**⏱️ Timelog Summary & Jira Integration**

A standout feature providing insights into actual work done:

**Key Features:**

* **Date Range Filtering**
* **Function Filtering** (e.g., QA, Development, Design)
* **User Search Filter (on UI)** for quick drill-down without new API calls.
* **Summary Table** with:
  + User-wise logged hours
  + Expected hours
  + Status color-coded (red = low, green = good)
  + Link to Jira issue navigator
* **Detailed Accordion View per User**:
  + Issues, parent summary, worklog hours
  + Expand/Collapse all
* **AI Summary**: Auto-generates summary of user work distribution (via OpenAI backend)
* **Jira Modal Preview**:
  + Click on Jira issue opens modal with an iframe preview inside the app.
  + Zoom support for better readability.

**6. 🔗 Jira ID Mapping**

* Admin interface to map internal users to their Jira accountIds.
* Critical for fetching worklogs and associating with the correct internal users.

**7. 🔍 Visual AI Testing (Experimental Module)**

* Placeholder for integrating screenshots or visual regression comparison between branches/sprints.
* Markdown preview support for visual notes.

**8. 🗓️ Sprint Calendar**

* Calendar interface (likely using FullCalendar.js) showing sprint boundaries, milestones, or tasks.
* Can be expanded for drag-and-drop rescheduling or backlog mapping.

**9. 🧩 Technical Stack & Integrations**

* **Frontend:** Bootstrap 5, jQuery, Bootstrap Select, FullCalendar
* **Backend:** Flask, SQLAlchemy ORM, OpenAI API (for AI summary)
* **Jira Integration:** Jira REST APIs v2 and v3 for issues and worklogs
* **Templating:** Jinja2 with dynamic Flask context
* **Modular Routing:** Clean route separation for each functionality

**10. ✅ User Experience Highlights**

* **Live dropdowns** with search and multiselect
* **Real-time UI filtering** (without server hit)
* **Error handling** (API failure fallback, form validation)
* **Consistent navigation bar and loader overlay**