

ISE-2 FOR Practical

Assignment: Evaluation of JIRA and Testing Tools for Course Scheduling

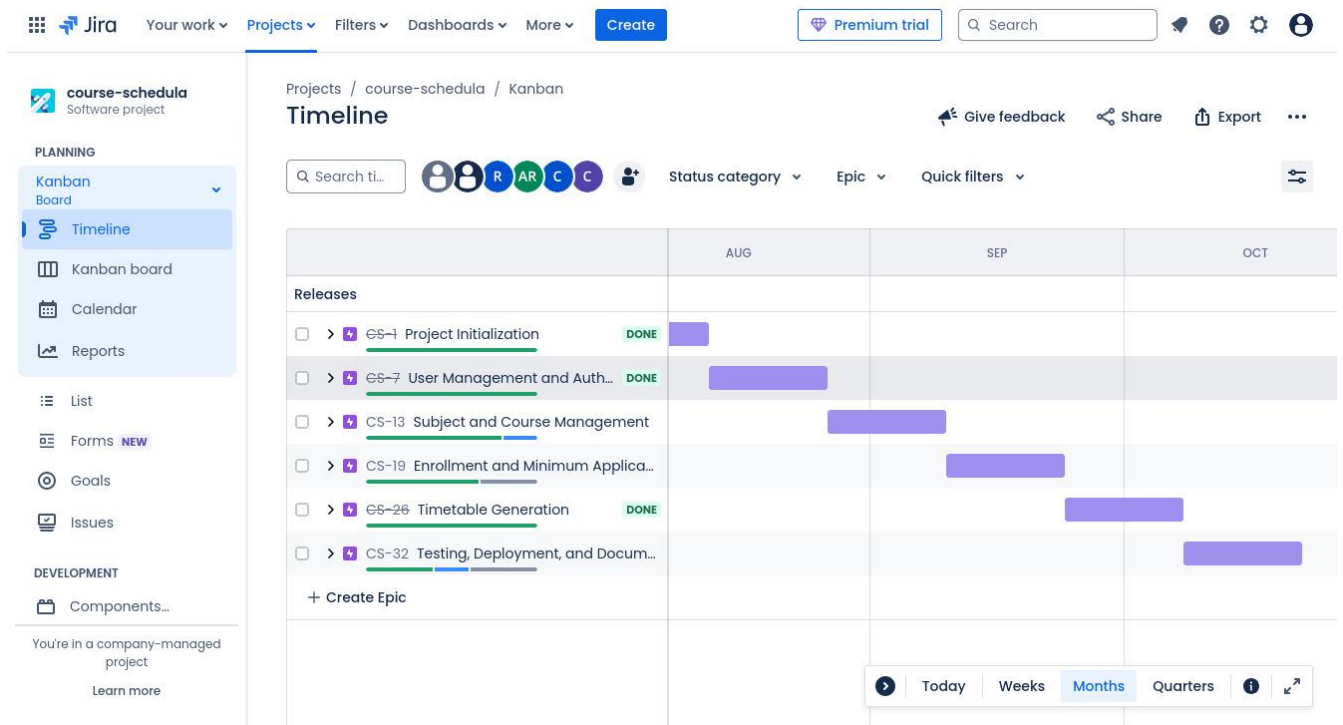
System Total Marks: 25

Based on the requirements specification document provided for the course scheduling system, this assignment evaluates your understanding and practical application of JIRA and testing tools. Answer each question in detail, demonstrating your approach and explaining the use of tools where applicable.

