PowerBook

A Web Platform for Habit-Building and Competitive Reading

CSCI-409 Senior Project II — Spring 2026

Nazarbayev University, School of Engineering and Digital Sciences

Toktar Sagyngali
Database & Documentation

Nurislam Bakhtybayev
Backend Development

Bekzhan Aktoreev
Leaderboard & Performance

Bektas Keldibayev
Frontend & UI/UX

Nurdaulet Otegenov
Telegram Bot Integration

Askar Boranbayev
Project Advisor

LIVE: https://power-book.app/ | GROUP 44

THE PROBLEM: HABIT DECAY

- **Attention Attrition:** Sustained reading habits are declining among students due to the friction of long-form content vs. short-form social media.
- **Passive Platforms:** Existing catalogs like Goodreads lack a "reason to return," focusing on logging rather than active habit triggers.
- **Social Isolation:** Research shows that consistency and accountability are the primary drivers of new habits, yet reading remains a solo activity.
- **Motivation Gap:** Current tools provide cataloging but fail to generate the social pressure or excitement required to sustain long-term engagement.

MARKET GAP: WHY APPS FAIL

📚

Goodreads

Strong cataloging and reviews. Weak real-time competition. No time-based rewards or streak-driven loops.

📊

StoryGraph

Great analytics. Still focused on the individual. Lacks head-to-head competition and social incentive mechanisms.

⚡

PowerBook Gap

Integrates time-based tracking, structured social competition, real-time leaderboard, and Telegram bot engagement.

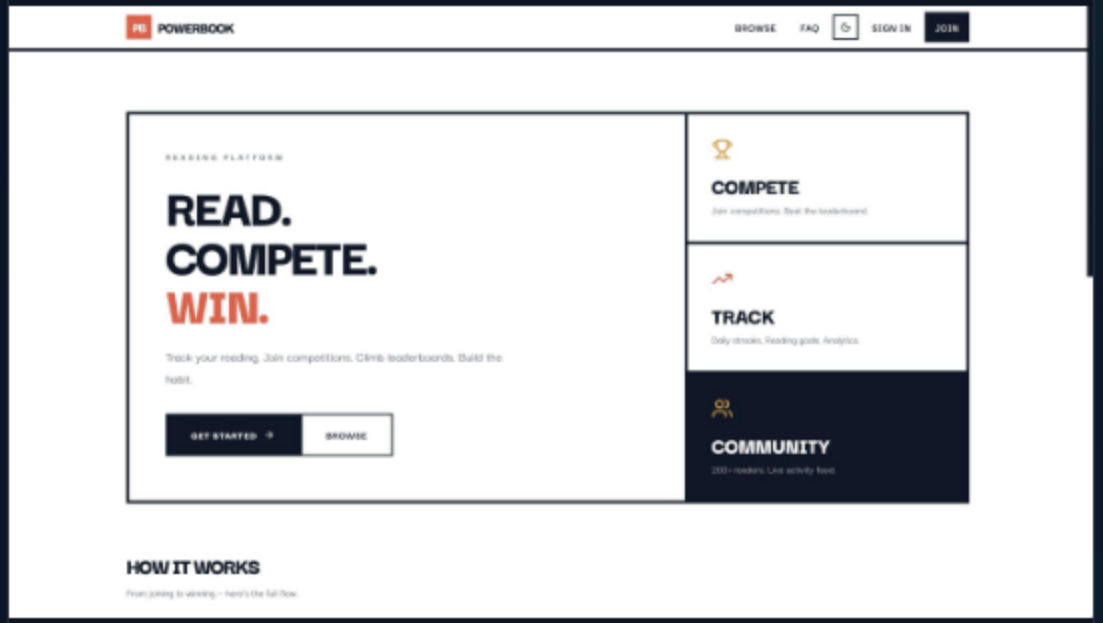
OUR SOLUTION

PowerBook: Habit-Building via Social Competition.

A web platform where reading is no longer a solo journey. Users log minutes, join time-boxed competitions, and earn status through a live hierarchy.

- ✓ Head-to-head reading challenges
- ✓ Live XP, Levels, and Badges
- ✓ Integrated Telegram habit triggers

Read. Compete. Win.



PROJECT DELIVERABLES

🗂️

Full-Stack App

Deploy a robust web application that gamifies reading activity through competitive mechanics.

🎁

Engagement Engine

Design a unique Gift-Exchange system to sustain interest after the initial novelty wears off.

✈️

Multi-Channel Bot

Integrate Telegram to maintain the habit loop even when the primary website is closed.

