

# Untitled spreadsheet - Sheet1

Aa Day	≡ Date	≡ Task	≡ Purpose	≡ Expected Output
1	June 02, 2025	Set up dev environment: Install Node.js, MongoDB, VS Code; initialize backend (Express), frontend (React), extension folders; set up Git.	Establish project foundation with tools and structure.	Project folder with backend, frontend, extension subfolders; MongoDB database; Git repo.
2	June 03, 2025	Research Chrome Extension APIs, TensorFlow.js (Face-API.js), Spotify API; create Notion/Trello task board.	Understand tech stack and organize tasks for clarity.	Notes on APIs, task board with project plan.
3	June 04, 2025	Build Chrome extension: Create manifest.json, background.js; track screen time with chrome.windows API.	Start data collection for screen time, core to ChillBoard.	Extension tracking screen time, stored in chrome.storage.local.
4	June 05, 2025	Add tab usage tracking to extension using chrome.tabs API; log URLs and time spent.	Capture browsing habits for analysis.	Extension tracks tab usage, stored in chrome.storage.local.
5	June 06, 2025	Add popup interface to extension (popup.html, popup.css); display screen time, tabs, and web app link.	Make extension user-friendly with quick stats access.	Extension popup showing stats and linking to web app.
6	June 07, 2025	Set up Express.js backend; connect to MongoDB; create /health endpoint.	Establish backend for data storage and API communication.	Running Express server, connected to MongoDB, with test endpoint.

Aa Day	≡ Date	≡ Task	≡ Purpose	≡ Expected Output
7	June 08, 2025	Define MongoDB models for User, ScreenTime, TriggerLink; test with /test-user endpoint.	Set up database structure for core data.	MongoDB models created, test user saved.
8	June 09, 2025	Implement user authentication: /signup, /login endpoints with JWT and bcrypt.	Enable secure user access and data personalization.	Backend with signup/login endpoints, returning JWT tokens.
9	June 10, 2025	Update extension to sync screen time/tab data to backend (/screen-time endpoint); add login to popup.	Connect extension to backend for data storage.	Extension syncs data to backend, saves in MongoDB.
10	June 11, 2025	Set up React frontend with react-router-dom, axios; build Landing Page with signup/login forms.	Create user-facing app and enable onboarding.	React app with routing, Landing Page for signup/login.
11	June 12, 2025	Build Dashboard Page: Fetch/display screen time, tab usage as charts; add extension reminder.	Display core metrics to users, validating data flow.	Dashboard showing screen time/tab usage charts.
12	June 13, 2025	Define MongoDB models for Mood, Recommendation, Challenge, Playlist; test with /test-mood endpoint.	Complete database for advanced features.	All MongoDB models defined, test mood saved.
13	June 14, 2025	Implement emotion detection in frontend using TensorFlow.js/Face-	Enable AI-based mood detection for recommendations.	Dashboard with webcam-based emotion detection.

Aa Day	≡ Date	≡ Task	≡ Purpose	≡ Expected Output
		API.js; add webcam toggle.		
<u>14</u>	June 15, 2025	Save mood data to backend (/mood endpoint); link to TriggerLink; add mood correction option.	Complete emotion detection with backend storage.	Mood data saved in MongoDB, linked via TriggerLink.
<u>15–20</u>	June 16–21, 2025	Build recommendation system: Create /recommendations endpoint; evaluate screen time/mood; display on Dashboard; track user acceptance.	Provide personalized suggestions based on data.	Recommendations generated, displayed, and tracked.
<u>21–27</u>	June 22–28, 2025	Integrate Spotify API: Authenticate, fetch playlists by mood, embed player, save playlists; handle rate limits with caching.	Add music-based wellness feature.	Spotify playlists fetched, played, and saved in MongoDB.
<u>28–34</u>	June 29–July 05, 2025	Build Challenges and Leaderboard: Create Challenges page, /challenges endpoints; track progress, rank users; display on Dashboard.	Encourage digital detox with gamification.	Challenges page, leaderboard, and progress tracking.
<u>35–41</u>	July 06–12, 2025	Build Profile, Settings, About/Help pages: Show trends, save settings, add guides/privacy policy, contact form.	Complete app with user customization and info.	Functional Profile, Settings, About/Help pages.
<u>42–60</u>	July 13–31, 2025	Polish and test: Style app (calming design), make responsive, add	Ensure app is user-friendly and bug-free.	Polished, responsive app with tested features.

Aa Day	≡ Date	≡ Task	≡ Purpose	≡ Expected Output
		animations; test extension, emotion detection, Spotify, challenges.		
<u>61–75</u>	August 01–15, 2025	Debug and optimize: Fix bugs, reduce API calls, re-test all features (tracking, mood, recommendations, challenges).	Improve reliability and performance.	Bug-free, optimized app with all features working.
<u>76–90</u>	August 16–31, 2025	Document and present: Write project report, record demo video, create presentation slides with ER diagram and takeaways.	Prepare for project submission and presentation.	Project report, 5–7 min demo video, presentation slides.
<u>Untitled</u>				