Focus Enhancer — Redesigning Campus Attention

1. Problem Statement

Modern students face an *"attention crisis"* – constant digital distractions fragment focus and leave them feeling busy but unproductive. Over the past decade, average attention spans have collapsed (from ~2.5 min to ~47 sec) due to endless notifications and social media. University students describe a form of **"brain rot"**: poor concentration, cognitive fatigue, and declining academic performance after binge-consuming fast-paced online content. In parallel, students juggle heavy workloads (courses, coding practice, projects, and internships) without a unified tool to plan priorities or track progress. Existing study apps or to-do lists are disjointed; there's **no single platform** that integrates coursework, coding goals, peer feedback, and reward mechanisms into one coordinated system. As a result, effort is invisible – students lack accountability or recognition for consistent study habits, and it's easy to game the system by falsifying progress.

- **Fragmented Attention & Overload:** Students are bombarded by short-form media and messages, leading to shrinking attention spans and mental fatigue. Deep focus on academic tasks is increasingly rare.
- **Reduced Productivity & Well-being:** Excessive screen time and multitasking correlate with lower productivity, poor decision-making, and even negative mood or anxiety. Many report being "always busy" yet falling behind.
- **Lack of Integrated Planning:** Coursework schedules, coding practice, project deadlines, and personal goals live in separate tools (notebooks, Google Calendar, various apps). There is *no unified, adaptive system* to optimize time across all these domains.
- **No Accountability or Feedback Loop:** Study effort (e.g. solving practice problems or revising notes) goes unnoticed by others. Without validation or penalties, students find it easy to skip tasks or cheat (e.g. copying solutions without learning). Peer input and faculty guidance on daily goals are seldom tracked.

These issues underscore the core problem: **Students have too many demands on their attention but no cohesive way to structure and verify their focus.** Traditional approaches (standard planners, rigid schedules, or solely intrinsic motivation) aren't enough in today's attention economy.

2. Relevance to the Theme (Attention Economy on Campus)

Our solution directly targets the "Attention Economy on Campus" by turning attention into a **measurable, gamified resource**. In an attention economy, focus is precious and scarce, so our platform **steers campus incentives towards healthy focus patterns**. We align our features with the hackathon theme's goals:

- **Control Time & Reduce Clutter:** An AI-powered scheduler breaks courses and personal goals into concrete tasks, prioritizing by deadlines and user goals (reducing decision fatigue). This addresses *"take control of my time"* by auto-organizing a student's workload into manageable, weighted blocks.
- **Gamify Focus & Loss Aversion:** By staking tokens on tasks, we leverage behavioral economics. Users **stand to lose** staked tokens if they skip tasks (loss aversion), which research shows can boost learning more than fixed rewards. Gamification elements (points, badges, leaderboards) directly tackle *"improve focus and decisions"*, providing instant feedback and motivation.
- **Enhance Energy & Engagement:** Token rewards and peer recognition turn studying into a competitive, rewarding activity.
- **Build Peer Ecosystem:** The platform fosters a supportive peer network. Students validate each other's work and vote on progress, increasing accountability. Structured peer assessment is known to **improve student accountability and engagement**. This communal element aligns with the theme by harnessing social motivation and trust.
- **Holistic Focus Metrics:** We create course-wise and personal-proficiency scores (blend of upvotes, consistency, time invested). This quantification of focus addresses *"make decisions with more focus/energy"* by showing where effort goes. Students (and CRs/administration) can see transparent "attention scores" backed by backend's immutable record[12], aligning with the theme's emphasis on reallocating energy wisely.

In short, **Focus Enhancer** transforms the campus attention economy by rewarding deep, goal-driven focus and introducing accountability mechanisms grounded in proven psychology (loss aversion[6], gamification[7], peer learning[11]). This directly fulfills the theme: we fight digital distraction by making focus itself a valuable commodity on campus.

