

# Software Engineering Task-4

**Kavya Aggarwal**

**HU22CSEN0100288**

## Create a Project Plan using JIRA Tool

### 1. Site Naming and Account Login



**Welcome back, Kavya**

You're logged in as **kaggarwa@gitam.in** [Switch account](#)

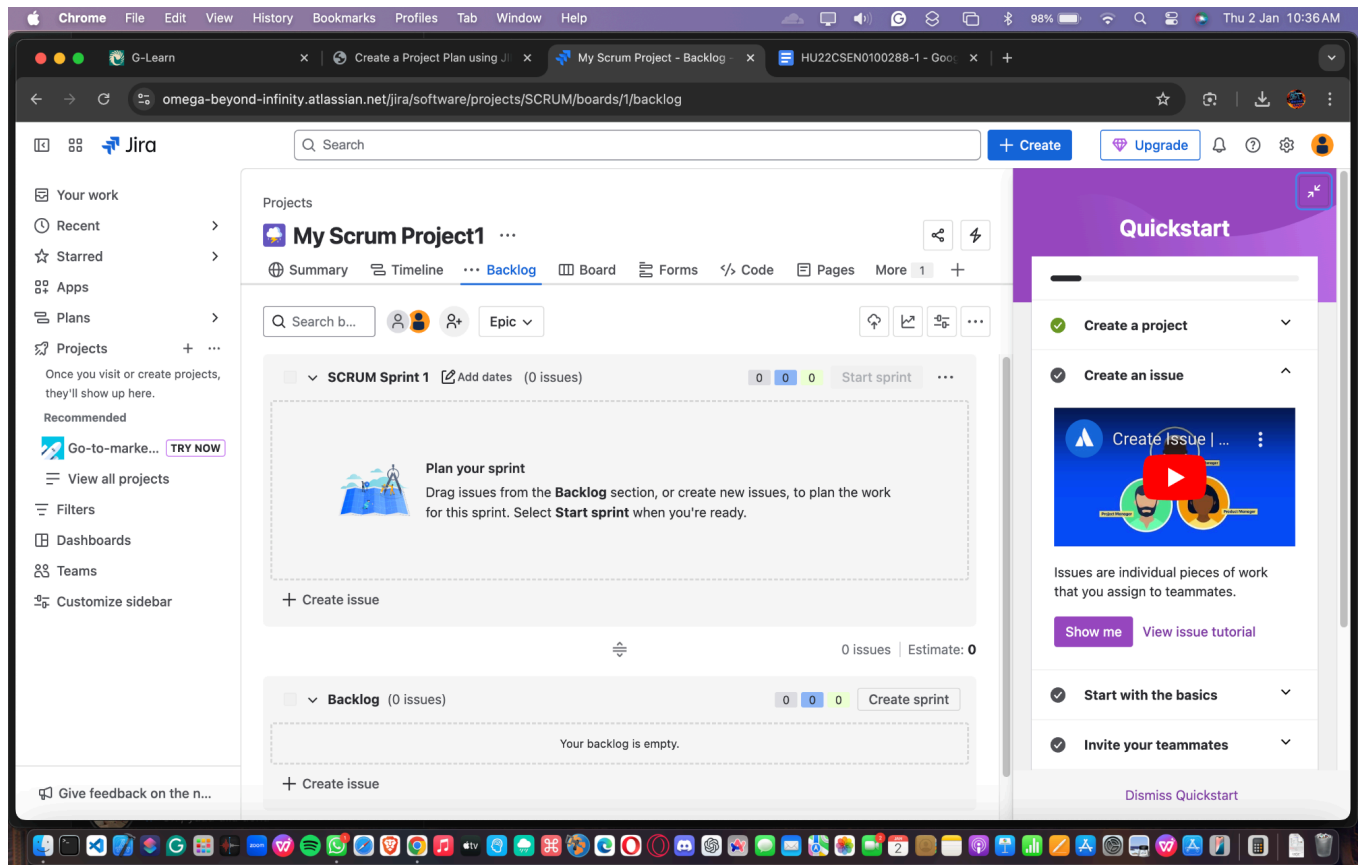
**Jira**  
omega-beyond-infinity

[Go to Jira](#)

