatting Buddy isn't a generic GPT — it's designed to support how students actually study.

Study Buddy

Knows what you're working on and adapts to your study habits.

Topic Analyzer

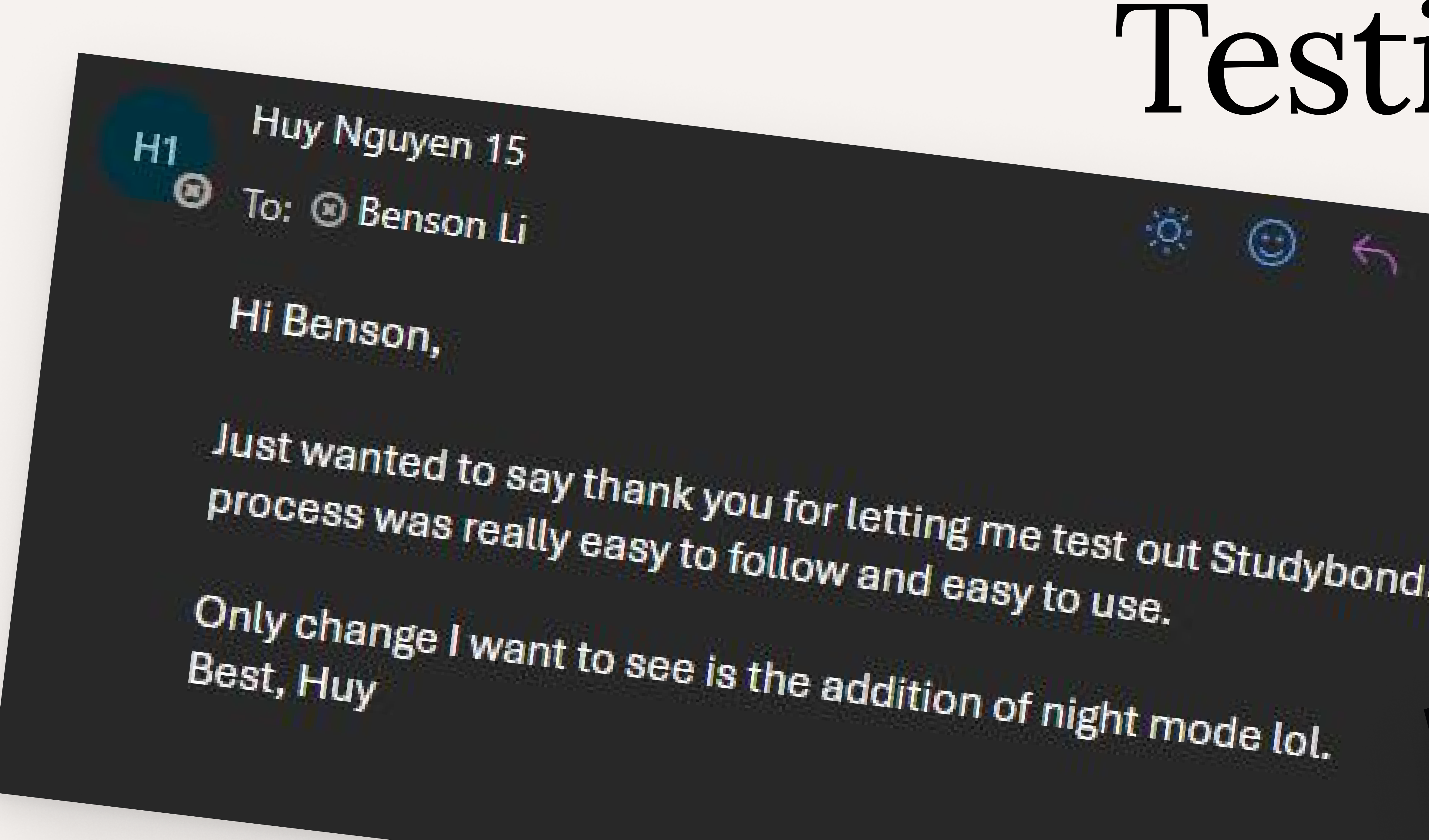
Breaks down a topic and surfaces relevant resources to start with.

Context-aware

Knows what you're working on and adapts to your study habits.

Integrated into StudyBond: No switching tabs or tools — Buddy works alongside you, right on the app.

