SCRUM USING JIRA

Shoug Alomrn

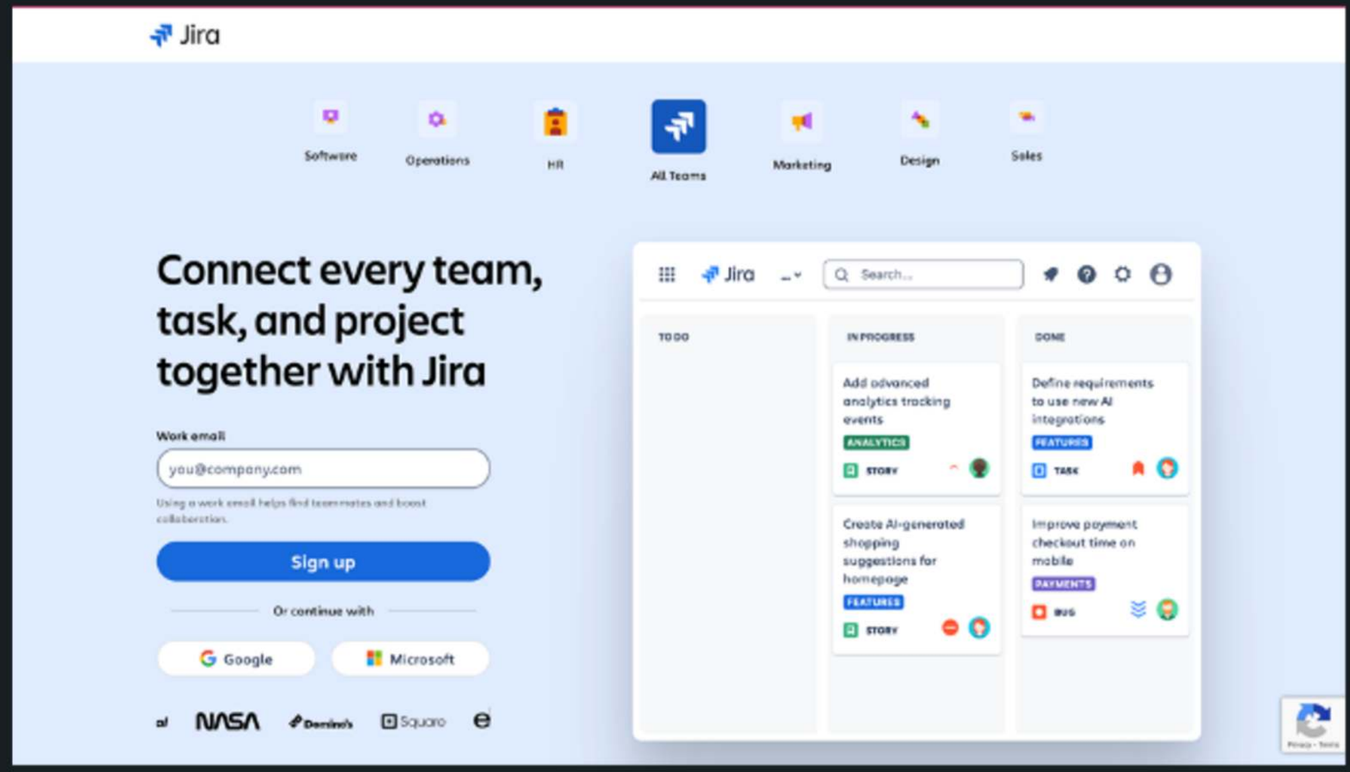
START PRESENTATION

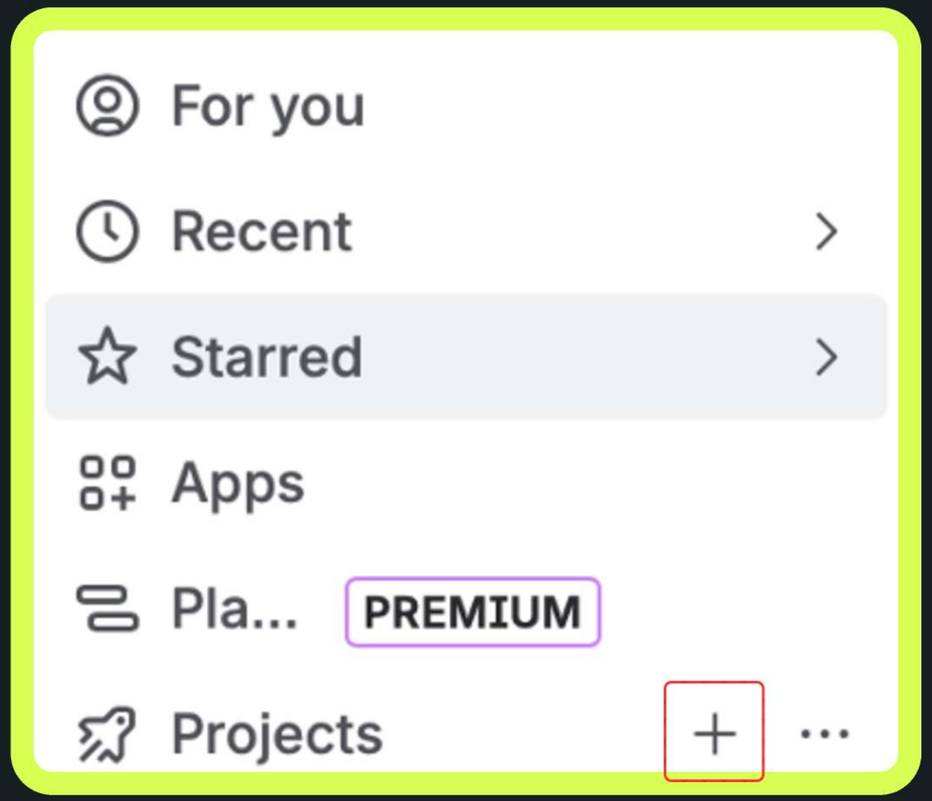
SIGN UP

Account Registration

To begin, we accessed Jira Software by navigating to atlassian.com and selecting the free sign-up option. Using a dedicated team email address, we completed the registration