PRD_EN_VER

AI-Powered Issue Tracking Web Application (Jira Lite MVP)

---

1. Overview

This challenge aims to design, implement, and deploy an **AI-powered lightweight issue tracking web application (MVP)** within 8 hours.

Participants will go through the entire process of requirement interpretation → design → implementation → deployment, comprehensively evaluating **AI-Native development capabilities, problem-solving skills, and web service development fundamentals**.

---

2. Core Objectives

- Evaluate the ability to **interpret incomplete requirements and organize them into a productizable structure**
- Assess web service fundamentals including **email-based authentication, Google OAuth, team management, project/issue management**
- Evaluate **AI-Native product development capabilities** such as Drag & Drop UI, AI summary/suggestion/auto-classification features
- Assess practical-level implementation abilities for **dashboards, notifications, and activity logs**
- Verify the ability to **implement a deployable service within limited time**

---

3. Detailed Functional Requirements (SRS Level)

This document defines functional requirements in **FR (Functional Requirement)** format.

3.1 Authentication

FR-001: Sign Up

Users must be able to sign up for the system using email/password.

Input

Field	Conditions
`email`	Unique, email format, max 255 characters
`password`	Min 6 characters, max 100 characters
`name`	1-50 characters

Processing

1. Check email duplication
2. Create User
3. Auto-login after creation or redirect to login screen (flexible)

Exceptions

- Email duplicate → Display error
- Format error → Display error

FR-002: Login/Logout

Users must be able to log in with email/password.

Input

- `email`
- `password`

Processing

1. Look up user by email
2. Verify password
3. On success, issue session/token (token expiration: **24 hours**)
4. Enable access to authenticated pages

Exceptions

- Mismatch → “Email or password is incorrect”

Notes

- Concurrent login (multiple devices) is flexible to implement

FR-003: Password Recovery/Reset

Users must be able to reset their password via email.

Processing

1. User enters email
2. Generate password reset link/token (expiration: **1 hour**)
3. **Actual email sending required**
4. User enters new password and completes change

FR-004: Google OAuth Login

Implement SNS login using Google account.

Processing

1. Implement Google OAuth authentication flow
2. New users are automatically registered
3. Existing users are logged in

Notes

- When implementing Google OAuth, treat it as a **separate authentication method** from email/password-based login
- Account merging for same email between regular signup and Google login is not considered

FR-005: Profile Management

Users must be able to view and edit their profile information.

Editable Fields

Field	Conditions
`name`	1-50 characters
`profileImage`	Image URL or file upload

Processing

1. Choose between URL input or file upload for profile image
2. Changes reflected immediately

FR-006: Password Change

Logged-in users must be able to change their password.

Input

Field	Conditions
`currentPassword`	Current password
`newPassword`	Min 6 characters, max 100 characters
`confirmPassword`	Must match newPassword

Processing

1. Verify current password
2. Change to new password
3. Display completion message

Exceptions

- Current password mismatch → Display error
- New password confirmation mismatch → Display error

Notes

- This feature is disabled for users who signed up only via Google OAuth

---

FR-007: Account Deletion

Users must be able to delete their account.

Processing

1. Re-confirm password (OAuth users only need confirmation button)
2. Check for owned teams
3. If owned teams exist, deletion not allowed → Guide to delete team or transfer ownership
4. Soft Delete account and related data

Exceptions

- If owned teams exist → "Please delete owned teams or transfer ownership first"

3.2 Team Features

FR-010: Create Team

Logged-in users must be able to create a new team.

Input

Field	Conditions
`name`	1-50 characters

Processing

1. Create Team (`ownerId = current user`)
2. Create TeamMember (role = OWNER)

Notes

- One user can **belong to multiple teams**
- Team roles are **OWNER / ADMIN / MEMBER**

FR-011: Update Team

Permission: Team OWNER, ADMIN

Editable Fields

- Team name

FR-012: Delete Team

Permission: Team OWNER only

Processing

- When team is deleted, all sub-projects, issues, comments, etc. are **Soft Deleted**
- Deleted teams can be restored within 30 days (restore feature is optional)

Notes

- OWNER cannot leave the team, only delete

FR-013: Invite Member

Permission: Team OWNER, ADMIN

Processing Methods (choose freely)

1. Enter invite email → Save invite info → Auto-join team when that email signs up
2. Enter invite email → Invited user approves from "Invite List" screen → Join team

Invite Rules

Item	Details
Invite expiration	**7 days**
Decline feature	None (remains pending if not accepted)
Resend	Possible (updates expiration date of existing pending invite)

Email Sending

- **Actual email sending required**

FR-014: View Members

Must be able to view the list of users in a team.