Testimonials



H1

Huy Nguyen 15

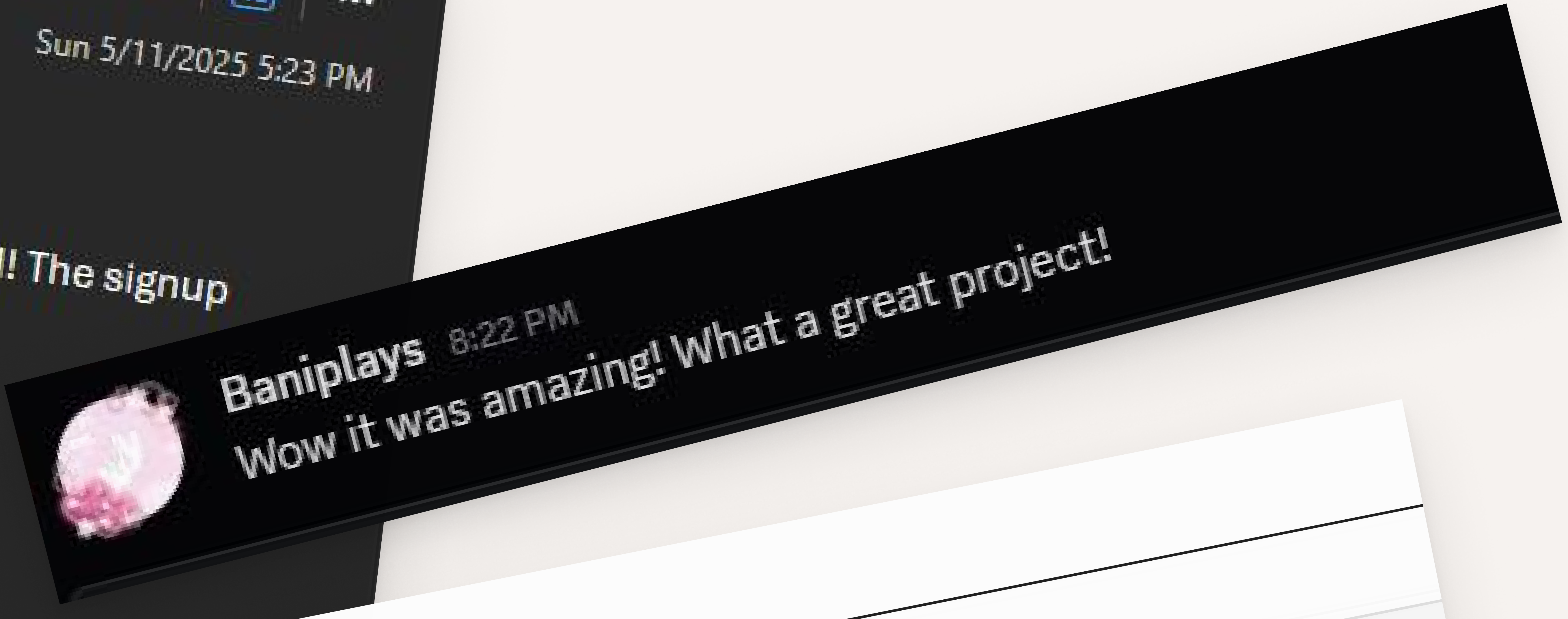
To: Benson Li

Sun 5/11/2025 5:23 PM

Hi Benson,

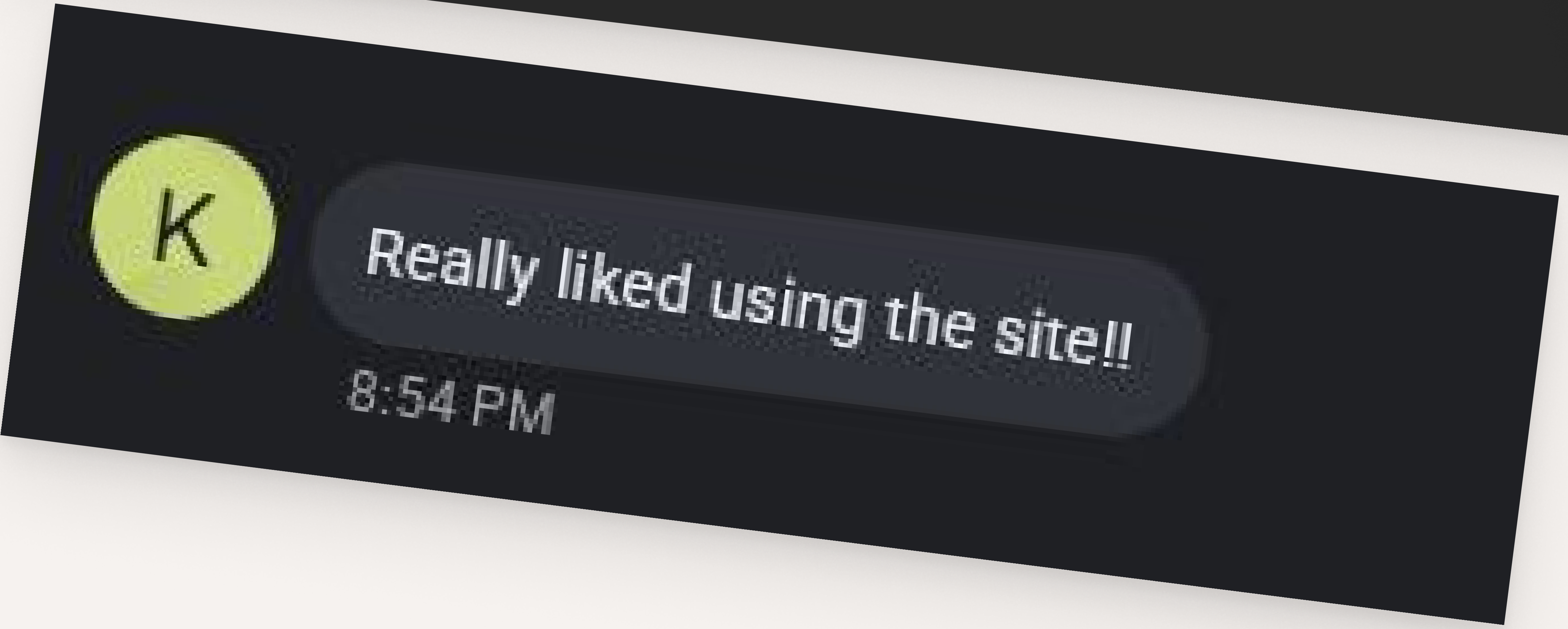
Just wanted to say thank you for letting me test out Studybond! The