process. We then chose a unique site name to host our project, finalizing the creation of our Jira workspace.

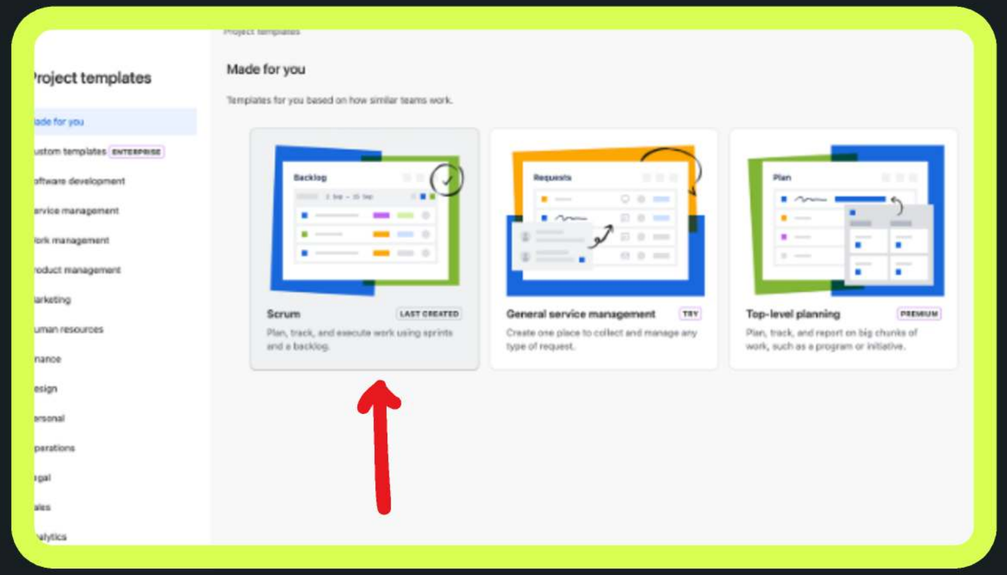
Learn More..

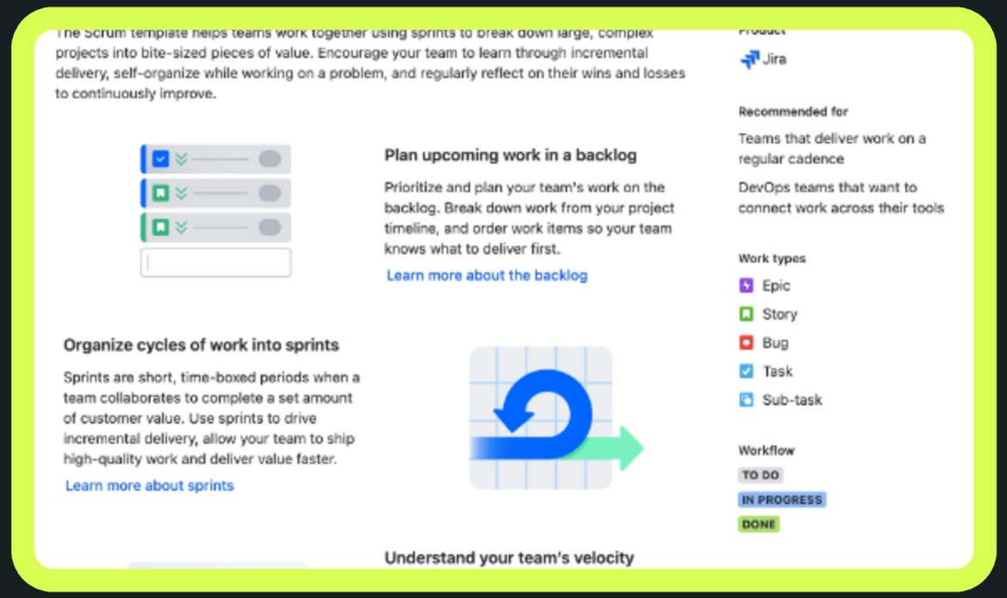




PROJECT CREATION

Upon logging in, we initiated a new project by selecting the "Create Project" button from the main dashboard.