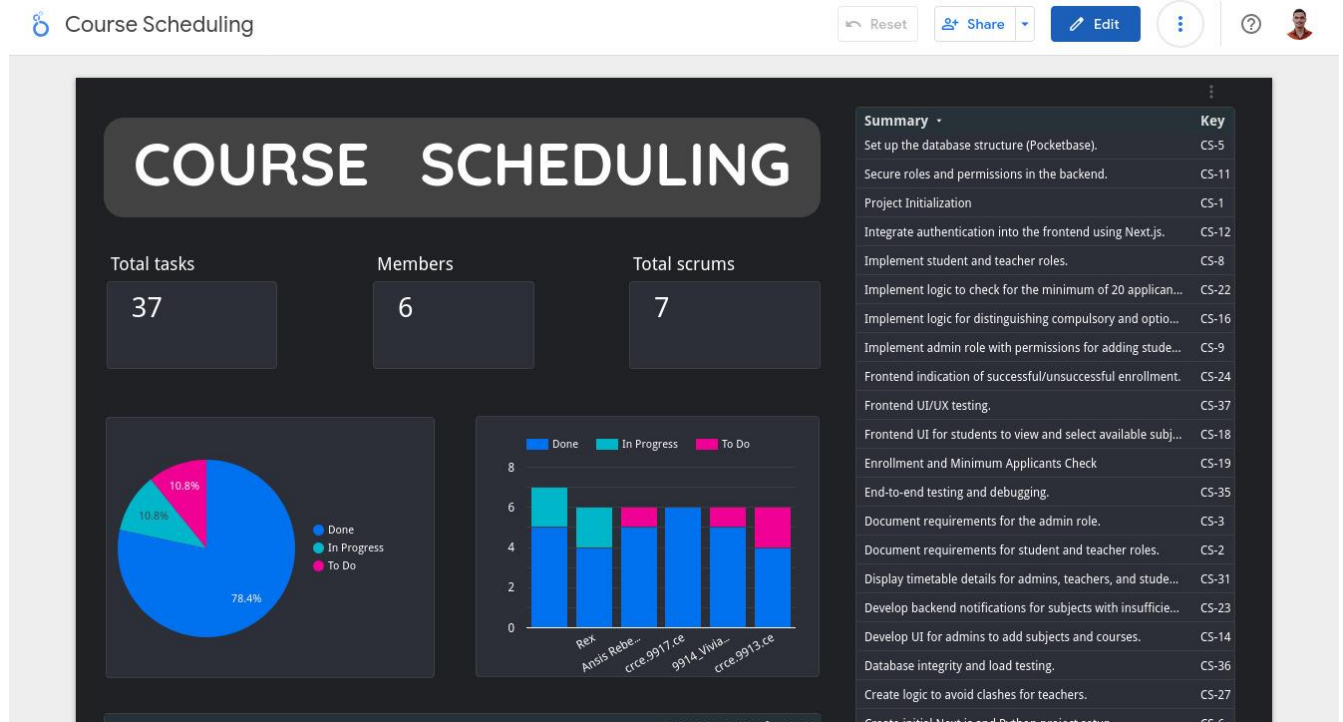
Project Tracking, Requirement Breakdown, and Defect Management in JIRA (10 Marks – CO: CSC502.5)

- **Requirement Tracking: Describe how you would structure a JIRA project to manage the requirements for the course scheduling system, covering project setup, task breakdown, and the use of epics and stories.**

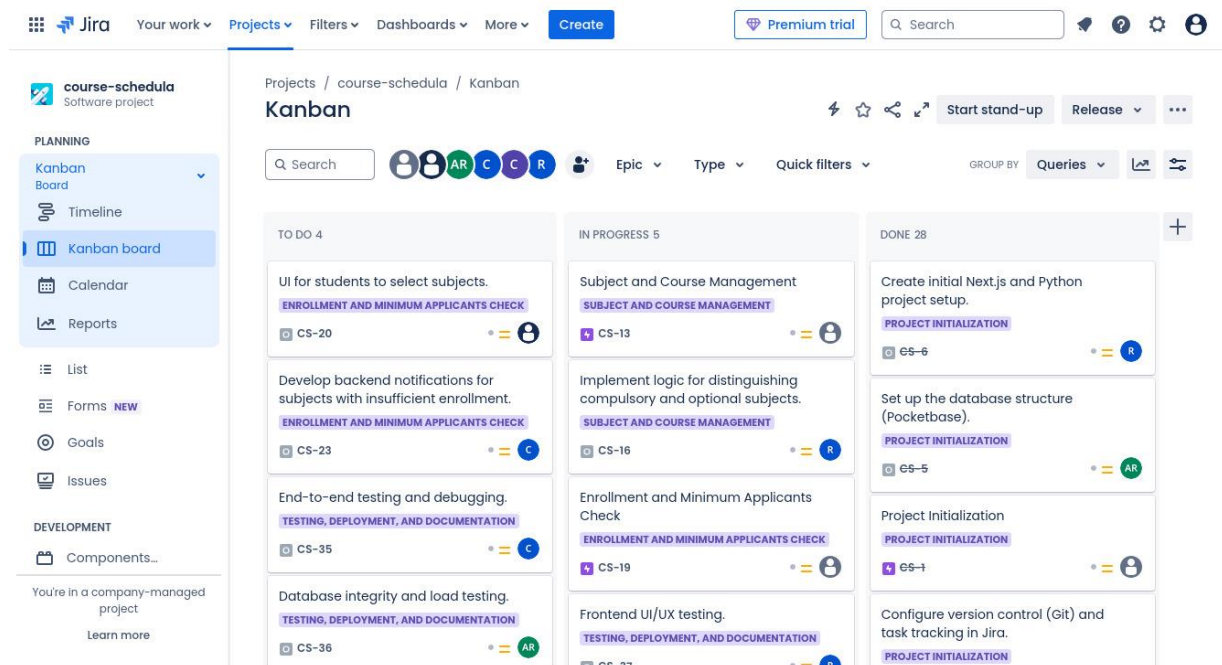
Create a JIRA project with epics for major features, break tasks into stories, and use sprints for milestone tracking.



- **Defect Tracking:** Identify three types of potential errors (e.g., input file errors, scheduling conflicts) based on the requirements, and explain how you would log and manage these issues in JIRA, focusing on risk identification and change management.



- **Progress Monitoring:** Explain how JIRA can help in tracking project progress and implementing requirement changes effectively.



2. Testing Automation and Performance Testing Using Testing Tools (10 Marks – CO: CSC502.4)

- **Test Case Design:** Select three key functional requirements and describe how you would set up automated test cases for these using a testing tool. Include details on validation criteria for inputs, expected outputs, and error handling.