Display Information

- Member name, email, role (OWNER/ADMIN/MEMBER), join date

FR-015: Kick Member

Permission:
- OWNER: Can kick any member
- ADMIN: Can only kick MEMBERs

Processing

- Remove the member from the team
- OWNER/ADMIN cannot kick themselves

FR-016: Leave Team

Permission: ADMIN, MEMBER only (OWNER cannot leave, only delete)

Processing

- Remove the user from the team

FR-017: Team Role System (OWNER/ADMIN/MEMBER)

Team roles are divided into 3 levels.

Role Definitions

Role	Description
OWNER	Team creator, highest authority, 1 per team
ADMIN	Admin privileges, can manage members
MEMBER	Regular member, can work on projects/issues

FR-018: Change Role

Permission: OWNER only

Processing

1. Promote MEMBER → ADMIN

2. Demote ADMIN → MEMBER
3. Transfer OWNER (when transferring OWNER to another ADMIN, original owner becomes ADMIN)

Restrictions

- Must maintain at least 1 OWNER

FR-019: Team Activity Log

Must be able to view major team activities in chronological order.

Recorded Events

- Member join/leave/kick
- Role changes
- Project create/delete/archive
- Team info updates

Display Information

- Activity content, performer, target, timestamp

Pagination

- Infinite scroll or pagination required

3.3 Project

FR-020: Create Project

Users must be able to create projects within teams they belong to.

Input

Field	Required	Conditions
`name`	Required	1-100 characters
`description`	Optional	Max 2000 characters

Processing

1. Create Project (`teamId` = current team, `ownerId` = creator)
2. Created project is viewable by all team members

Restrictions

- **Max 15 projects** per team

Notes

- Project owner = creator

FR-021: View Projects

Users must be able to view all projects of teams they belong to.

Display Information

- Project name, description (summary), issue count, creation date, favorite status

Sorting

- Favorite projects displayed first
- Then by creation date descending

FR-022: Project Detail Page

Display project info + issue list/kanban board for that project.

Display Information

- Project name, description
- Issue statistics (count by status)
- Kanban board or issue list (tab switching)

FR-023: Update Project

Permission: Team OWNER, ADMIN, or project owner

Editable Fields

- Project name

- Project description

FR-024: Delete Project

Permission: Team OWNER, ADMIN, or project owner

Processing

- When project is deleted, all sub-issues, comments, etc. are **Soft Deleted**

FR-025: Project Description

Projects have a description field.

Conditions

- Max 2000 characters
- Markdown support recommended

FR-026: Archive Project

Permission: Team OWNER, ADMIN, or project owner

Processing

1. Change project to archived status
2. Archived projects are displayed separately or hidden from list
3. Issues in archived projects are read-only

Restore

- Archived projects can be restored anytime

FR-027: Favorite Project

Users must be able to favorite projects.

Processing

1. Toggle favorite (add/remove)
2. Favorited projects displayed at top of list

Notes

- Favorites are managed per user

3.4 Issue

FR-030: Create Issue

Must be able to create issues within a project.

Input

Field	Required	Conditions
`title`	Required	1-200 characters
`description`	Optional	Max 5000 characters
`assigneeUserId`	Optional	Must be a team member
`dueDate`	Optional	Date format
`priority`	Optional	HIGH/MEDIUM/LOW (default: MEDIUM)
`labels`	Optional	Multiple selection from project labels

Processing

- Created with `status = Backlog`
- Issue owner = creator

Restrictions

- **Max 200 issues** per project

FR-031: Issue Detail View

The following information must be displayed:

- Title
- Description
- Status
- Priority

- Assignee
- Due date
- Labels
- Creation date
- Subtask list
- Comment list
- AI Summary (button)
- AI Suggestion (button)
- Change history (button or tab)

FR-032: Update Issue

Permission: All team members

Editable Fields

- Title, description, assignee, due date, status, priority, labels

FR-033: Update Status

Status can be changed via Drag & Drop or detail screen.

Default Statuses

- `Backlog`
- `In Progress`
- `Done`

Custom Statuses

- Additional statuses can be defined per project (see FR-053)

Status Transition Rules

- **Direct movement allowed** between all statuses (no sequential restrictions)

Error handling for invalid status value change attempts.

FR-034: Assign User

`assigneeUserId` must be **one of the team members** of that project.

FR-035: Delete Issue

Permission: Issue owner (creator), project owner, team OWNER, team ADMIN

Processing

- Soft Delete

FR-036: Issue Search/Filtering

Search

- Title text search

Filters

- By status (Backlog / In Progress / Done / Custom)
- By assignee
- By priority (HIGH / MEDIUM / LOW)
- By label
- Has due date
- Due date range

Sorting

- Creation date, due date, priority, last modified date

FR-037: Issue Priority

Issues have priority levels.