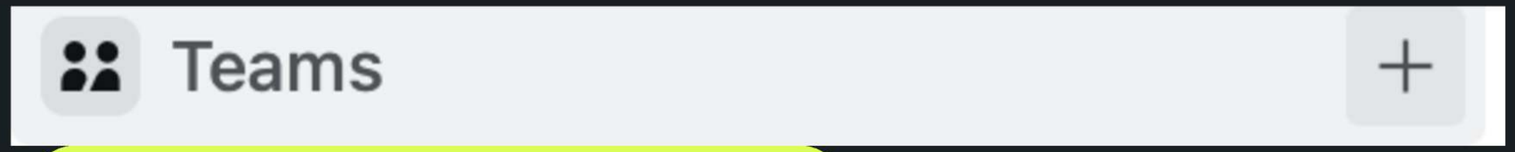


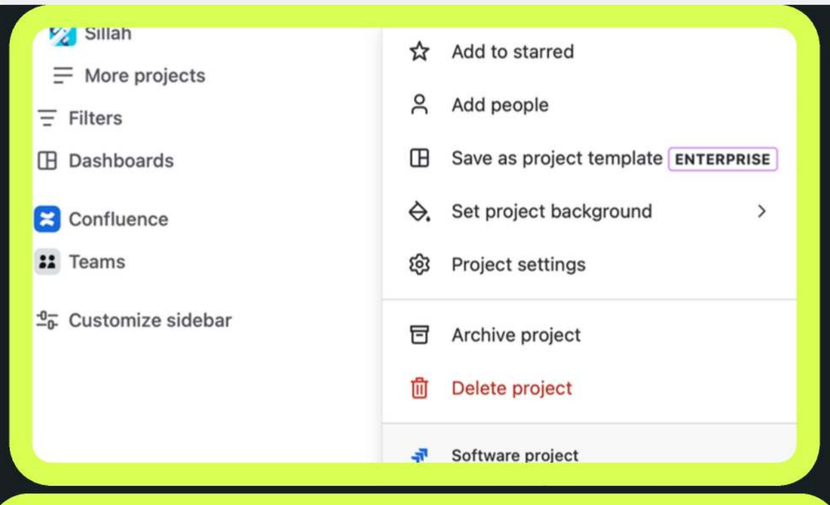


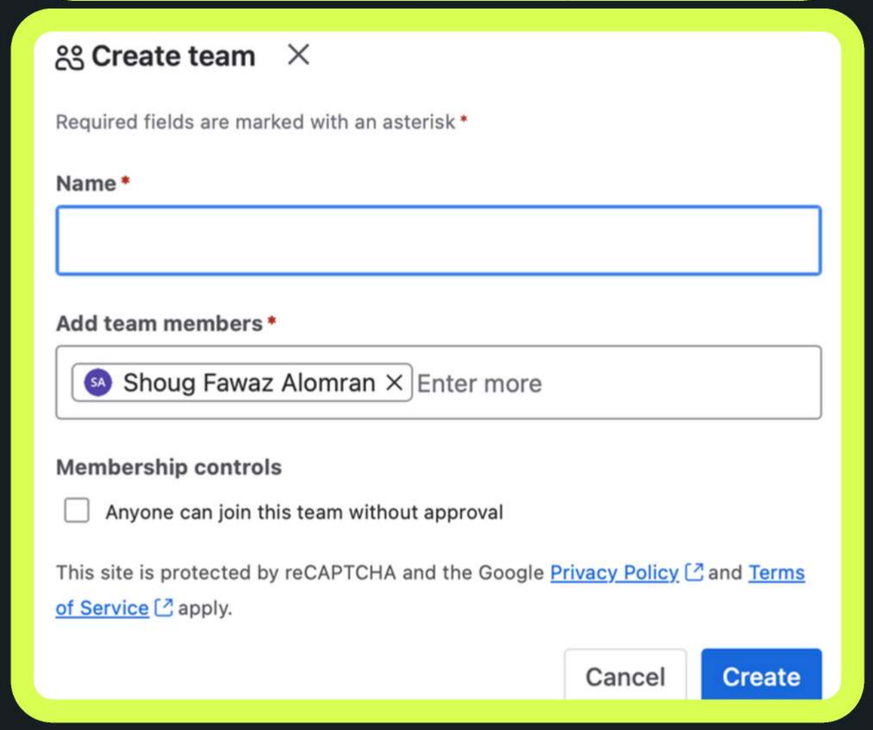
TEMPLATE: SCRUM

Selecting the Scrum Framework

To establish a workflow for sprint-based development, we selected the "Scrum"template during the project creation process. This automatically configured a product backlog and a scrum board for our team.







CREATE YOUR TEAM

Inviting and Managing Team Members

A core step in setting up the Jira project was inviting team members to ensure collaborative task management and visibility.