signup process was really easy to follow and easy to use.

Only change I want to see is the addition of night mode lol.
Best, Huy



Baniplays 8:22 PM

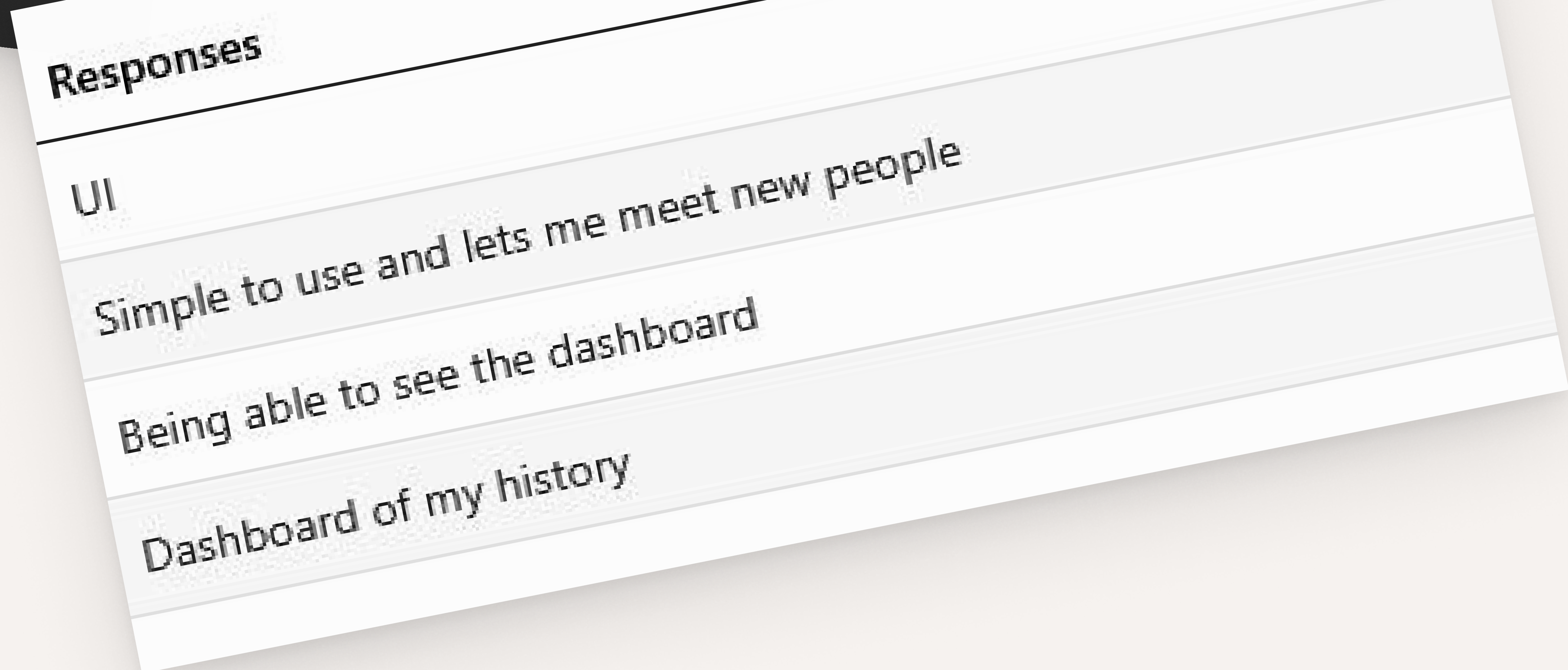
Wow it was amazing! What a great project!