Priority Levels

Level	Description
HIGH	Urgent, needs immediate attention
MEDIUM	Normal priority (default)

Level	Description
LOW	Low priority

Display

- Visual distinction on kanban board and issue list (colors, icons, etc.)

FR-038: Issue Labels/Tags

Must be able to create custom labels per project and apply them to issues.

Label Management

Field	Conditions
`name`	1-30 characters
`color`	HEX color code

Processing

1. Create/edit/delete labels within project
2. Multiple labels can be applied to an issue

Restrictions

- **Max 20 labels** per project
- **Max 5 labels** per issue

FR-039: Issue Change History

Must be able to view the change history of an issue.

Recorded Events

- Status change
- Assignee change
- Priority change
- Title change
- Due date change

Display Information

- Changed field, previous value, new value, changer, change timestamp

FR-039-2: Subtasks

Must be able to add checklist-style subtasks under an issue.

Input

Field	Conditions
`title`	1-200 characters

Processing

1. Add subtasks from issue detail screen
2. Show complete/incomplete with checkbox
3. Drag to reorder

Restrictions

- **Max 20 subtasks** per issue

Display

- Show subtask progress on issue card (e.g., 3/5)

3.5 AI Features

FR-040: AI Summary Generation

Must generate a 2-4 sentence summary by sending issue description to LLM API.

Behavior

- Generated **on button click** (not automatic)

Caching

- Once generated, AI results are **saved (cached)**
- If description is edited, **cache is invalidated** (needs regeneration)

Restrictions

- AI feature cannot run if description is **10 characters or less** (disable button or show message)

Error Handling

- Display error message on AI API call failure

FR-041: AI Solution Suggestion

Send a request like “Suggest an approach to solve this issue” to LLM and display results.

Behavior

- Generated **on button click**

Caching

- Same caching and invalidation as FR-040

Restrictions

- Cannot run if description is **10 characters or less**

Error Handling

- Display error message on AI API call failure

FR-042: AI Rate Limiting

Must implement rate limiting for AI API calls.

Policy

- 10 requests per minute per user
- Or 100 requests per day
- Must implement at least one

Processing

- On limit exceeded, display error message and remaining time/count

FR-043: AI Auto-Label

Must auto-recommend appropriate labels based on title and description when creating an issue.

Behavior

1. Show “AI Label Recommendation” button when creating issue
2. On click, LLM recommends suitable labels from existing project labels
3. User can accept/reject recommended labels

Input

- Issue title, description
- List of labels in project

Output

- Recommended labels (max 3)

FR-044: AI Duplicate Detection

Must detect and warn about similar issues when creating a new issue.

Behavior

1. When title input is complete (or on button click) in issue creation form
2. Compare similarity with existing issues
3. Display warning if similar issues exist

Output

- List of similar issues (max 3)
- Links to navigate to each issue

Processing

- User can ignore warning and proceed with creation

FR-045: AI Comment Summary

Must summarize discussion content for issues with many comments.

Behavior

- Generated **on button click**
- Button only enabled when there are **5 or more comments**

Caching

- **Cache invalidated** when new comment is added

Output

- Discussion summary (3-5 sentences)
- Key decisions (if any)

3.6 Kanban Board

FR-050: Kanban Board Display

Display issues in columns divided by status.

Default Columns

- Backlog / In Progress / Done

Custom Columns

- Display columns for user-defined statuses if they exist

Issue Card Display Information

- Title
- Assignee
- Priority (visual indicator)
- Labels
- Due date
- Subtask progress
- Creation date

FR-051: Drag & Drop Movement

Must be able to change status by dragging issue cards to other columns.

FR-052: Reorder Within Same Column

Must be able to reorder issues up/down by dragging within the same status (column).

Processing

- Save order information (position/order field)
- New issues added to bottom of column

FR-053: Custom Columns (Custom Status)

Must be able to define additional statuses beyond defaults per project.

Input

Field	Conditions
`name`	1-30 characters
`color`	HEX color code (optional)
`position`	Column order

Restrictions

- Default 3 + **max 5** custom statuses (total 8)

Processing

1. Create/edit/delete custom statuses
2. When status is deleted, issues with that status move to Backlog

FR-054: WIP Limit (Work In Progress Limit)

Must be able to limit max issue count per column.

Input

Field	Conditions
`wipLimit`	1-50 or unlimited (null)

Processing

1. Set/unset WIP Limit

2. Display warning when exceeded (allow movement but show visual warning)

Display

- Show current issue count / WIP Limit in column header (e.g., 5/10)
- Highlight column header when exceeded (color change, etc.)