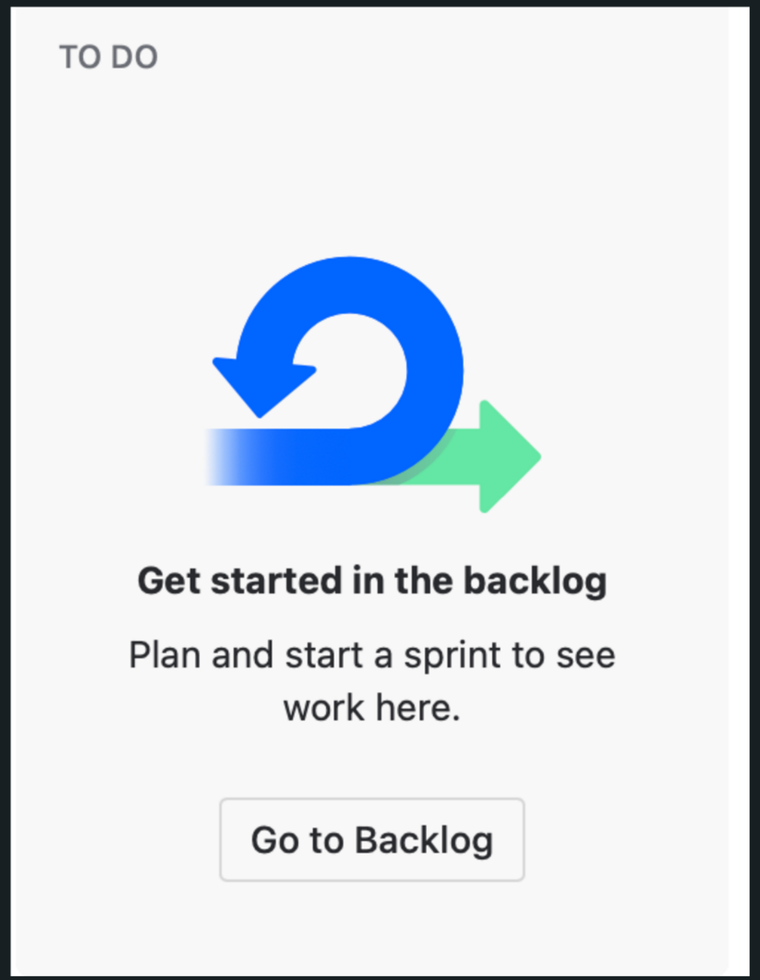
Process:

1. From the project sidebar, we navigated to Project Settings and selected People.
2. We clicked the "Add People" button.
3. We entered the email addresses of our teammates (Shoug Alomran, Aljawarah Alsaleh, Aljawhara Alruzuq, Aljoharah Albawardi).
4. The system sent email invitations to each member.

Outcome:

Once the invitations were accepted, all team members appeared in the project's user list. This allowed us to:

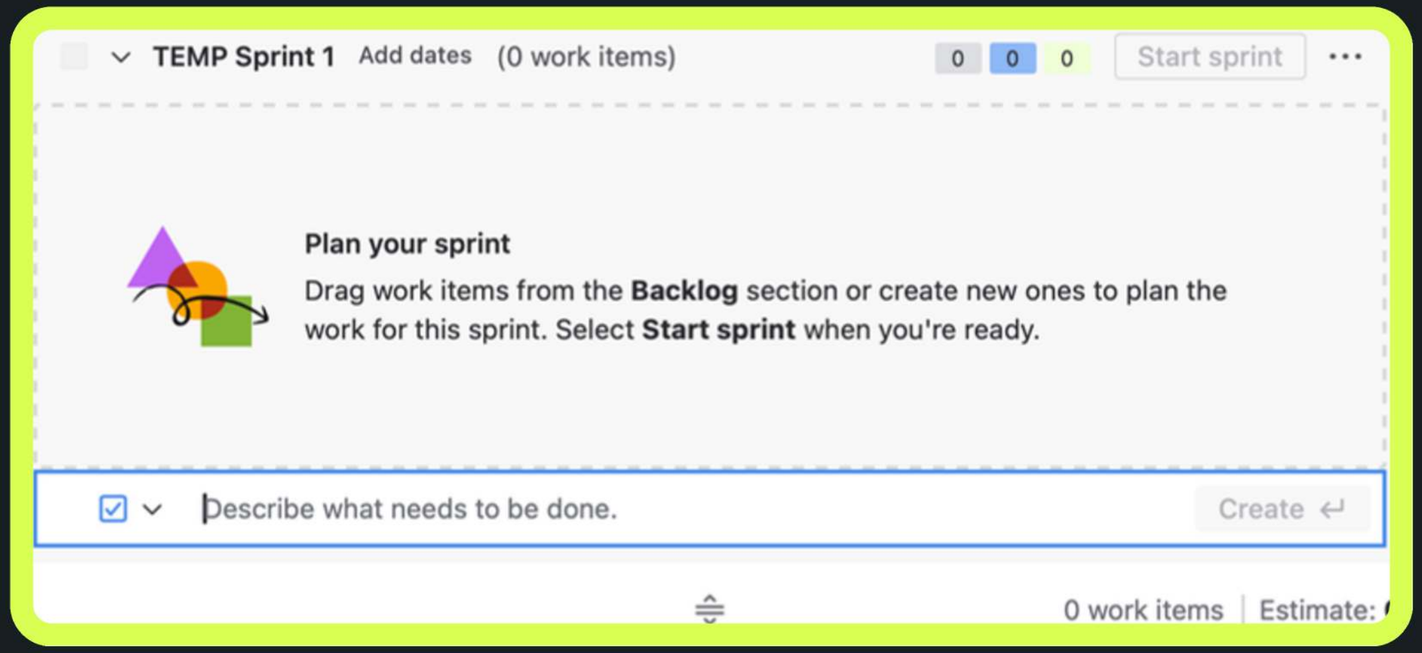
- Assign stories and tasks to specific individuals from the "Assignee" field.
- Filter the board and backlog to view any team member's workload.
- Ensure collaboration and transparency, as every member could view, comment on, and update issues across the project.