K

Really liked using the site!!

8:54 PM



Responses
UI
Simple to use and lets me meet new people
Being able to see the dashboard
Dashboard of my history

Technical Stack

Frontend

- React 18 + TypeScript for robust UI; Tailwind CSS + Shadcn UI for styling; Framer Motion & TanStack Query for smooth UX and performant data handling; Forms with React Hook Form + Zod.

Backend

- Express.js server with RESTful APIs, real-time WebSocket support, and secure session-based auth via Passport.js. Multer handles uploads; Zod enforces validation.

Dev & Infra Built

- Built with Vite + ESBuild; sessions stored in-memory for dev, environment-based configs, and full CI-ready deployment pipeline.

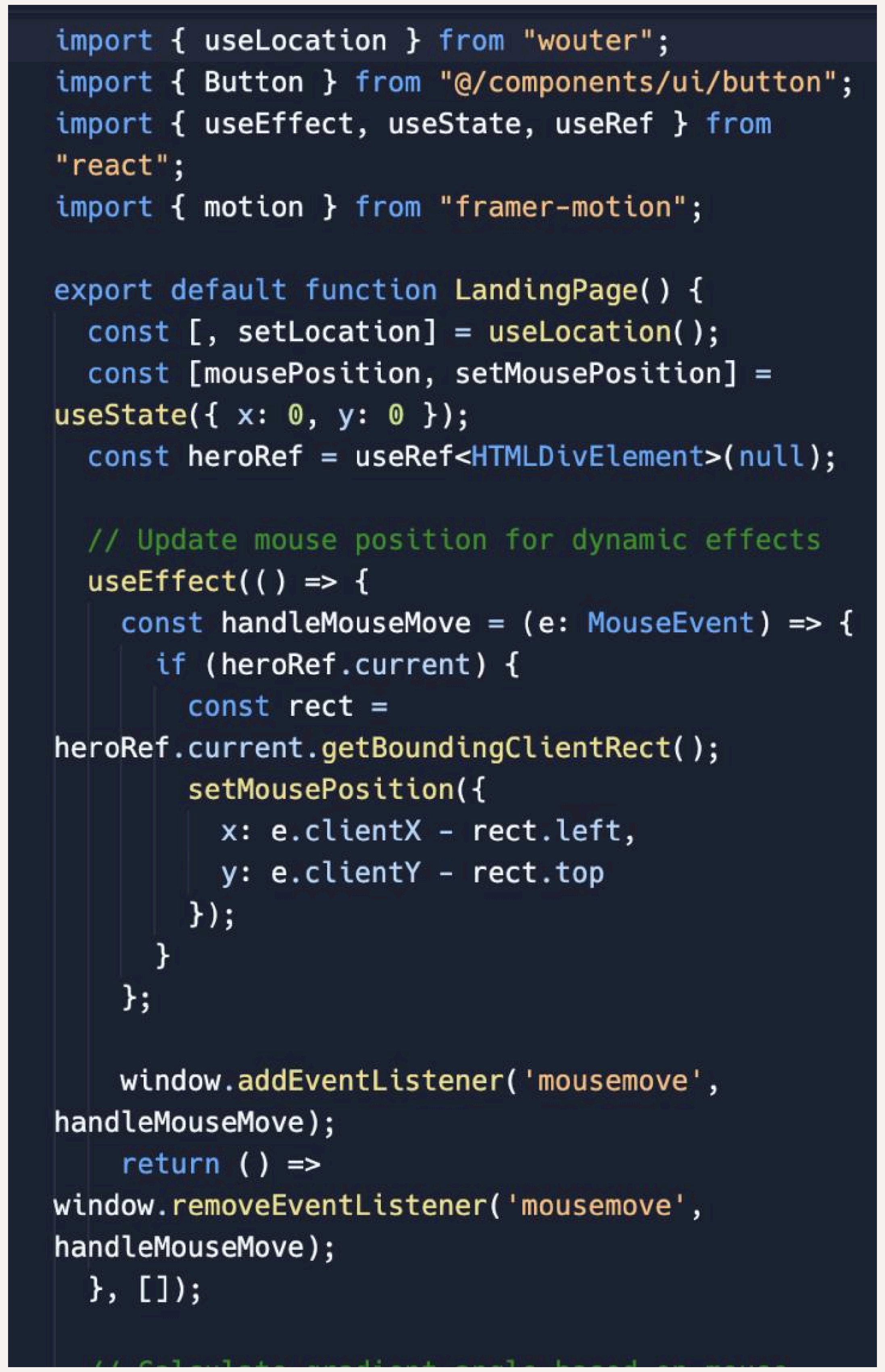
Database

- PostgreSQL with Drizzle ORM for type-safe queries, schema migrations, and advanced features (arrays, JSON types, foreign keys, timestamps).

Authentication

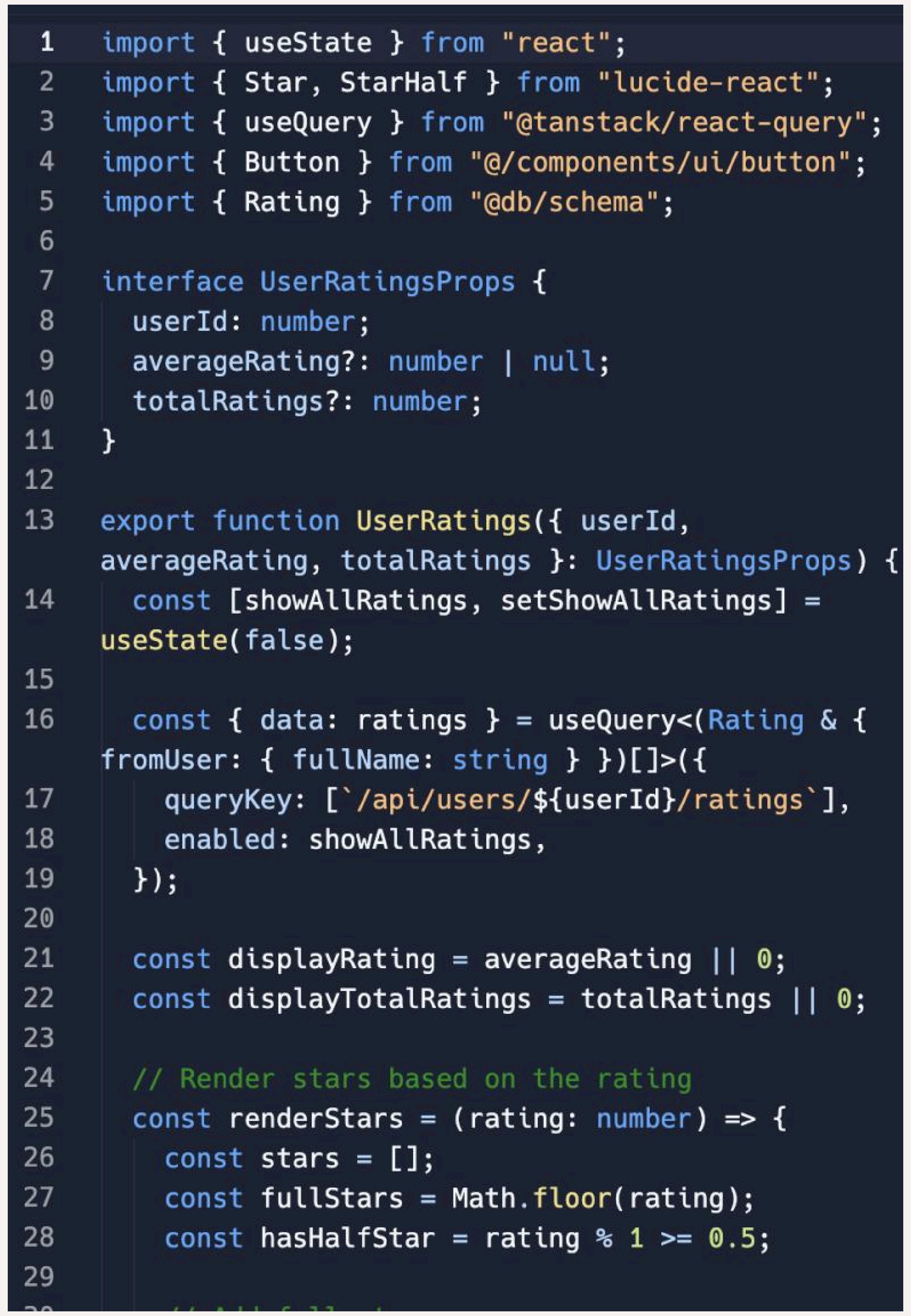
- OpenAI API powers Buddy AI with real-time chat, topic breakdown, and context-aware responses.