3. Proposed Solution ("Focus Enhancer")

Core Concept

Focus Enhancer is a decentralized, AI-driven platform where students **plan tasks, stake tokens, and receive peer-verified rewards** for completing meaningful work. At its heart, it turns studying and project work into an incentivized, gamified process:

- **AI-Generated Task Scheduling:** The system ingests a student's course data (syllabus, credit weight, deadlines) and personal goals (e.g. solving 100 coding problems). Using an AI engine, it automatically generates a **personalized task list** with weighted priorities. For example, "Solve LeetCode problems" , "Study PRML for 2 hours" . Each task has a recommended token stake (higher for harder/urgent tasks). This component is akin to the "AI Personal Study Buddy" concept: smart scheduling + practice generation to make studying efficient.

- **Tokenized Commitment Contracts:** Before starting a task, the student **stakes campus tokens** (earned earlier or bought) as a commitment. If they complete the task by its deadline, they get a reward (their stake plus bonus tokens). If they fail, their stake got subtracted from the total points/ tokens they possess (like gambling). This uses loss aversion: people strongly prefer avoiding loss, so staking creates a psychological push to finish work. Token economics also let us gamify progress (e.g. leveling up reputation).
- **Peer Verification & Voting:** To prevent cheating, task completion (especially for writing or problem-solving tasks) is **verified by peers**. After a student uploads a handwritten or typed solution (e.g. scanned notes or code), other students review it. They upvote correct efforts or downvote with comments. We integrate an AI assistant to generate quick quiz questions on submitted content to test understanding (ensuring the submitter didn't just copy). Well-justified downvotes are validated by the AI for bias control. Research on peer assessment shows students tend to take rating seriously and improve performance, gaining accountability and metacognitive skills.
- **Reputation & Proficiency Scoring:** Each student has a dynamic score: weighted sum of tokens earned, upvotes received, tasks completed, and consistency streaks. This proficiency metric (inspired by cognitive load theory emphasis on sustained effort) feeds a course-wise leaderboard. High scoring students become informal "course champions," gaining recognition.
- **Course Rep (CR) System:** The position of CR is volunteering. Any person who takes a specific course can be a CR for that course. The allocation will be first come first serve. The CR channel collects anonymous feedback and can invoke batches of polls (misinformation by the CR) (e.g. dismiss a CR via collective vote) – resulting in a fair dismissal . The CR will get a slight amount of tokens/points for an announcement of anything upcoming (quiz/assignment).

Key Features & Mechanisms

1. **AI Study Assistant:** Natural Language APIs (e.g. OpenAI or Gemini) analyze syllabus PDFs, past grades, and student inputs to tailor tasks and generate study quizzes. Similar systems (like the "AI Personal Study Buddy") have been shown to boost organization and comprehension.
2. **Gamified Economy:** Points and badges for milestones (weekly streaks, peer vetting) motivate short-term engagement. The token economy ensures objective rewards: e.g. completing 80% of tasks yields daily bonus tokens (mirroring "play-to-earn" game dynamics but for learning).
3. **Peer Governance:** We embed recommendation and review features. By empowering peers to upvote quality solutions, we harness the documented benefits of structured peer review (greater motivation, accountability and learning gains).
4. **Immutable Records:** All stakes, votes, and credentials are tracked on backend. This prevents tampering (e.g. a student cannot illegitimately erase penalties or false

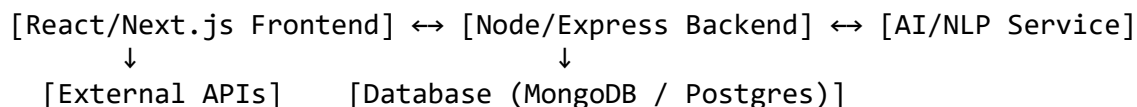upvotes). Transparent database for this project align with industry calls for trustworthy credentialing.

