Team Working Agreement

Product Name: Groovo
Team Name: Groovo

Purpose of This Working Agreement

This document defines the norms, expectations, and collaborative guidelines for the Groovo development team. Its purpose is to ensure:

1. Clear communication
2. Efficient collaboration
3. Accountability
4. Healthy team dynamics
5. Consistency in code quality and workflow

The agreement is reviewed at the start of each sprint and may be updated by team consensus.

Logistics

1. Meeting Times

Day	**Time**	**Type / Attendees**
Mon	After class (9:00 AM)	Daily Standup (In-person)
Wed	After class (9:00 AM)	Daily Standup (In-person)
Fri	After class (9:00 AM)	Daily Standup (In-person)
Tue	~8:00 PM	Daily Standup (Discord)
Thu	~8:00 PM	Daily Standup (Discord)
Mon (Weekly)	After class	Sprint Planning (every 2 weeks)
Mon (Weekly)	2:00–2:45 PM	TA Meeting + Sprint Review/Demo
Mon (End of Sprint)	Immediately after Sprint Review	Sprint Retrospective (Team Only)

Meeting Rules

1. Meetings start and end on time
2. Cameras encouraged for Discord meetings
3. Stay focused; avoid unrelated phone or laptop use
4. One speaker at a time
5. Decisions recorded in weekly notes

2. Project Repository

Location: GitHub – https://github.com/shawndhillon/groovo

Branching model:

a. MAIN – stable, production-ready code
b. DEV – integration branch for completed features
c. Feature Branches – one per user story

Repository Organization Structure:

```
├── app/                # Next.js application source
│   ├── (routes)/       # Page route groups
│   ├── api/            # API route handlers
│   ├── components/     # Shared React components
│   ├── data/           # Billboard and static data
│   ├── hooks/          # Custom React hooks (useLibrary, useUserReviews, etc.)
│   ├── types/          # Shared TypeScript interfaces
│   ├── utils/          # Helper functions and utilities
│   ├── globals.css     # Global styling
│   ├── layout.tsx      # Root layout for the app
│   └── page.tsx        # Main homepage route
│
├── docs/               # Architecture, design docs, diagrams
├── scrum_docs/         # Sprint plans, retrospectives, Scrum artifacts
├── release_summary/    # Final Release Summary deliverables
├── testing/            # Test plans, reports, and utilities
├── scripts/            # Billboard scraper, utility scripts
│
├── public/             # Static assets (images, icons, misc.)
│
├── .env.example        # Environment variable template
├── README.md           # Project overview and setup instructions
└── package.json        # Project metadata and dependencies
```

3. Communication Channels

Primary communication: Discord

Urgent communication: Phone

Response expectation:

a. Weekdays: within 8 hours

b. Weekends: within 24 hours

Development Environment

2. **Platform**
    a. Next.js frontend
    b. Node.js backend
    c. MongoDB database
    d. Local dev tools: Vercel local dev / VSCode

3. **Tools**
    a. VSCode (team standard)
    b. Figma (UI design)
    c. Vercel (deployment)
    d. MongoDB Atlas (database hosting)

Work (Process) Patterns

1. **Team Collaboration**
    a. Everyone communicates blockers early
    b. Ask for help before getting stuck for >30 minutes
    c. Be respectful and constructive
    d. Assume positive intent
    e. Decisions are made collaboratively

2. **Areas of Responsibility**
    a. **Frontend:** Adam, Chris, and Srikar
    b. **Backend:** Shawn and Carter
    c. **UI/UX / Design:** Adam
    d. **Scrum Master:** Changes every Sprint

3. **Work Handoff & Integration**
    a. Feature branches must be up-to-date before PRs
    b. All PRs require at least 1 peer reviewer
    c. No pushing directly to **MAIN**
    d. Integration testing occurs before merging to **MAIN**

Product Design Patterns

1. **UI/UX Look and Feel**
    a. Dark theme as base standard
    b. Consistent spacing, typography, style, and component structure
    c. Responsive for desktop/laptop screen sizes

   d. All components follow shared UI patterns

2. **Product Architecture**
   a. Frontend uses modular React components
   b. Backend follows REST API conventions
   c. MongoDB schemas standardized in: /lib/models
   d. Avoid duplication — use shared utilities

3. **Error Handling**
   a. User-facing errors should be friendly and actionable
   b. Never fail silently
   c. Logged-out actions must show a "Please sign in" prompt

4. **Common Approach to Common Problems**
   a. For UI state issues → use React hooks consistently
   b. For shared data → use backend as single source of truth
   c. For async issues → use loading skeletons and error states

Working Agreement Review Process

1. **Reviewed** at the beginning of each sprint
2. **Amendments** require team consensus
3. Team members **commit** to following the agreement