Code Snippets



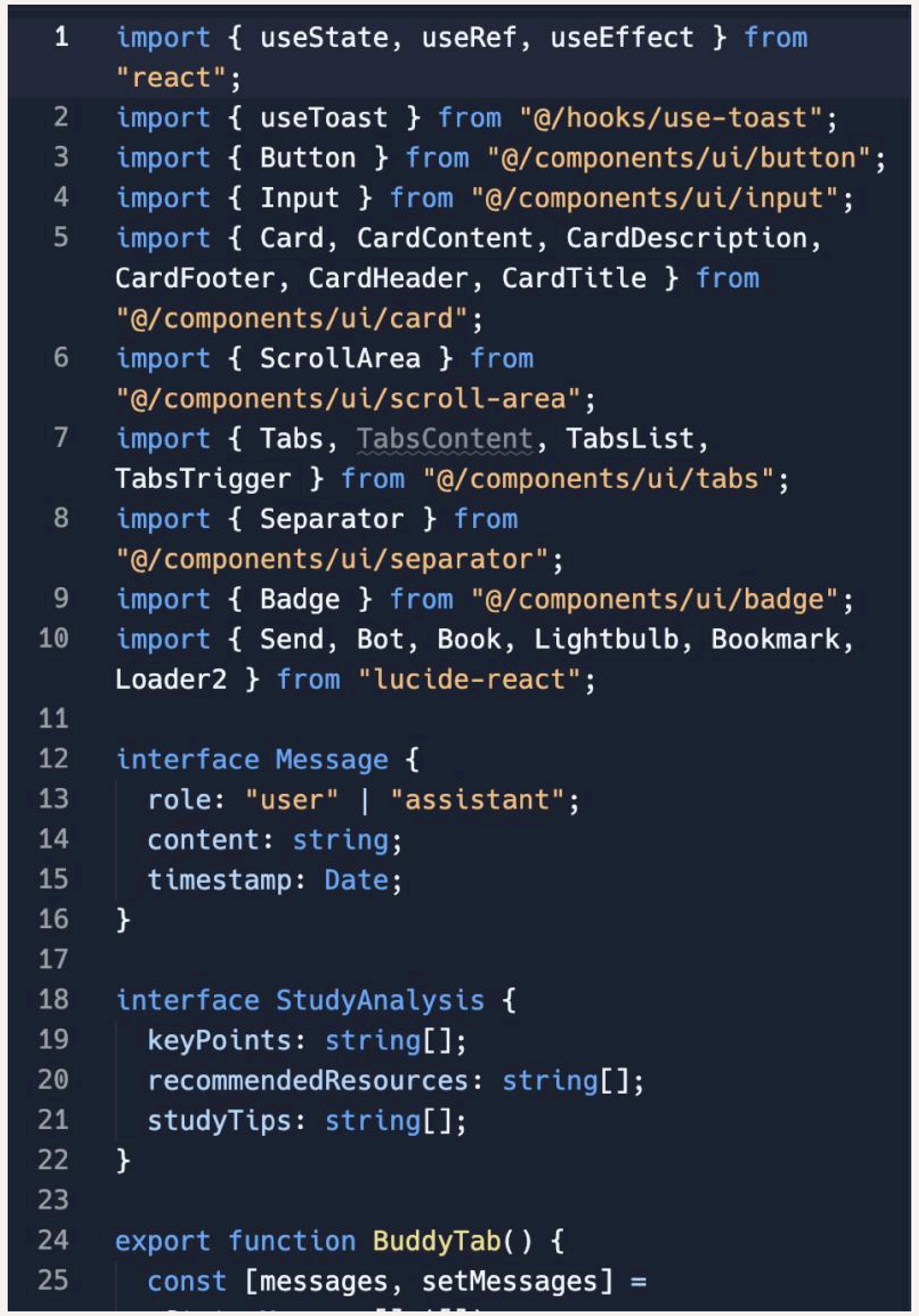
```
import { useLocation } from "wouter";
import { Button } from "@/components/ui/button";
import { useEffect, useState, useRef } from
"react";
import { motion } from "framer-motion";

export default function LandingPage() {
  const [, setLocation] = useLocation();
  const [mousePosition, setMousePosition] =
useState({ x: 0, y: 0 });
  const heroRef = useRef<HTMLDivElement>(null);

  // Update mouse position for dynamic effects
  useEffect(() => {
    const handleMouseMove = (e: MouseEvent) => {
      if (heroRef.current) {
        const rect =
heroRef.current.getBoundingClientRect();
        setMousePosition({
          x: e.clientX - rect.left,
          y: e.clientY - rect.top
        });
      }
    };

    window.addEventListener('mousemove',
handleMouseMove);
    return () =>
window.removeEventListener('mousemove',
handleMouseMove);
  }, []);
```