3.7 Comments

FR-060: Create Comment

Must be able to write comments on issue detail screen.

Input

Field	Conditions
`content`	1-1000 characters

FR-061: Comment List

Comments are displayed **in chronological order**.

Pagination

- Infinite scroll or pagination required

FR-062: Update Comment

Permission: Comment author only

FR-063: Delete Comment

Permission: Comment author, issue owner, project owner, team OWNER, team ADMIN

3.8 Dashboard/Statistics

FR-080: Project Dashboard

Must provide a dashboard to view project status at a glance.

Display Information

- Issue count by status (pie chart or bar chart)
- Completion rate (Done / Total)
- Issue count by priority
- Recently created issues (max 5)
- Issues due soon (within 7 days, max 5)

FR-081: Personal Dashboard

Must provide a dashboard showing the user’s personal work status.

Display Information

- My assigned issues list (categorized by status)
- Total count of my assigned issues
- Issues due soon (within 7 days)
- Issues due today
- My recent comments (max 5)
- My teams/projects list

FR-082: Team Statistics

Must provide statistics on overall team activity.

Display Information

- Issue creation trend by period (line graph)
- Issue completion trend by period (line graph)
- Assigned issues per member
- Completed issues per member
- Issue status per project

Period Selection

- Last 7 days / 30 days / 90 days

3.9 Notifications

FR-090: In-App Notification

Must provide notifications for important events to users.

Notification Triggers

Event	Notification Target
Issue assignee assigned	Assignee
Comment written on issue	Issue owner, assignee
Due date approaching (1 day before)	Assignee
Due date today	Assignee
Team invite	Invite target
Member role change	That member

Display

- Notification icon + unread count in header
- Notification list dropdown or page

FR-091: Mark as Read

Features

1. Mark individual notification as read (on click)
2. Mark all as read button

Display

- Visual distinction between read/unread

3.10 Permissions/Security

FR-070: Team Membership Verification

Team membership verification required on all API endpoints.

Access Control

- Return **404 Not Found** when attempting to access other team's project/issue
- Return **403 Forbidden** when attempting unauthorized action

FR-071: Soft Delete Implementation

Must apply Soft Delete to all major entities.

Applicable Entities

- User
- Team
- Project
- Issue
- Comment

Implementation

- Add `deletedAt` field
- Record `deletedAt` timestamp instead of physical deletion
- Return only items with null `deletedAt` on queries

4. Data Limits Summary

Item	Limit
Projects per team	Max 15
Issues per project	Max 200
Subtasks per issue	Max 20
Labels per project	Max 20
Labels per issue	Max 5
Custom statuses per project	Max 5 (default 3 + custom 5 = total 8)
WIP Limit per column	1-50 or unlimited

Item	Limit
Team name	1-50 characters
Project name	1-100 characters
Project description	Max 2000 characters
Issue title	1-200 characters
Issue description	Max 5000 characters
Subtask title	1-200 characters
Label name	1-30 characters
Custom status name	1-30 characters
Comment content	1-1000 characters
User name	1-50 characters
Email	Max 255 characters
Password	6-100 characters
Token expiration	24 hours
Password reset link expiration	1 hour
Team invite expiration	7 days
AI feature minimum description length	Over 10 characters
AI comment summary minimum comments	5 or more
AI Rate Limit (per minute)	10 requests
AI Rate Limit (per day)	100 requests

5. Technical Requirements

- FE/BE structure flexible (React, Vue, Next.js, Svelte, Express, FastAPI, etc.)
- DB flexible (SQLite, Supabase, PlanetScale, file-based, etc.)
- AI model API provider flexible (Claude, GPT, Gemini, etc.)
- **Google OAuth implementation** (Google Cloud Console setup required)
- **Actual email sending** (SendGrid, AWS SES, Nodemailer, etc.)
- Sufficient documentation for hand-off

6. Deployment Requirements

- Must deploy to an actually accessible URL
    - “Test account” or “environment where actual signup is possible” required
- Vercel, Netlify, Render, Fly.io, etc. flexible

7. UI/UX Requirements

Required

- Loading state UI
- Error state UI
- Pagination (issue list, comment list, activity log) - infinite scroll or pagination
- **Mobile responsive**
- Dashboard charts/visualization

8. Real-time Updates

- Basically, refreshing should reflect the latest data.
- Real-time updates via WebSocket, etc. are **bonus points**

9. Additional Notes

- The core of this contest is **AI-Native development capability**, so no restrictions on frameworks/libraries
- Team invite, team member-based assignee assignment recommended to implement simply
- Both email+password-based authentication and Google OAuth are **required**
- Your own structure/extension design is welcome, explaining in README can earn bonus points