5. **External API Integration:** Verified achievements on coding platforms (e.g. GitHub commits, Codeforces/LeetCode contest ranks) are fetched via APIs and converted to tasks/rewards. This validates personal goals (e.g. "rank 500" on CF yields tokens), linking intrinsic motivation with our system. (this is an additional feature which we are thinking to implement , cant guarantee)

Taken together, Focus Enhancer is **an all-in-one academic productivity ecosystem**. Instead of isolated apps (calendars, timers, forums), it tightly integrates **AI planning, incentives, and social proof**. Each mechanism is backed by research: gamification ↑ motivation, loss aversion ↑ commitment, and peer assessment ↑ accountability.

4. System Architecture

The system is a multi-tier web application with High-level components:

- **Frontend (Web/Mobile):** A responsive React/Next.js interface where users log in, view tasks, stake tokens, upload solutions, and vote. It connects to the backend via REST/GraphQL APIs .
- **Backend/API Layer:** A Node.js/Express server handles business logic: user auth, task generation, stake calculations, and interaction with AI services. It also orchestrates calls to smart contracts and external APIs.
- **AI Engine:** Hosted on cloud (or via API to OpenAI/Gemini), this module parses course materials, student inputs, and generates tasks/questions. It returns structured tasks (with text, deadlines, weights) and evaluates downvote justifications.
- **Database:** A central database (MongoDB or PostgreSQL) stores user profiles, course data, tasks, submission records, votes, and off-chain token balances. It caches backend submission states (like token holdings) for performance. The schema includes collections/tables for **Users, Tasks, Submissions, Votes, Courses, TokenLedger, Reputation**.
  External Integrations: Connectors to platforms like Codeforces, LeetCode, GitHub, etc., periodically fetch user activity to auto-validate personal goals. OAuth/webhooks tie their achievements back into our task-reward engine.

```
[React/Next.js Frontend] ↔ [Node/Express Backend] ↔ [AI/NLP Service]
         ↓                              ↓
   [External APIs]       [Database (MongoDB / Postgres)]
```

Each layer is modular: - The **API layer** uses JWT auth and role-based access (Student, CR).

5. Tech Stack

We propose using widely adopted, hackathon-friendly tools to deliver a robust MVP:

- **Frontend:** *React.js* (with Next.js for SSR if needed) for fast UI development and routing. TailwindCSS or Material-UI for styling. Wallet integration (MetaMask) using **ethers.js** or **web3.js**.
- **Backend:** *Node.js* with *Express.js* to build RESTful APIs. Use *TypeScript* for type safety. Authentication with JWT. File upload (PDF scans) handled by Multer.
- **AI/NLP:** Leverage *OpenAI API* (e.g. GPT-4) or *Google Gemini* for task/question generation. Use vector embeddings (with Pinecone or FAISS) to detect cheat/hint overlap.
- **Database:** *MongoDB* (for flexible document schema) or *PostgreSQL* (for relational data), hosted on a managed service (Mongo Atlas or AWS RDS).
- **DevOps:** Containerize with Docker; deploy to AWS/GCP/Azure (Node backend and AI services). Use *Infura/Alchemy* to connect to Ethereum nodes (avoids running full node).
- **Security:** Sanitize all inputs. Encrypt PDF uploads at rest. Rate-limit API. Smart contracts audited with MythX.
- **Testing:** Jest/Mocha for unit tests

6. MVP Scope

MVP Focus: We aim to showcase the **core loop** of planning, staking, and verifying tasks. In hackathon scope, the MVP would include:

- **User Registration/Login** and course enrollment (simulated data).
- **Task Scheduler:** A few hardcoded tasks per course (e.g. "Solve 2 problems", "Read 1 lecture").
- **Token Economy:** Users start with a balance of mock tokens. They can stake and receive payouts via the smart contract.
- **Peer Review:** A simple voting interface (even without AI checks) to approve task completion. This demonstrates accountability.
- **Leaderboard:** Showing that staking and votes translate to visible scores.

By prioritizing these elements, our MVP will convincingly demonstrate how **staking+AI+peer review** transforms student focus into trackable, rewarded action.