YouMatter Gamification Module: Project Documentation

Project: Gamification Overhaul for the YouMatter Wellness Platform
Chosen Track: Track 1: Behavioural Psychology Integration

FIgma Prototype link : link

Github code Link : link

Video Demo : link

1. Technical Architecture

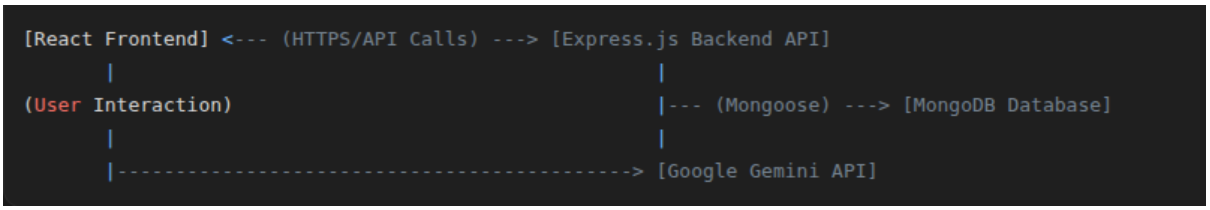
1.1. Detailed System Design

The architecture is a modern, scalable MERN stack designed for a reactive user experience and secure data handling.

- **Frontend (Client-Side):**
  - **Framework:** React (with Vite for a fast development experience).
  - **Responsibilities:** Renders the user interface, manages component-level and global state, handles user input, and communicates with the backend via REST API calls.
  - **State Management:** React Context API (AuthContext) is used for global management of user authentication, session data, and core user stats (points, streak).
  - **API Communication:** The axios library is used for all HTTP requests, with an interceptor configured to automatically attach the JWT authentication token to protected requests.
-
- **Backend (Server-Side):**
  - **Framework:** Node.js with Express.js.
  - **Responsibilities:** Provides a RESTful API for the frontend, handles all business logic (calculating streaks, awarding points, validating tasks), authenticates users via JWT, and interacts with the database.
  - **Architecture:** Modular, with distinct files for routes, models, and middleware to ensure separation of concerns and maintainability.
-
- **Database:**
  - **System:** MongoDB (a NoSQL document database).
  - **Responsibilities:** Persistently stores all user data, including profiles, task completion logs, journal entries, and activity lists.

    - **Interaction:** Mongoose is used as the Object Data Modeling (ODM) library to define schemas and interact with the database in a structured way.
-
- **Third-Party Services:**
    - **AI Chatbot:** Google Gemini API is integrated for the AI emotional companion, providing conversational AI capabilities.
-

System Flow Diagram:



1.2. Database Schemas

(This section details the structure of the data as stored in MongoDB)

users Collection (User.js):

- _id: ObjectId (Primary Key)
- username: String, required
- email: String, required, unique
- password: String, required (stored as a hash)
- points: Number, default: 0
- streak: Number, default: 0
- lastTaskCompletionDate: Date (tracks the last day any task was done, for streaks)
- dailyTasks: Object (tracks completion of specific tasks for the day)
    - Meditation: Date
    - Jogging: Date
    - Journal: Date
    - *(Dynamically adds other tasks like "Yoga")*
-
- preferences: Object (stores user's preferred tasks)
    - preferredMental: String, default: 'Meditation'
    - preferredPhysical: String, default: 'Jogging'
-
- rewards: Array (of embedded reward documents)
    - name: String
    - description: String
    - dateEarned: Date

        - redeemed: Boolean
-
- createdAt, updatedAt: Timestamps

activities Collection (Activity.js):

- _id: ObjectId
- name: String, required, unique
- type: String, required, enum: ['Mental', 'Physical']
- description: String
- points: Number, required
- duration: String

journal_entries Collection (JournalEntry.js):

- _id: ObjectId
- userId: ObjectId, ref: 'User'
- content: String, required
- upvotes: Array of ObjectId, ref: 'User'
- createdAt, updatedAt: Timestamps

tasks Collection (Task.js):

- _id: ObjectId
- userId: ObjectId, ref: 'User'
- taskType: String, required
- pointsEarned: Number, required
- createdAt, updatedAt: Timestamps

1.3. API Specifications

(This is a summary of the key API endpoints)

Authentication (/api/auth)

- POST /register: (Public) Creates a new user.
- POST /login: (Public) Authenticates a user and returns a JWT.
- GET /user: (Private) Retrieves the logged-in user's data.

User Actions (/api/users)

- GET /leaderboard: (Private) Gets the top 10 users by streak.
- PUT /preferences: (Private) Updates the user's preferred tasks.
- POST /rewards/claim: (Private) Claims a reward by spending points.

Tasks (/api/tasks)

- POST /complete: (Private) Logs a completed task, updates points and streak. Validates that the task can only be done once per day.

Journal (/api/journal)

- GET /: (Private) Gets all journal entries for the logged-in user.
- POST /: (Private) Creates a new journal entry.
- GET /community: (Private) Gets an anonymized feed of all journal entries.
- PUT /upvote/:id: (Private) Toggles an upvote on a specific journal entry.

Activities (/api/activities)

- GET /: (Private) Retrieves the list of all possible activities.
- POST /seed: (Public, Dev-Only) Populates the database with default activities.