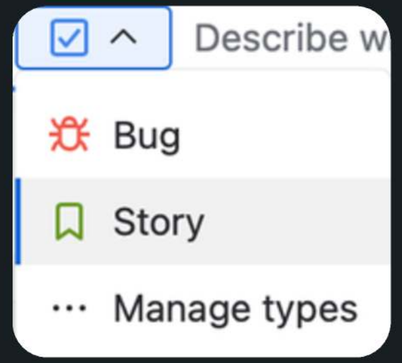


ENTER BACKLOG

Populating the Product Backlog

We navigated to the Backlog view and used the "Create Issue" function to add our initial user stories. Each story was defined with a title, description, story point estimate (1 point = 2 hours), and assignee. Before organizing them into Epics, this created a prioritized list of all required functionalities for sprint planning.





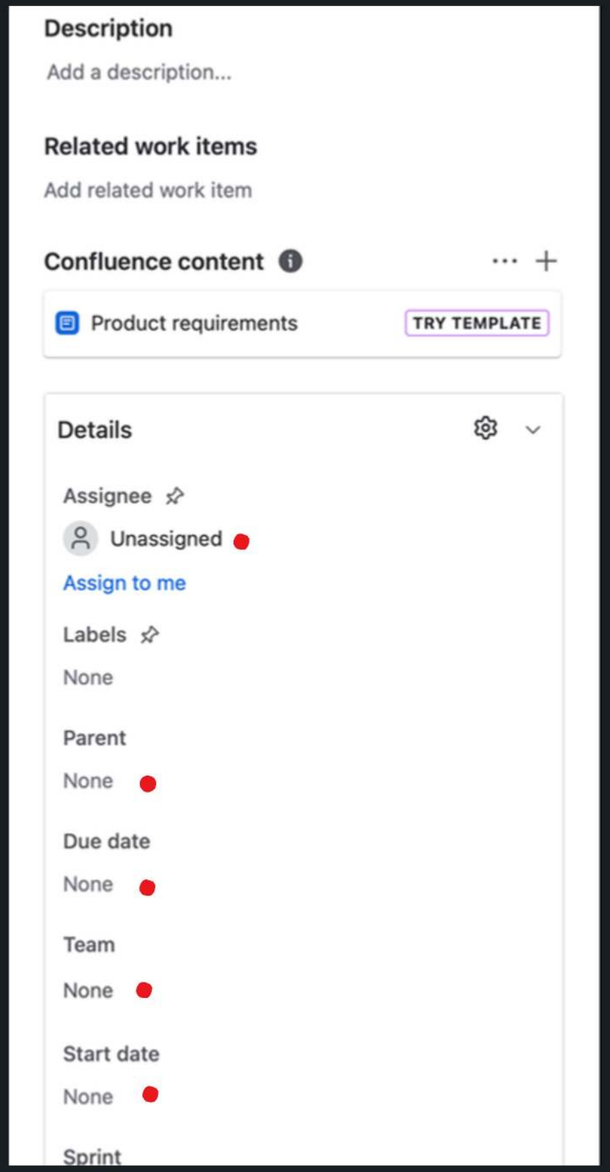
CREATING A STORY IN THE BACKLOG

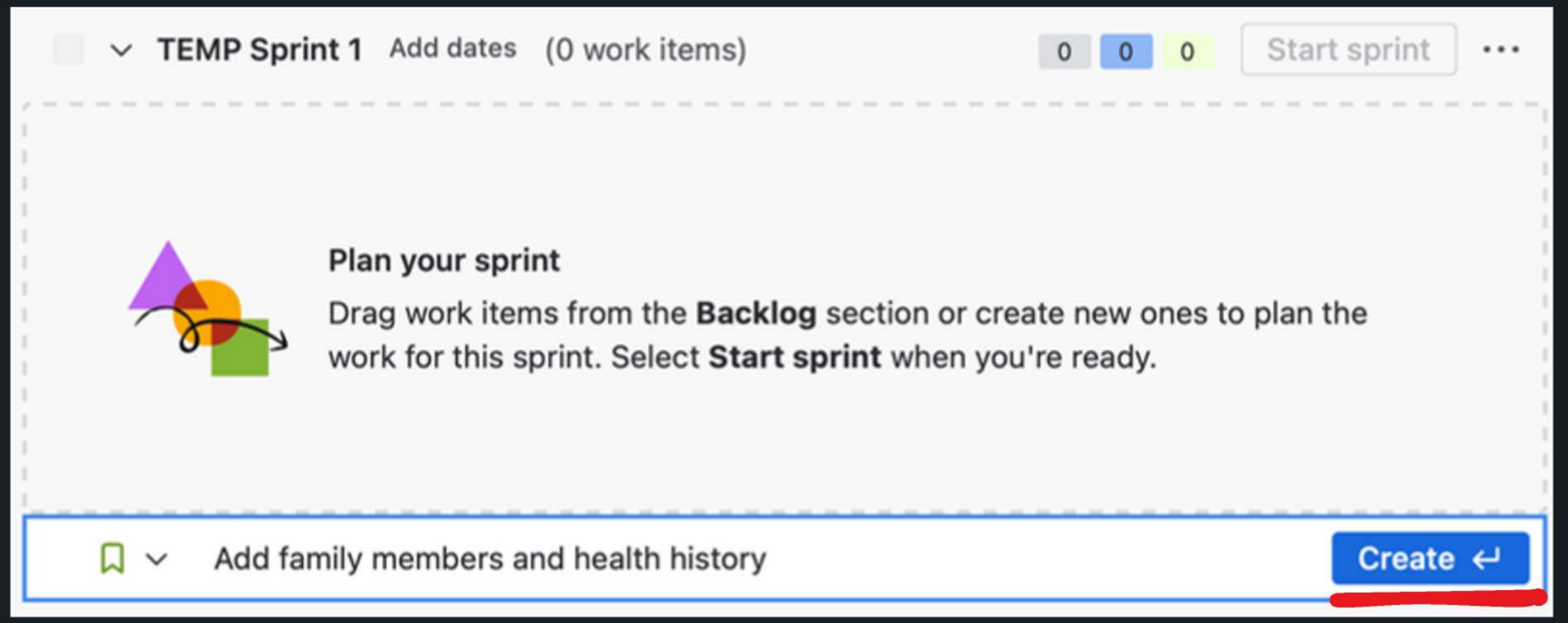
After navigating to the project's Backlog, we created a new issue by clicking the "Create" button (usually represented by a + icon or a blue “Create” button at the top of the screen).

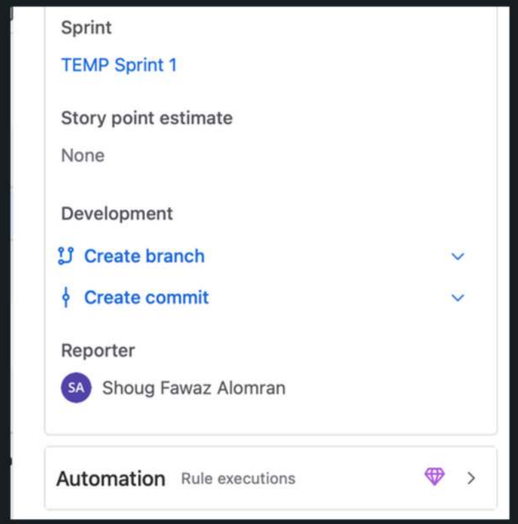
This opened a dialog box for creating a new issue. The default issue type is often a “Task.” To create a User Story, we clicked on the Issue Type field, which opened a dropdown menu with options like Bug, Story, and Epic. From this list, we selected “Story”.

The primary input field in this dialog is placeholder with the text “Describe what needs to be done...”. This is where we entered the concise, user-centered title for each story, summarizing the functionality from the user's perspective.

This process was repeated for each user story we added to the backlog.







COMPLETING THE STORY CREATION FORM

After selecting "Story" as the issue type, you are presented with a detailed form to fully define the work item. Here is an explanation of the key fields and their purposes:

1. Title ("Add family members and health history")

- What it is: This is the core description of the story, replacing the "Describe what needs to be done..." placeholder. It should be a concise, user-focused goal.
- Example: "User can add a new family member and link health conditions."

2. Description

- What it is: A rich-text field for detailed information. This is where you write the full user story narrative, acceptance criteria, and any specific technical or functional details.
- Example: "As a family admin, I want to add a parent or child to the family tree so that I can build a comprehensive health history map."
  - Acceptance Criteria:
  - Form must capture Name, Relationship, Date of Birth.
  - User can associate one or more known health conditions from a predefined list.

3. Details Section

- Assignee: Used to assign the story to a specific team member responsible for its completion (e.g., Shoug Alomran).
- Labels: Tags for easy filtering and categorization (e.g., family-tree, backend, database).
- Parent: Used to link this story to a larger Epic (e.g., "Family Health Tree & Profile Management"). This is crucial for organization.