THE PRODUCT LOOP

1

Join

User registers and joins an active competition.

2

Log

Reading minutes submitted via Web or Bot.

3

Level Up

XP awarded, ranks updated on live leaderboard.

4

Exchange

Competition ends; Top vs. Bottom gift pairing.

SIGNATURE: GIFT EXCHANGE

Inclusive Competition

To avoid a "winner-takes-all" burnout, the platform pairs the **top-half** (Givers) with the **bottom-half** (Receivers).

- ✓ Top 1 paired with Bottom 1
- ✓ Encourages low-rank users to stay active
- ✓ Incentivizes mentorship over elitism

Lifecycle Automation

Competitions transition automatically through states: Created → Joinable → Active → Closed.

System triggers final XP awards and badge evaluation immediately upon deadline expiration.

PROGRESS & GAMIFICATION

10 Levels of Mastery

From **Newbie** (0 XP) to **Book King** (14,000 XP). Each level represents a significant milestone in reading volume and consistency.

Page Turner: 100 XP

Bookworm: 300 XP

Story Sage: 4,200 XP

Grand Scholar: 9,500 XP

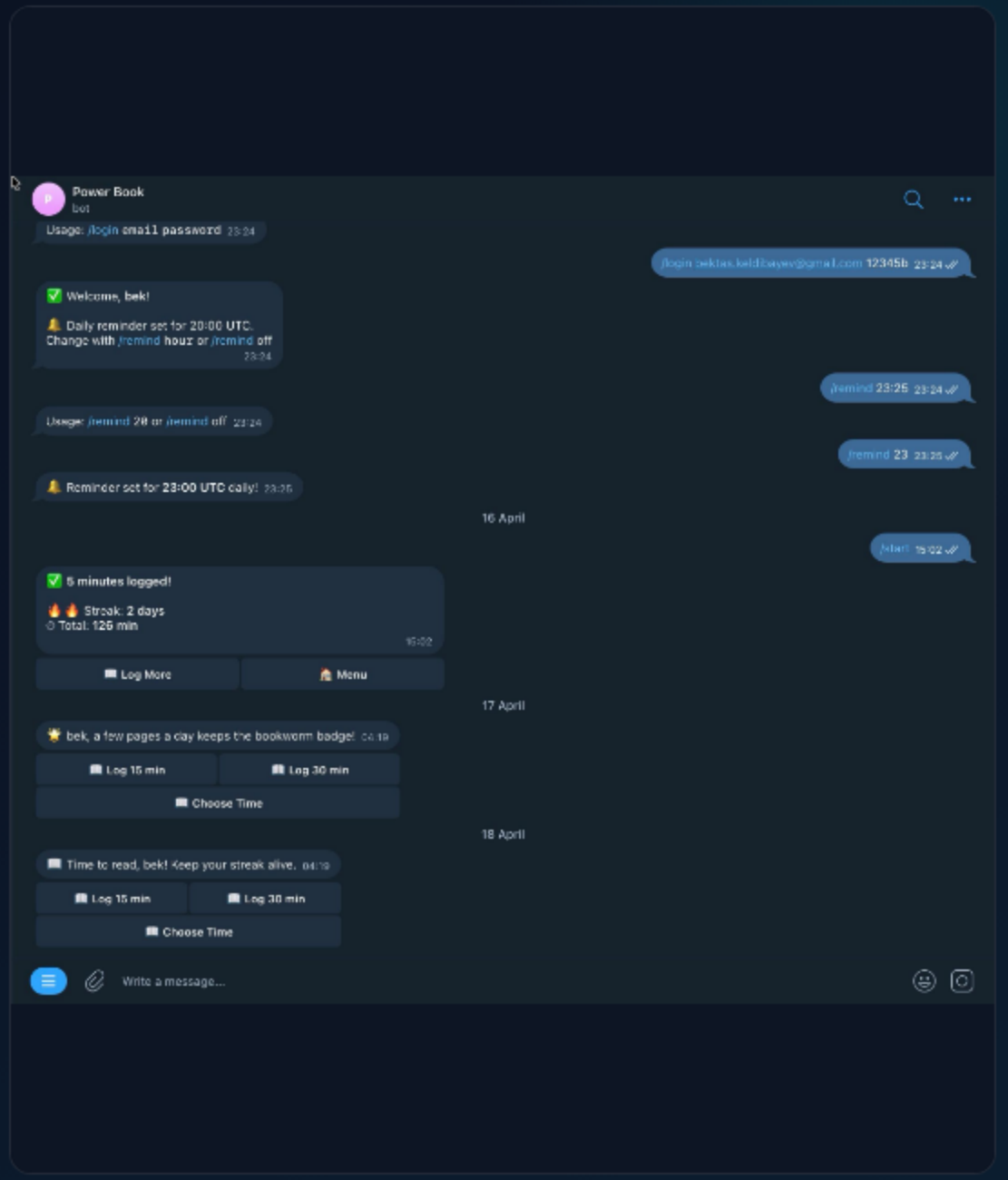
30-Badge System

- ✓ **Reading:** Total minutes, daily milestones.
- ✓ **Social:** Sending/receiving gifts.
- ✓ **Competition:** Podium finishes, streak milestones.