[Or start a new site](#)



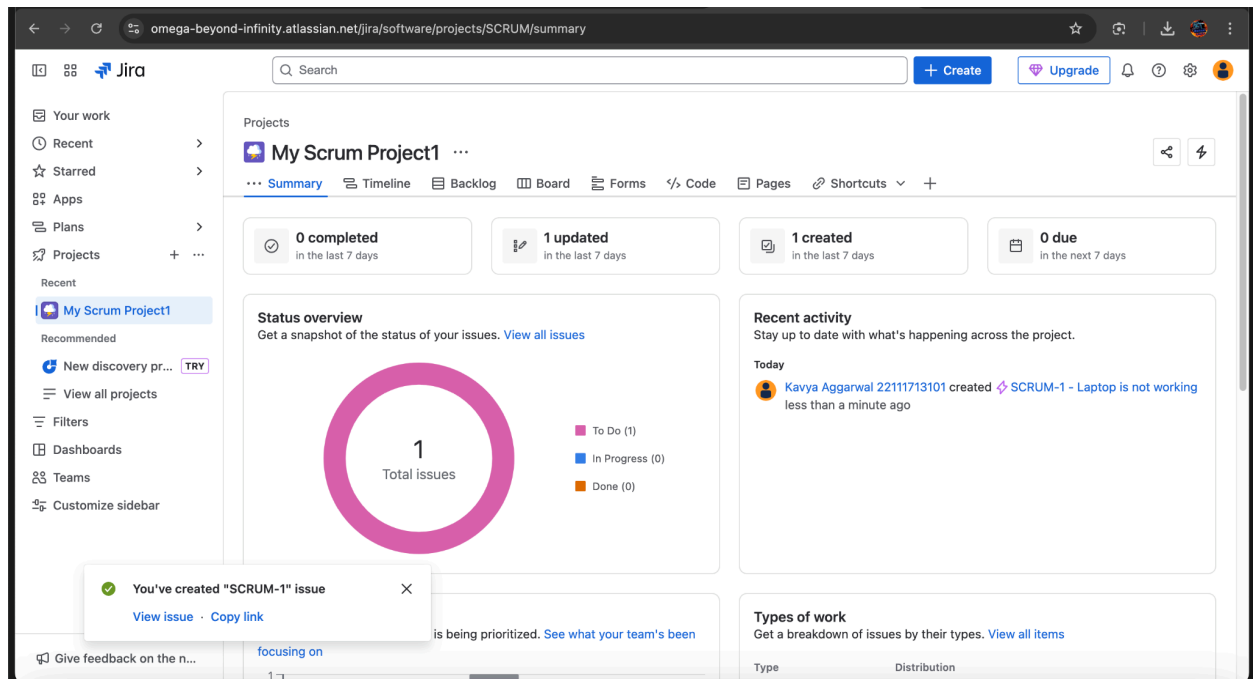
## 2. Jira Site Creation



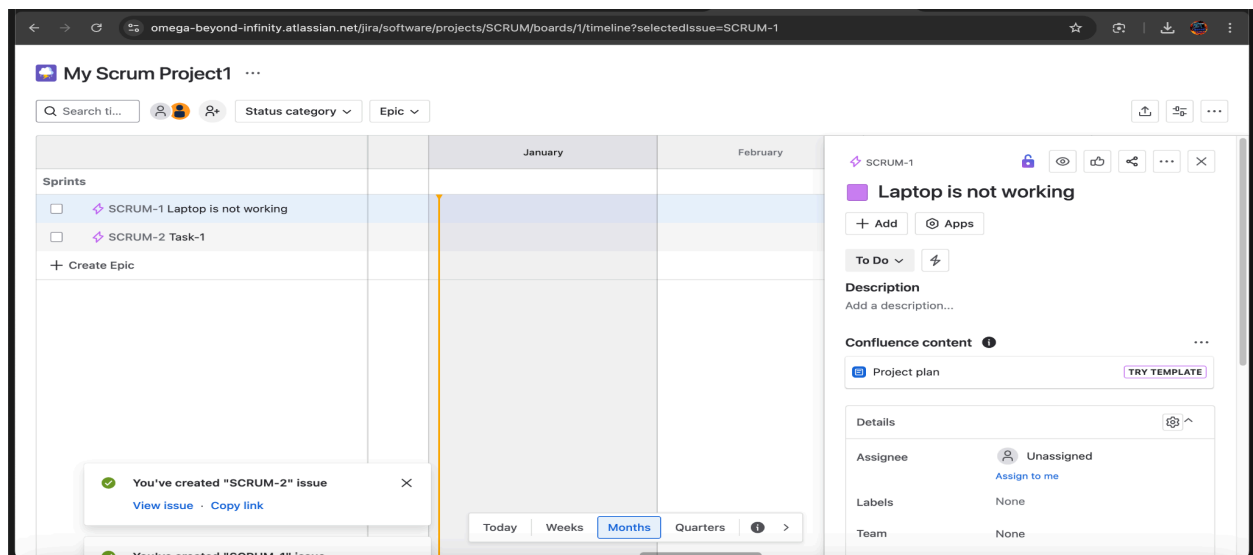
## 3. Jira Site

<https://omega-beyond-infinity.atlassian.net/jira/software/projects/SCRUM/boards/1/backlo...>

## 4. Creating an Epic in the newly created Jira Project



## 5. All epic Tickets created



## **6. JIRA OVERVIEW**

**JIRA is a software tool developed by Atlassian, primarily designed for bug tracking, issue tracking, and project management. The name “JIRA” originates from the Japanese word “Gojira,” meaning “Godzilla.” Initially developed to handle software bugs, JIRA has evolved into a comprehensive project management tool, widely adopted for managing software development, mobile app issues, and agile project workflows.**

## **7. Uses of JIRA**

- JIRA serves diverse purposes across industries. Below are its most prominent use cases:**

**1.Bug Tracking: Tracks and manages bugs, ensuring they are resolved efficiently.**

**2.Project Management: Provides tools for planning, tracking, and managing projects from start to finish.**

**3.Software Development: Supports agile frameworks like Scrum and Kanban to manage software lifecycles.**

**4.Agile Methodology: Facilitates agile workflows using Scrum and Kanban boards.**

**5.Product Management:** Helps define product requirements, prioritize features, and align development goals.

**6.Requirements and Test Case Management:** Tracks requirements and integrates with tools like Zephyr or Xray for test management.

**7.Task Management:** Organizes tasks with priority, assignees, and deadlines to enhance productivity.

## **8. JIRA Features**

### **1. JIRA Scheme**

- **JIRA schemes** define the structure and behavior of projects. They include:
  - **Workflows:** Define the life cycle of an issue, including statuses and transitions.
  - **Issue Types:** Categorize work into types like Bug, Task, and Story.
  - **Custom Fields:** Enable users to add unique data fields to issues (e.g., severity level, estimated time).

- **Screens:** Control the fields visible during issue creation, editing, or resolution.
- **Field Configuration:** Customize the behavior and appearance of fields.
- **Notifications:** Define email updates for specific events (e.g., issue creation, assignment).
- **Permissions:** Control user access to projects and operations.

## **2. JIRA Components**

- **Components are subsections within projects that help organize issues:**
  - **Examples include grouping issues by features, modules, or teams.**
  - **Enhance reporting by providing statistics based on components.**
  - **Details include:**
    - **Name:** Identifier for the component.
    - **Description:** Brief explanation of its purpose.
    - **Lead:** Person responsible for the component.

- **Default Assignee:** The individual automatically assigned to issues within the component.

## **9. JIRA Issues and Issue Types**

### **What is a JIRA Issue?**

- A JIRA issue represents any task, bug, or feature related to a project. Each issue includes:
  - **Types:** Categories like Bug, Task, Story, or Sub-task.
  - **Attributes:** Characteristics like statuses, resolutions, and priorities.

### **Types of Issue Type Schemes**

#### **1. Default Issue Type Scheme:**

- Automatically applies to all new issues in a project.
- Includes standard types like Bug, Task, and Story.