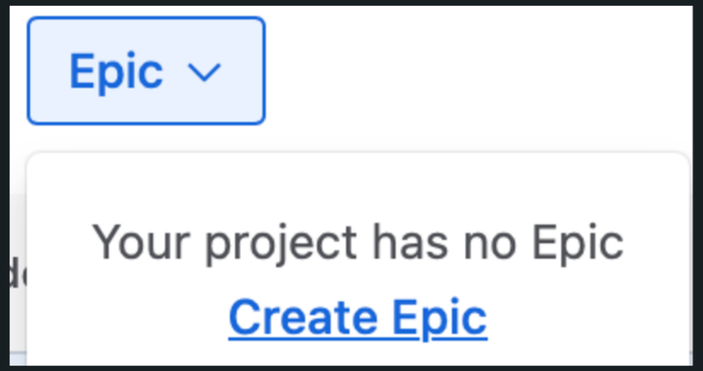
4. Smart Section (Agile-Specific Fields)

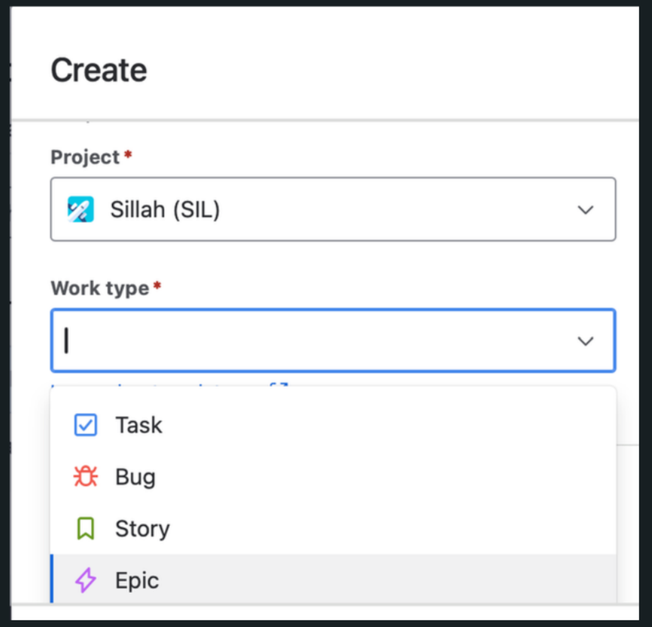
- Sprint: This dropdown is used to assign the story to a specific, active, or future sprint during planning.
- Story Point Estimate: The field for entering the effort estimate for this story (e.g., 5 points).
- Reporter: Automatically set to the person who created the issue.