TELEGRAM INTEGRATION

Ubiquitous Habit Loop: The bot ensures users never have to leave their primary messaging environment to maintain their habit.

- Inline-keyboard menu (Log, Stats, Streaks)
- Daily reminders with quick-log buttons
- Automatic skip for active logs



ARCHITECTURE OVERVIEW

Frontend

Next.js 16 + React 19 + Tailwind CSS v4 deployed on Vercel for Edge performance.

≡GO

Backend API

Go + Gin Framework (Hexagonal Architecture). JWT Auth, bcrypt hashing, Dockerized.

Infrastructure

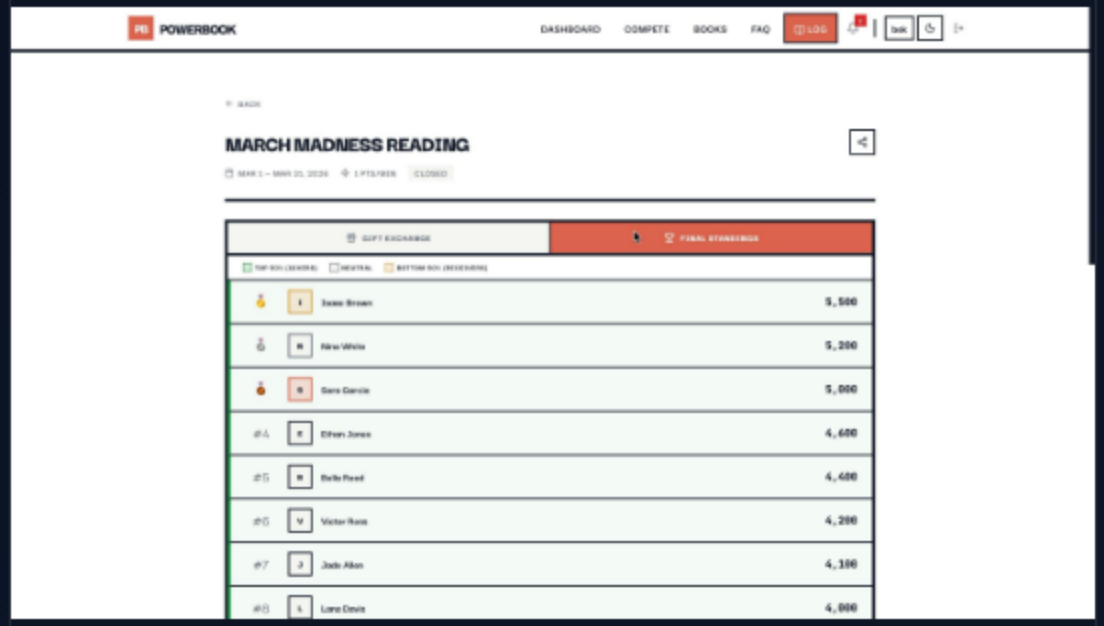
PostgreSQL (Persistence), Redis (Real-time Leaderboard), Caddy (Reverse Proxy).

BACKEND ENGINEERING

- **Hexagonal Design:** Decouples domain logic from external ports (API/DB) for robust testing and maintenance.
- **Efficiency:** Written in Go for high concurrency and minimal memory footprint (<90MB RAM in prod).
- **Reliability:** Background schedulers handle competition closing and gift pairing every 5 minutes.
- **Integrity:** Server-side daily caps (1,440 mins) and JWT stateless authentication ensure data correctness.

SCHEMA & DATA MODEL

TABLE	KEY COLUMNS	ROLE
users	id, xp, level, streak, telegram_handle	Core identity & status
reading_logs	user_id, minutes, source, created_at	Activity historical stream
competitions	creator_id, scoring_rule, status, deadline	Event orchestration
participants	user_id, competition_id, points, rank	Competition state
user_badges	user_id, badge_type, awarded_at	Gamification state



REAL-TIME LEADERBOARD

Redis Sorted Sets

Implemented to replace PostgreSQL aggregates as the user base scaled. Atomic score updates via **ZINCRBY**.