#### **2. Agile Scrum Issue Type Scheme:**

- Tailored for agile projects, including Epic, Story, and Sub-task.

## **JIRA Issue Attributes**

- Each issue has multiple attributes that define its properties and state:
  1. **Statuses:** Represent the progress of an issue.
    - **Common statuses:** To Do, In Progress, Open, Resolved, Closed, Reopened.
  2. **Resolutions:** Indicate why an issue was resolved.
    - **Examples:** Fixed, Won't Fix, Duplicate, Done, Cannot Reproduce.
  3. **Priorities:** Reflect the urgency of an issue.
    - **Examples:** Critical, Major, Minor, Blocker, Trivial.

## **10. Key JIRA Functionalities**

### **1. Issue Security Schemes**

- **Control who can view specific issues.**
- **Use security levels to restrict sensitive data to authorized users.**



## **2. System Administration**

**Includes advanced administrative features like:**

- **Audit Logs:** Track changes to issues.
- **Issue Linking:** Link related or dependent issues.
- **Mail Notifications:** Configure email alerts via POP/IMAP.
- **Events:** Customize system events for workflows (e.g., issue created).

## **3. Watch List**

- **Allows users to subscribe to updates for specific issues and receive notifications on any changes.**

## **Workflows**

**Workflows define how issues move through different stages during their lifecycle. Key elements include:**

### **1. Statuses:**

- **Examples:** Open, In Progress, Resolved, Reopened, Closed.

### **2. Transitions:**

- **Actions that move issues between statuses (e.g., “Start Progress”).**

### **3. Other Workflow Elements:**

- **Assignees:** Responsible individuals.
- **Resolution:** Explains why an issue moved to the closed state.
- **Conditions, Validators, and Properties:** Define transition rules and ensure valid transitions.

## **11. JIRA Agile**

### **Scrum:**

- **Used for managing sprints and iterative development.**
- **Workflow stages:** To Do → In Progress → Done.
- **Features:**
  - **Sprints:** Fixed time-boxed iterations.
  - **Backlog:** Prioritized task list.
  - **Reports:** Includes Burndown Charts and Velocity Charts.

### **Kanban:**

- **Focused on continuous delivery and flow.**
- **Features:**
  - **Workflow Visualization:** Displays tasks across stages.

- **WIP Limits:** Prevents overloading by capping work in progress.
- **Cycle Time Measurement:** Tracks task completion times.

## **12. Creating and Managing Issues**

### **1. Creating Issues:**

- **Log in to JIRA.**
- **Open the Projects tab.**
- **Select a template (e.g., Simple Issue Tracking, Agile Kanban).**
- **Provide issue details, assign responsibilities, and set priorities.**

### **2. Editing and Searching Issues:**

- **Use the Issues Menu to:**
  - **Edit, comment, or reassign issues.**
  - **Apply filters for quick searches.**

### **3. Sub-Tasks:**

- **Split parent issues into smaller tasks.**
- **Notes:**
  - **Sub-tasks cannot have their own sub-tasks.**
  - **Sub-tasks can be converted into parent issues.**

### **4. Cloning Issues:**

- **Create duplicate issues to allow parallel work by multiple teams.**

## **13. Reports**

- **JIRA includes various reporting tools for progress tracking and analysis:**
  1. **Burndown Chart: Tracks completed and remaining work in sprints.**
  2. **Epic and Sprint Reports: Measure sprint and epic progress.**
  3. **Pie Charts: Visualize priorities or issue statuses.**
  4. **Control Chart: Analyze issue cycle times and bottlenecks.**

## **14. Plugins**

**- JIRA supports numerous plugins to enhance functionality:**

- **Examples:**

- **Zephyr: Test management.**
- **GitHub: Repository integration.**
- **Salesforce: Customer relationship management.**

- **Benefits:**

- **Extend JIRA's capabilities for reporting, issue linking, and data synchronization.**

## **15. Summary**

**JIRA is a powerful, versatile tool for managing software development, agile projects, and tasks. It offers customizable workflows, robust reporting, and seamless integration with other tools. Its adaptability makes it indispensable for teams striving for efficiency and collaboration.**