Dynamic UI



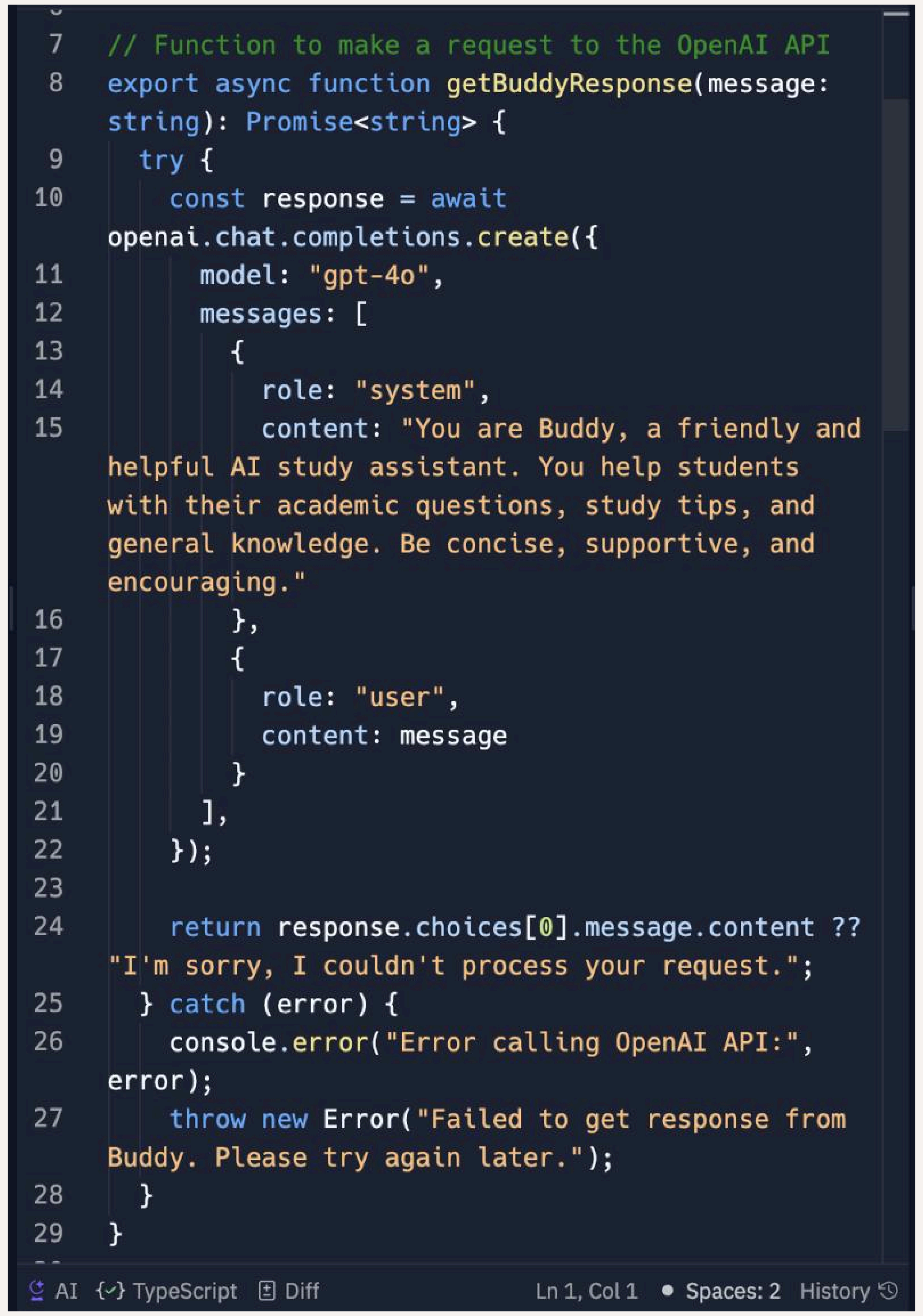
```
import { useState } from "react";
import { Star, StarHalf } from "lucide-react";
import { useQuery } from "@tanstack/react-query";
import { Button } from "@/components/ui/button";
import { Rating } from "@db/schema";

interface UserRatingsProps {
  userId: number;
  averageRating?: number | null;
  totalRatings?: number;
}

export function UserRatings({ userId,
averageRating, totalRatings }: UserRatingsProps) {
  const [showAllRatings, setShowAllRatings] =
useState(false);

  const { data: ratings } = useQuery<(Rating & {
fromUser: { fullName: string } })[]>({
    queryKey: [`/api/users/${userId}/ratings`],
    enabled: showAllRatings,
  });

  const displayRating = averageRating || 0;
  const displayTotalRatings = totalRatings || 0;

  // Render stars based on the rating
  const renderStars = (rating: number) => {
    const stars = [];
    const fullStars = Math.floor(rating);
    const hasHalfStar = rating % 1 >= 0.5;

```

User Ratings



```
import { useState, useRef, useEffect } from
"react";
import { useToast } from "@/hooks/use-toast";
import { Button } from "@/components/ui/button";
import { Input } from "@/components/ui/input";
import { Card, CardContent, CardDescription,
CardFooter, CardHeader, CardTitle } from
"@/components/ui/card";
import { ScrollArea } from
"@/components/ui/scroll-area";
import { Tabs, TabsContent, TabsList,
TabsTrigger } from "@/components/ui/tabs";