- ✓ "Find Me" navigation for rapid rank check
- ✓ Motivational banners based on half-split
- ✓ Rank reads < 20ms under load

USER EXPERIENCE (UX)

🎨

Brutalist Style

Bold typography and high-contrast dark/light theme for focus and hierarchy.

⚡

React Query

Smart caching and skeleton loading states to eliminate generic "loading..." text.

👤

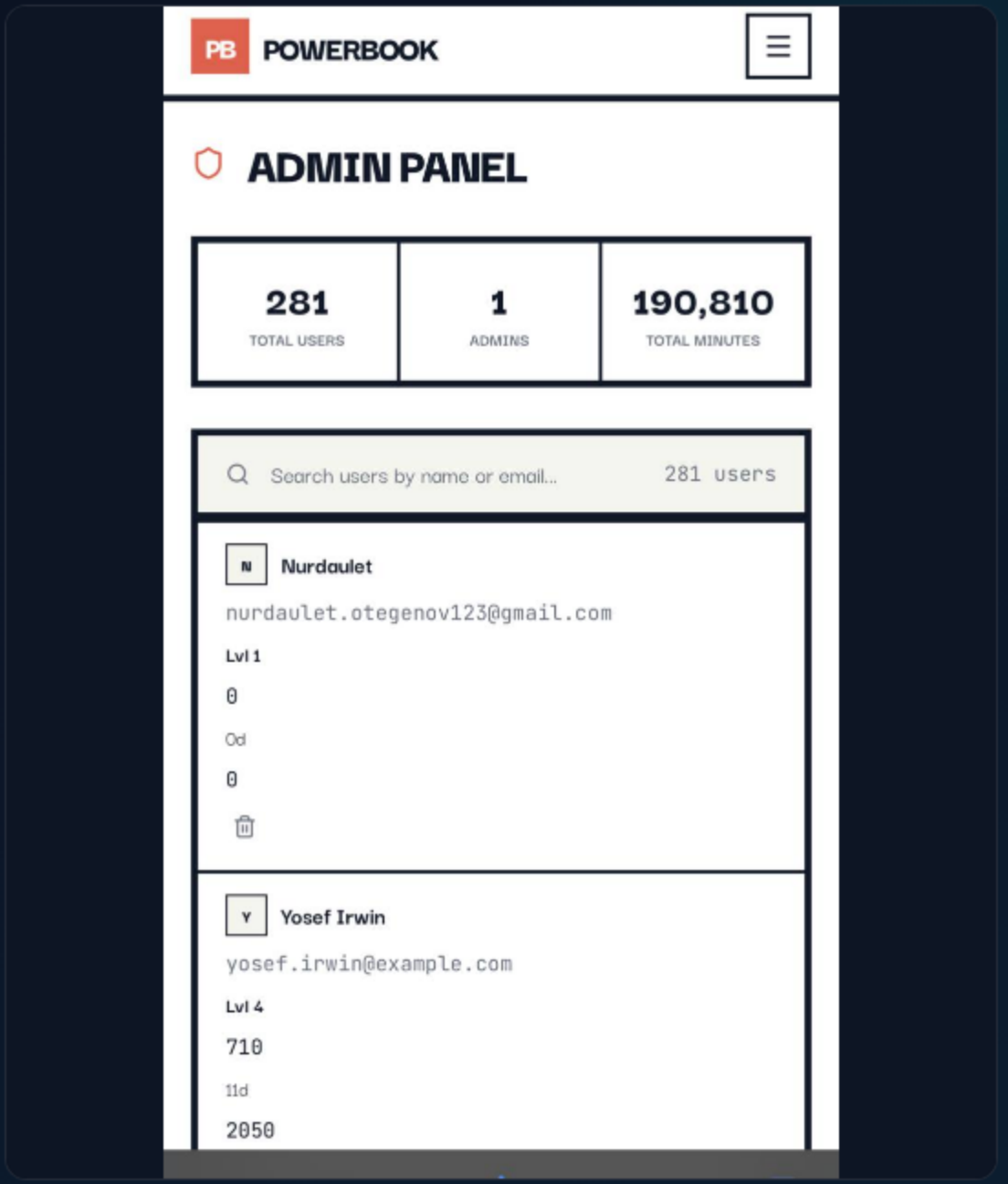
Dynamic Avatars

Unique color hashes generated from display names for instant identity recognition.