---

2. User Journey Maps

2.1. The Daily Engagement & Habit Formation Loop

- **Goal:** Encourage daily app usage until it becomes a habit.
- **Psychology:** The Habit Loop (Cue, Routine, Reward) & Progressive Achievement.
- **Journey:**
    1. **Cue:** User receives a push notification ("Your daily tasks are ready!") or opens the app out of habit.
    2. **Routine (Interaction):**
        - User opens the Dashboard.
        - Sees their personalized Mental and Physical tasks.
        - Clicks a task button (e.g., "Yoga").
    3.
    4. **Reward (Feedback):**
        - A toast notification appears: "Great job! You earned 50 points."
        - The task card updates to show "✅ Done".
        - The "Points" and "Streaks" stats visibly update on the dashboard.
    5.
    6. **Investment:** The user sees their streak increase. The desire not to "break the chain" motivates them to return the next day. They also check the Leaderboard and feel motivated to climb higher.
-

2.2. The Social Validation & Community Loop

- **Goal:** Increase user-generated content and foster a sense of community.
- **Psychology:** Social Proof, Social Accountability, Dopamine release from social validation (likes).
- **Journey:**
    1. **Cue:** User completes the "Journal" task.
    2. **Routine (Interaction):**
        - User writes their thoughts on the Journal page.
        - The entry is posted anonymously to the Community Feed.
    3.
    4. **Reward (Feedback):**
        - Another user reads the entry and clicks the "Like" button.
        - The original author receives points for their entry getting likes (future feature).
        - The author feels validated and understood by the community.
    5.
    6. **Investment:** The user is more likely to write again to contribute and receive validation. They spend more time reading other posts, strengthening their connection to the app.
-

---

3. User Experience (UX) Documentation

3.1. Wireframes & Mockups

The provided Figma designs serve as the high-fidelity prototypes and UI mockups for this project. All components have been developed to match these specifications.

3.2. User Stories

- **As a new user,** I want to see personalized daily tasks on my dashboard, **so that** I feel the app is tailored to me and I'm motivated to start.
- **As a daily user,** I want to maintain my streak by completing tasks, **so that** I can climb the leaderboard and feel a sense of accomplishment.
- **As a competitive user,** I want to see a leaderboard, **so that** I can compare my progress with others and stay motivated.
- **As a user seeking support,** I want to write my thoughts in a journal and share them anonymously, **so that** I can express myself without fear of judgment and feel connected to a community.
- **As a goal-oriented user,** I want to earn points and redeem them for tangible rewards, **so that** my efforts feel valued.
- **As a user feeling lonely,** I want to talk to an AI companion, **so that** I can have a supportive and non-judgmental conversation anytime.

3.3. Accessibility Considerations

- **Color Contrast:** UI elements have been designed with high color contrast to ensure readability for users with visual impairments (e.g., the dark text on colored task headers).
- **Semantic HTML:** Standard HTML elements (<header>, <nav>, <button>) are used to ensure proper document structure for screen readers.
- **Keyboard Navigation:** All interactive elements (buttons, links) are keyboard-accessible.
- **Image Alt Text:** All illustrative images include descriptive alt tags.
- **Responsive Design:** The UI is built to be responsive and usable on various mobile screen sizes.

---

4. Business Case Analysis

4.1. ROI Projections & KPI Impact

- **Increase DAU by 40%:** The daily task and streak system directly targets this KPI. The habit loop and fear of breaking the chain are powerful psychological drivers for daily return visits.
- **Drive Organic Downloads by 50%:** The Leaderboard and anonymous Community Feed create "viral loops." Users may share screenshots of their high rank or insightful posts, acting as organic marketing and driving new installs.
- **Improve Feature Adoption by 60%:** By assigning points to a wide variety of features (tasks, journaling, podcast listening), we incentivize users to explore parts of the app beyond the core policy servicing. The Rewards Store further encourages this exploration as users "hunt" for points.

4.2. User Acquisition Cost (UAC) Impact

The social features directly reduce UAC by promoting word-of-mouth marketing. As the community grows, the platform becomes more valuable, creating a network effect that attracts new users organically, thus lowering the reliance on paid advertising.

4.3. Retention Modeling

The gamification module is designed to significantly boost long-term retention.

- **Short-Term Retention (Days 1-7):** Driven by the immediate satisfaction of earning points and the desire to build an initial streak.
- **Mid-Term Retention (Weeks 2-4):** Reinforced by the Leaderboard, where users start seeing meaningful rank changes, and the unlocking of initial rewards.

- **Long-Term Retention (Months 1+):** Solidified by habit formation ("I do my YouMatter tasks every morning") and social investment in the community.

---

5. Implementation Roadmap

Phase 1: Core Gamification Engine (MVP - Complete)

- **Features:** User Authentication, Dashboard with static tasks, Points & Streak System, basic Journaling.
- **Goal:** Establish the foundational habit loop.
- **Resources:** 1 Backend Dev, 1 Frontend Dev.
- **Timeline:** 1 Month.

Phase 2: Personalization & Social Integration (Complete)

- **Features:** Dynamic tasks based on user preferences, Profile Page for setting preferences, Community Feed, Leaderboard, Rewards Page.
- **Goal:** Enhance engagement through personalization and social accountability.
- **Resources:** 1 Backend Dev, 1 Frontend Dev, 1 UI/UX Designer.
- **Timeline:** 1 Month.

Phase 3: AI & Advanced Engagement (Next Steps)

- **Features:**
    - **True AI Personalization:** Implement an ML model that analyzes task completion history to automatically suggest the most effective tasks for a user, moving beyond explicit preferences.
    - **Advanced Rewards:** Integrate real-world partner rewards or blockchain-based tokens.
    - **Community Challenges:** Introduce time-based challenges where the entire community works towards a collective goal.
    - **Push Notification Engine:** Develop intelligent, personalized reminders based on user behavior patterns.
- **Goal:** Maximize long-term retention and create a deeply personalized wellness journey.
- **Resources:** 2 Backend Devs (one with ML focus), 1 Frontend Dev, 1 Data Scientist.
- **Timeline:** 3-6 Months.