Timemaxx

An AI-Powered Study Scheduler for Johns Hopkins Students

Anna Mintzer, Anvii Mishra, Maria Romo Nichols, Ria Talwar, Tanvi Ranade, Aviel Rosen
— Johns Hopkins University, Whiting School of Engineering — EN.601.423

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

The Problem

JHU students juggle dozens of assignments across multiple courses each semester with no reliable way to estimate how long each task will take or when to work on them.

Generic calendar apps don't understand academic workload. Students consistently underestimate complex tasks — leading to cramming, missed deadlines, and burnout.

Existing tools require manual entry of every assignment and offer no intelligence about task duration or optimal scheduling.

Tech Stack & Architecture

- **Frontend: React + TypeScript + Vite + TailwindCSS**
- **Backend: Convex (serverless DB, queries, mutations, actions), Mastra (agentic workflows)**
- **Auth: Google OAuth via Convex Auth**
- **AI Models: Claude Sonnet 4 (primary LLM), Gemini embeddings**
- **Chat: Mastra agent framework on a separate Node.js server with 4 specialist agents**
- **Calendar: Google Calendar API (OAuth 2.0) for read, write, conflict detection, and auto-export**
- **Data Sources: JHU SIS API for course search, user transcripts and syllabi with pdfjs-dist for client-side PDF parsing**

Core Capabilities

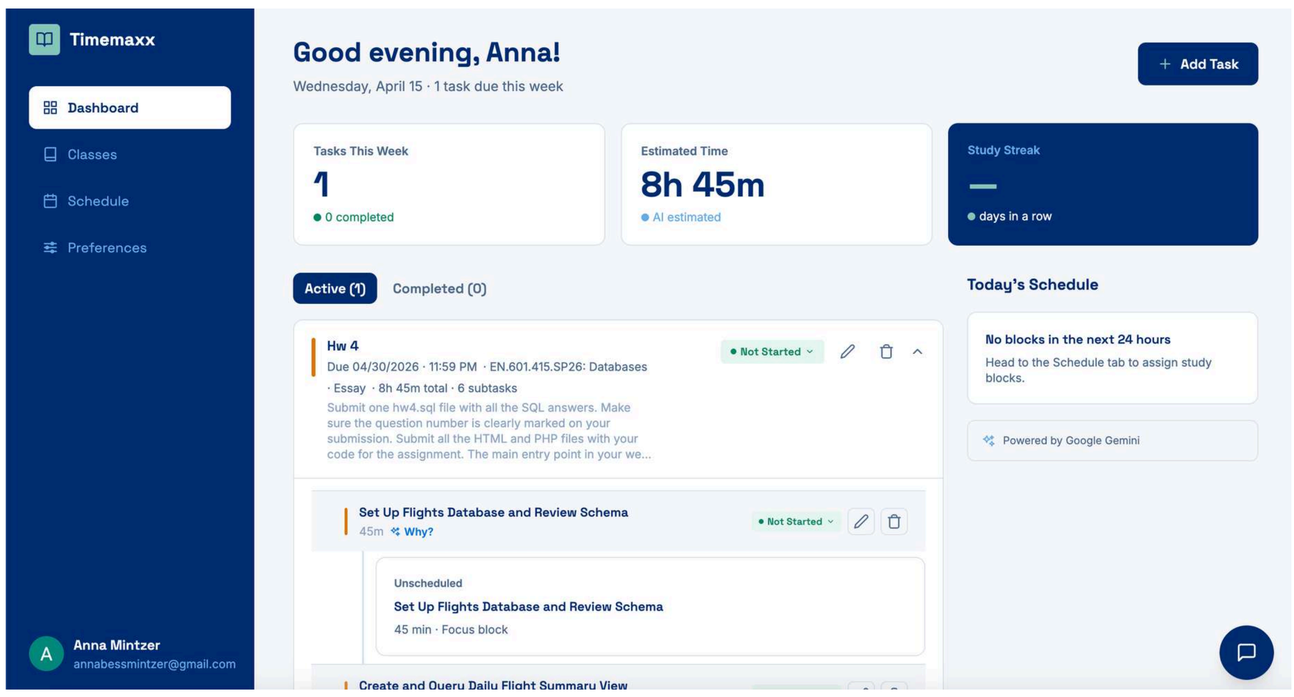
1

SYLLABUS PARSING
- **Upload a PDF syllabus and an LLM extracts every assignment with title, type, and due date automatically. Users review and confirm before tasks are created.**

RAG-POWERED TIME ESTIMATION
- **Claude Sonnet 4 predicts task duration using retrieval-augmented context: similar past tasks, course difficulty, student profile, and chat history.**

SMART SCHEDULING
- **AI ranks candidate time slots against Google Calendar events, work-hour preferences, and productivity traits — then places study blocks automatically.**

CONVERSATIONAL AGENT (MAXX)
- **A multi-agent chat assistant lets students create tasks, log feedback, and update preferences through natural language.**



Objectives

Build a system that automatically extracts syllabus assignments and estimates task duration using AI.

Generate personalized study schedules based on each student's availability and productivity patterns.

Create a feedback loop so time estimates improve over time with use.

3

THE AI PIPELINE

1. **Extraction: An LLM parses syllabus and transcript PDFs into structured task and course data.**
2. **Embeddings: All tasks, courses, and chat chunks are embedded using Gemini's embedding model for semantic retrieval.**
3. **Retrieval-Augmented Estimation: For each new task, the system retrieves the 10 most similar past tasks, 10 related courses, relevant chat excerpts, and user profile — then sends it all to Claude Sonnet 4 for a time estimate with reasoning trace.**
4. **Feedback Loop: Students rate estimates and log actual time. This data feeds back into future RAG context, making estimates more personalized over time.**

Conclusion

AI can reliably extract and estimate academic work — combining Gemini for semantic search with Claude's RAG-based reasoning produces usable time estimates from raw syllabus PDFs.

Personalization improves over time — the feedback loop and embedding-based retrieval mean the system gets smarter the more a student uses it.

Conversational UX lowers the barrier — students manage workload through natural language instead of navigating forms.

Scan to check it out!