PLATFORM GOVERNANCE

Protected Admin Panel (`/admin`) allows platform management beyond the core user flow.

- ✓ Global activity stats (total mins, users)
- ✓ User search and deletion actions
- ✓ Enforced RBAC authorization



PROJECT EXECUTION TIMELINE

Semester 1: Fall 2025

- Gamification & habit formation research
- Architecture definition & API design
- PostgreSQL schema & Docker environment
- Auth, logs, & competition lifecycle MVP
- Initial frontend integration

Semester 2: Spring 2026

- Redis leaderboard & "Find Me" functionality
- Gift exchange & XP/Leveling system
- 30-badge system & Telegram Bot redesign
- Admin panel & public user profiles
- Skeleton loading & production deployment

CHALLENGES & ITERATIONS

Leaderboard Lag

Original SQL aggregates slowed down significantly with demo data. Iterated to Redis for atomic O(log N) updates.

Bot Usability

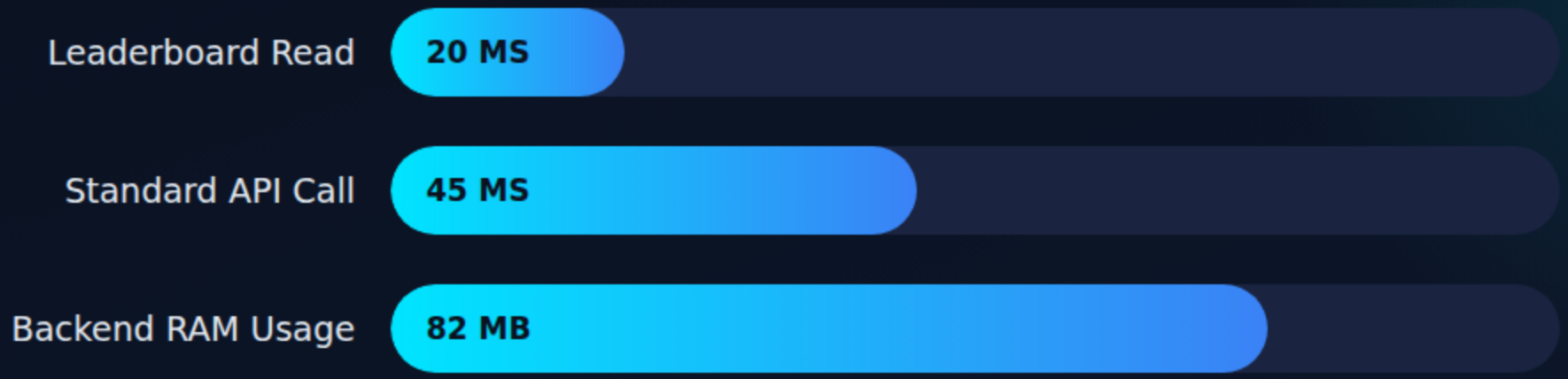
Slash commands were hard to discover. Iterated to an inline-button menu system for intuitive interaction.

"Integrity was mitigated using server-side timestamps and a 1,440-minute daily entry cap."

EVALUATION APPROACH

- **Functional:** Seeded 50 users and carried 3 competitions through full gift exchange flows.
- **Performance:** Checked production VPS response times and memory usage under load.
- **UX Analysis:** Internal testing identified friction in leaderboard navigation, leading to "Find Me" and status filters.
- **Integrity:** Tested gift logic for odd-numbered participants (Neutralization logic).

PERFORMANCE RESULTS



ALL CORE FEATURES FUNCTIONAL END-TO-END

PRODUCT DEMONSTRATION

Web Walkthrough

- Sign-in & Dashboard overview
- Competitions & Live Leaderboard
- User Profiles & Badge Showcase

System Logic

- Telegram Bot logging & reminders
- Closed competition Gift Exchange
- Admin Panel platform statistics

FINAL DELIVERABLES

1	3	10	30
LIVE APP	SERVICES	LEVELS	BADGES

A technically serious platform deployed at **power-book.app** with CI/CD automation.

FUTURE SCALING

Social Expansion

Friend networks, team competitions, and institutional rollouts.

Technical Maturity

Native mobile app, OAuth 2.0 sign-in, and AI recommendation engine.

Research

Long-term real-user study on behavioral change and habit sustainability.

Conclusion

PowerBook is a deployed, technically complete platform for competitive reading. It demonstrates that computing-based social accountability can solve the "Habit Fragment" problem.

Read. Compete. Win.

Thank You

power-book.app

Group 44 | Nazarbayev University