import { Separator } from
"@/components/ui/separator";
import { Badge } from "@/components/ui/badge";
import { Send, Bot, Book, Lightbulb, Bookmark,
Loader2 } from "lucide-react";

interface Message {
  role: "user" | "assistant";
  content: string;
  timestamp: Date;
}

interface StudyAnalysis {
  keyPoints: string[];
  recommendedResources: string[];
  studyTips: string[];
}

export function BuddyTab() {
  const [messages, setMessages] =
```

Tab Components



```
// Function to make a request to the OpenAI API
export async function getBuddyResponse(message:
string): Promise<string> {
  try {
    const response = await
openai.chat.completions.create({
      model: "gpt-4o",
      messages: [
        {
          role: "system",
          content: "You are Buddy, a friendly and
helpful AI study assistant. You help students
with their academic questions, study tips, and
general knowledge. Be concise, supportive, and
encouraging."
        },
        {
          role: "user",
          content: message
        }
      ],
    });

    return response.choices[0].message.content ??
"I'm sorry, I couldn't process your request.";
  } catch (error) {
    console.error("Error calling OpenAI API:",
error);
    throw new Error("Failed to get response from
Buddy. Please try again later.");
  }
}
```

Buddy AI

Live Demonstration

Thank you!

Any Questions?

- Special thank you to Professor Banjo for his support this semester!

← Sign up for our interest form here