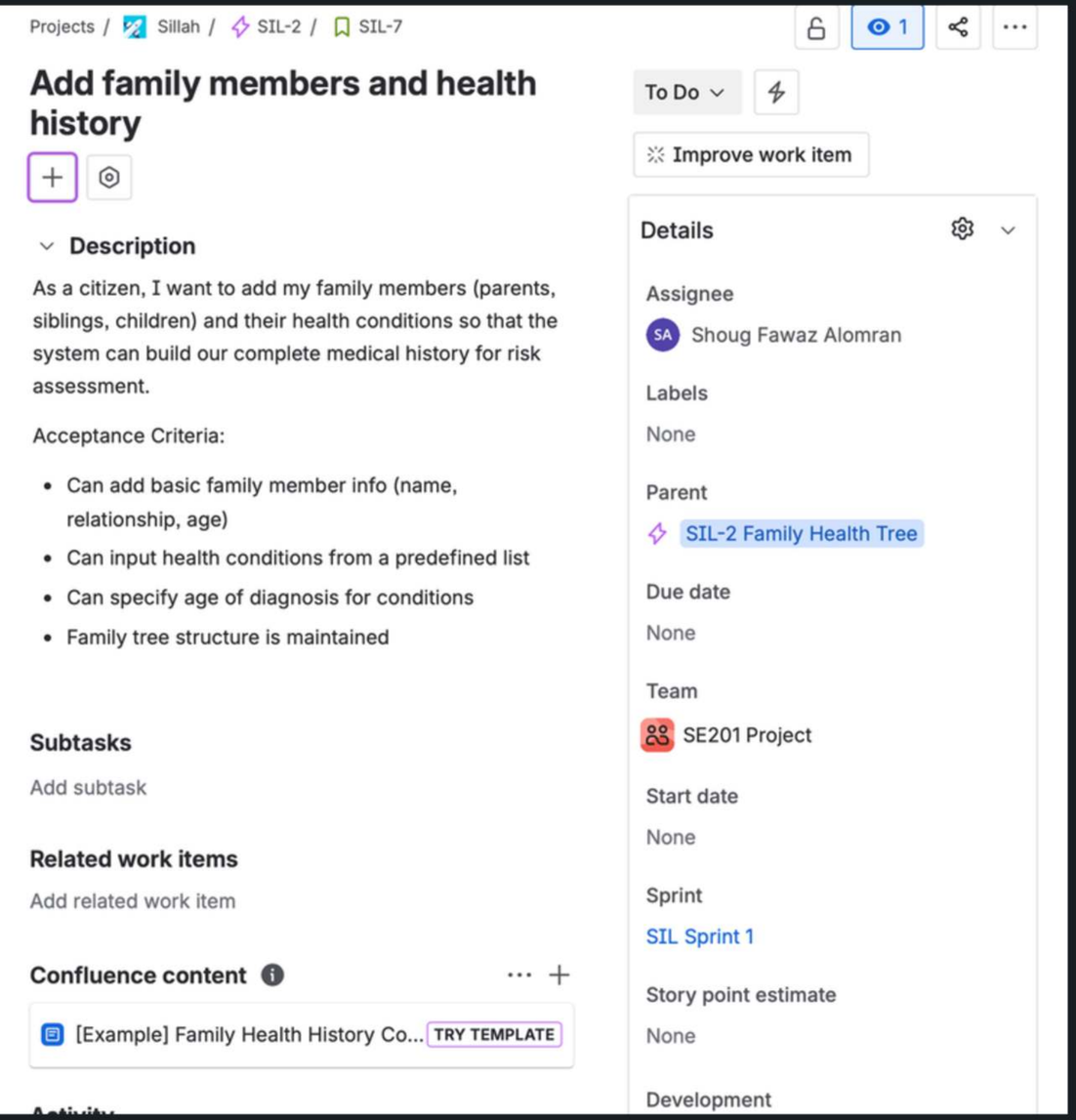
CREATING EPICS FOR BACKLOG ORGANIZATION

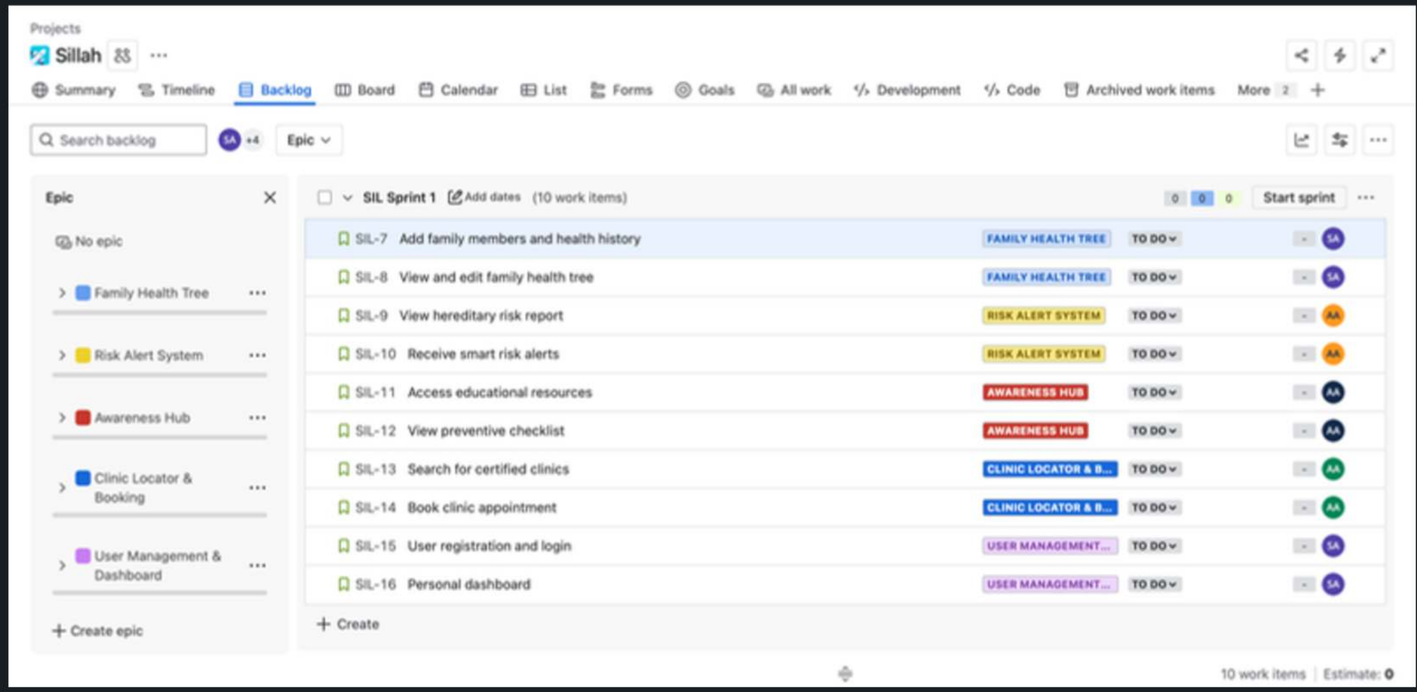
We created Epics to group related user stories under major features (e.g., "Family Health Tree," "Risk Engine"). This provides high-level project structure, improves progress tracking per feature, and clarifies how individual stories contribute to broader goals.

To create them, we used the **"Create"** button in the Backlog, selected **"Epic"**, and named each one. We then linked relevant stories to their corresponding Epic using the **"Parent"** field.









FINAL BACKLOG STRUCTURE

The image shows our finalized Jira backlog, structured for sprint execution. The left panel contains “**Sprint 1**" with committed user stories ready for development. The right panel displays the **Product Backlog**, where all remaining stories are organized under their parent Epics like "Family Health Tree" and "Risk Assessment Engine." This clear hierarchy allows us to easily drag stories into future sprints during planning sessions while maintaining visibility into how each task contributes to larger project goals.