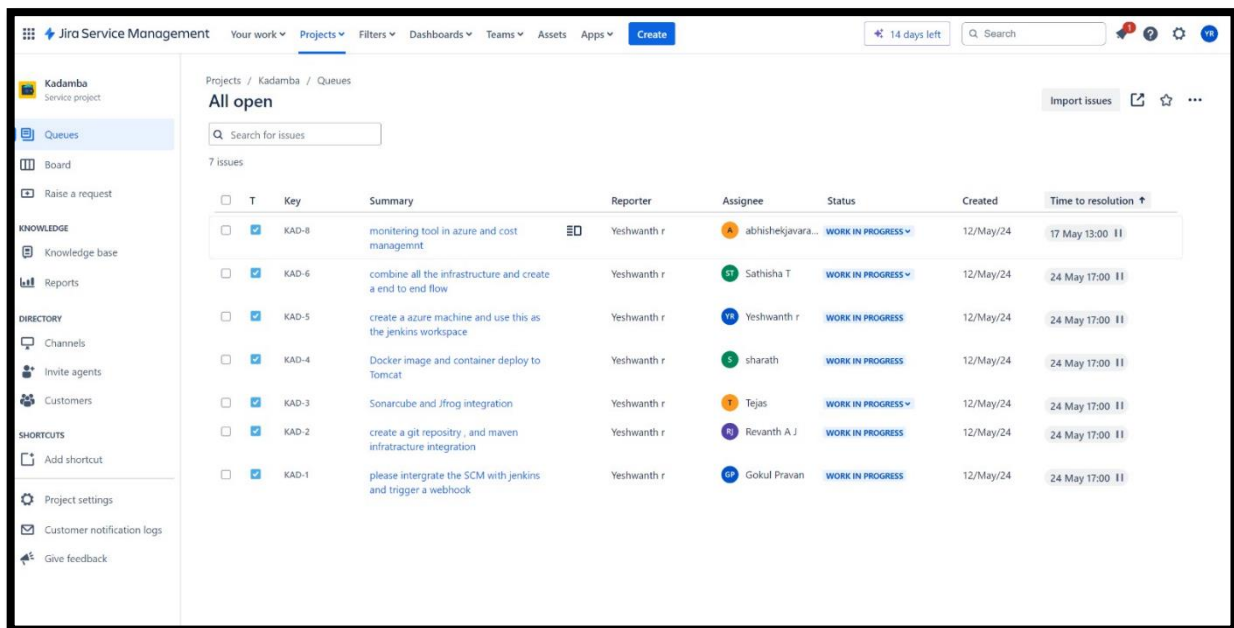


Team: **Kadamba**

Team members:

1. Abbu Yeshwanth Ramesh
2. Abhishek J M
3. Gokul Pravan
4. Revanth Reddy A J
5. Sathisha T
6. Sharath Kumar B M
7. Tejas H R



The screenshot displays the Jira Service Management interface for the 'Kadamba' project. The left sidebar contains navigation options like 'Queues', 'Board', 'Raise a request', 'KNOWLEDGE', 'Reports', 'DIRECTORY', 'Channels', 'invite agents', 'Customers', 'SHORTCUTS', 'Project settings', 'Customer notification logs', and 'Give feedback'. The main area shows a list of 7 open issues under the 'All open' filter. Each issue entry includes a checkbox, a key (KAD-8 to KAD-1), a summary, the reporter (Yeshwanth r), the assignee, the status (WORK IN PROGRESS), the creation date (12/May/24), and the time to resolution.

|                          | T                                   | Key   | Summary   | Reporter    | Assignee          | Status           | Created   | Time to resolution |
|--------------------------|-------------------------------------|-------|---|-------------|-------------------|------------------|-----------|--------------------|
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-8 | monitoring tool in azure and cost managemnt                   | Yeshwanth r | abhishekjavara... | WORK IN PROGRESS | 12/May/24 | 17 May 13:00 II    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-6 | combine all the infrastructure and create a end to end flow   | Yeshwanth r | Sathisha T        | WORK IN PROGRESS | 12/May/24 | 24 May 17:00 II    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-5 | create a azure machine and use this as the jenkins workspace  | Yeshwanth r | Yeshwanth r       | WORK IN PROGRESS | 12/May/24 | 24 May 17:00 II    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-4 | Docker image and container deploy to Tomcat                   | Yeshwanth r | sharath           | WORK IN PROGRESS | 12/May/24 | 24 May 17:00 II    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-3 | Sonarcube and jfrog integration                               | Yeshwanth r | Tejas             | WORK IN PROGRESS | 12/May/24 | 24 May 17:00 II    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-2 | create a git repository, and maven infrastructure integration | Yeshwanth r | Revanth A J       | WORK IN PROGRESS | 12/May/24 | 24 May 17:00 II    |
| <input type="checkbox"/> | <input checked="" type="checkbox"/> | KAD-1 | please intergrate the SCM with jenkins and trigger a webhook  | Yeshwanth r | Gokul Pravan      | WORK IN PROGRESS | 12/May/24 | 24 May 17:00 II    |

<https://github.com/adikarthik/Project.git>

Overall, this project will manage the user registration, login, and displaying a welcome page with a list of documents associated with the logged-in user. The registration and login methods use form submission and validation, while the welcome method retrieves and displays user-specific data.

## 1. Version Control System (VCS):

- Ensure your project is hosted on a version control system like Git. This allows for collaboration and facilitates CI/CD workflows.

## 2. Choose a CI/CD Platform:

- Select a CI/CD platform that integrates well with your VCS. Popular options include Jenkins.

### 3. Setup CI Pipeline:

- Create a CI pipeline configuration file (**Jenkinsfile**) in our project repository.
- Define stages and jobs for building, testing, and analyzing our code.
- To analyse the static code, we can use the SonarQube.
- For the given multi-module Java project we can use the below steps.
  - Building the project using Maven.
  - Running static code analysis tools (e.g., SonarQube) for code quality checks.
- Make sure our CI pipeline triggers(webhook) any changes made in the SCM(Git).

### 4. Automate Deployment (CD):

- Configure our CI pipeline to automatically deploy your application after successful testing.
- Deployments can be automated using tools like Ansible, Docker.
- Here we can use Tomcat server to deploy our application.

### 5. Deploy to required environment:

- We can deploy our application to the required environment (e.g., development, QA, production).

### 6. Secrets Management:

- Safely manage sensitive information such as database passwords, API keys, etc., by using a secrets management solution provided by your CI/CD platform or by using encrypted environment variables, RBAC, Azure vaults, IAM.

### 7. Monitoring and Alerting:

- Implement monitoring and alerting for your deployed application using tools like Prometheus, Grafana, or built-in monitoring services provided by cloud platforms.
- Set up alerts for critical metrics (e.g., high CPU usage, memory leaks) By using shell script.

