

4DEV — Functional Specification Document

General Vision:

4DEV is a gamified, all-in-one development sandbox for building, visualizing, and collaborating on cutting-edge projects in Web3, AI (LLMs, agents), blockchain, crypto, and SaaS. The platform is intuitive, modern, and designed to enable creation without any coding knowledge. It features real-time collaboration, project management, code execution, live previews, and a modular backend architecture.

1. Landing Page (Homepage)

Objective:

Introduce 4DEV and its core value proposition while directing users to explore, create, or sign in.

Functionalities:

- Hero section with a headline, subheadline, and primary CTAs ("Explore Projects", "Create New Project", "Sign In")
- Feature highlights section
- Carousel for featured or trending projects
- Community highlights and testimonials
- Footer with navigation links and legal mentions

Interactions:

- Clickable CTAs
- Responsive design
- Hover animations and section transitions

2. Authentication (Login / Sign Up)

Objective:

Allow users to securely register and access their accounts.

Functionalities:

- Email/password login and registration
- Social login (Google, GitHub, Discord) [optional]
- Forgot password flow
- Onboarding preferences after sign-up (interests: AI, Web3, etc.)

Interactions:

- Real-time validation
- Redirect to dashboard after login
- Save onboarding data to user profile

3. User Dashboard (Main Workspace)

Objective:

Serve as the user's central workspace to manage, access, and create projects.

Functionalities:

- Sidebar with navigation (Projects, Templates, Explore, Tutorials, Gamification, Settings)
- "Create New Project" button
- Project grid or list view
- Activity timeline (recent edits, logins, comments)
- Gamification summary: level, XP, badges

Interactions:

- Open project on click
- Right-click menu: rename, duplicate, delete
- Drag-and-drop reordering
- Gamified UI feedback for actions

4. Project Editor (Core Sandbox)

Objective:

Provide a powerful, real-time collaborative development environment.

Functionalities:

- File explorer with folder structure
- Code editor with syntax highlighting (support for multiple languages)
- Live preview window
- Output console/logs
- Real-time collaboration with multiple cursors
- In-editor comments and suggestions
- Version history and checkpoints
- Run / Build / Deploy controls
- AI assistant [optional future module]

Interactions:

- Live updates when editing
- Fork or clone project
- Simulated terminal for commands
- Inline code commenting

5. Collaborative Chat (in-project)

Objective:

Enable real-time communication and discussion inside each project.

Functionalities:

- Project-specific chat window
- @mentions for collaborators
- Comment threads (with code line linking)
- File sharing or link insertion
- Typing indicators and read receipts

Interactions:

- Message history is persistent
- Notifications in sidebar or via email (optional)
- Emoji reactions to messages [optional]

6. Templates Page

Objective:

Allow users to explore and use pre-built templates to start projects quickly.

Functionalities:

- Search and filter by category (Web3, AI, etc.)
- Template cards with preview, description, and tags
- "Use Template" button to copy to user workspace
- Rating and popularity metrics

Interactions:

- Hover for quick preview
- Click to see full template page
- Clone into dashboard

7. Explore Projects (Public Gallery)

Objective:

Showcase public projects from the community to inspire and encourage collaboration.

Functionalities:

- Sortable and filterable project list
- Search bar
- Cards with owner, likes, forks, tags
- Project page with details, comments, and fork button
- Star and report functions

Interactions:

- Fork a public project into your workspace
- Comment or rate projects
- Follow project creators

8. Tutorials / Learning Center

Objective:

Provide users with educational resources and interactive tutorials.

Functionalities:

- Step-by-step tutorials with code blocks
- Video content and embedded sandboxes
- Progress tracking
- Certificates or XP rewards for completion

Interactions:

- "Mark as completed"
- Interactive code testing
- Integration with gamification system

9. Gamification System

Objective:

Encourage engagement through a reward-based system tied to platform usage.

Functionalities:

- XP points system for key actions (project creation, forking, completing tutorials, etc.)
- User levels based on XP thresholds
- Unlockable badges and achievements
- Leaderboards (global, weekly, friends-only)
- Profile page showcasing accomplishments

Interactions:

- Progress bars
- Level-up animations
- Shareable badges on social media [optional]

10. User Settings

Objective:

Provide full user account control.

Functionalities:

- Profile: name, username, bio, avatar, social links
- Preferences: dark mode, language, notification settings
- Billing info and plan management
- Account deletion and data export

Interactions:

- Real-time form validation
- Toggle preferences
- Secure handling of billing data

11. Admin Panel (Internal Use)

Objective:

Manage platform users, content moderation, and analytics.

Functionalities:

- User management dashboard
- Reports for flagged content
- Metrics (active users, project counts, etc.)

- Role assignment (admin, beta-tester, etc.)
- Template approval workflow

Interactions:

- Filter and sort by role, activity
- Assign/revoke roles
- Review flagged projects