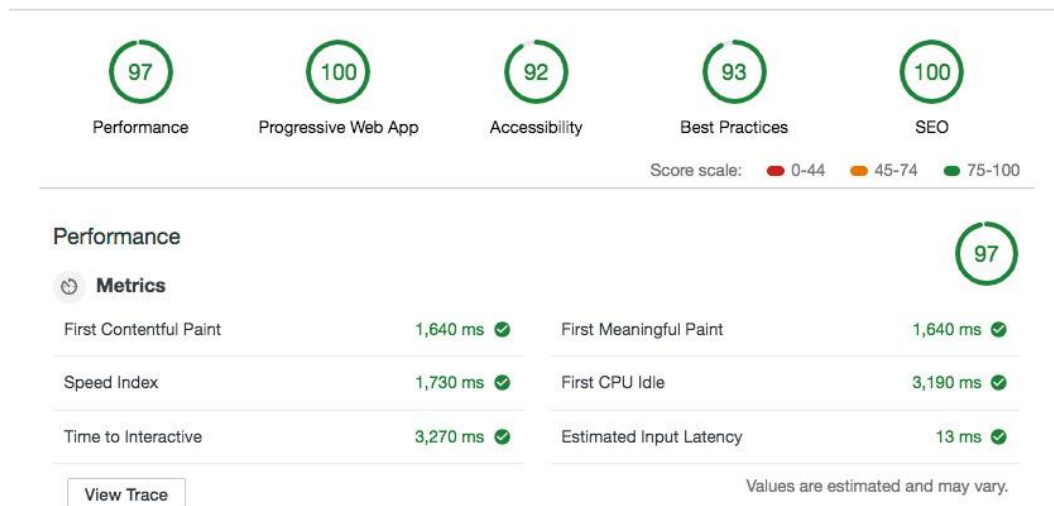
Test Case ID	Description	Expected Result	Status
CSV-001	Valid CSV file upload for teachers	Correct parsing, all teacher data added to system	Pass
CSV-002	Invalid CSV structure (missing columns)	Throws error or rejects file, alerts admin	Pass
CSV-003	Validate classroom seating capacities	Accepts only positive integers; rejects any non-integer input	Pass
CSV-004	Handle duplicate entries	Ignores duplicates or throws a warning	Pass
CSV-005	Large file parsing test	Successfully parses large files (1,000+ records) without performance issues	Pass

Test Case ID	Description	Expected Result	Status
ENR-001	Validate branch selection logic	Correct branches populate based on student input	Pass
ENR-002	Verify elective subject availability	Only electives relevant to branch displayed; all electives loaded for common option	Pass
ENR-003	Ensure input field validation	Rejects empty or invalid data entries	Pass
ENR-004	Boundary test on name field length	Accepts name within 1–100 characters; rejects overflows	Pass
ENR-005	Prevent duplicate enrollments	Does not allow re-enrollment for the same student	Pass

Test Case ID	Description	Expected Result	Status
TBL-001	Ensure no schedule conflicts	Classes do not overlap for students and teachers	Pass
TBL-002	Classroom capacity check during allocation	Classroom only assigned if capacity fits student enrollment	Pass
TBL-003	Confirm lunch break enforcement	1 PM - 2 PM slot remains empty for all schedules	Pass
TBL-004	Weekly subject frequency validation	Each subject occurs exactly 4 times per week, on separate days	Pass
TBL-005	Handle schedule for a maximum number of students	Timetable generated efficiently with large datasets	Pass
TBL-006	Test weekend constraints	No classes scheduled on weekends	Pass

Test Case ID	Description	Expected Result	Status
ACL-001	Verify teacher access control for student data	Teachers can view only students relevant to their assigned subjects	Pass
ACL-002	Test personalized timetable view	Teachers only see their own class schedule	Pass
ACL-003	Test unauthorized access for admin resources	Teachers cannot access admin functions or data	Pass
ACL-004	Validate logout and session termination	After logout, teacher cannot access panel until re-authenticated	Pass
ACL-005	Boundary test for maximum data display in panel	Teacher panel handles large amounts of student and schedule data smoothly	Pass

- **Performance Testing:** Given the performance constraint (generating reports within one minute for a dataset of 20 courses), outline a performance test plan and describe how you would use a testing tool to ensure compliance with this requirement.



3. Acceptance Criteria Management and Quality Assurance in JIRA (5 Marks – CO: CSC502.5)

- Explain how you would define, manage, and document acceptance criteria in JIRA based on the requirements and acceptance criteria provided.

The screenshot displays the Jira Code interface for a project named "Course_Scheduling". The left sidebar shows navigation options under "PLANNING" (Timeline, Board, List, Forms) and "DEVELOPMENT" (Code, Security, Releases). The main content area is titled "Code" and shows a message: "Nothing to show here, yet. You'll find work suggestions here when any pull requests need your attention." Below this, there is a section for "Linked pull requests" with a search bar and filters for Repository, Author, Reviewer, and Status. A table lists a pull request: "CS-7 Completed the part frontend and backend" with status "MERGED", no reviewers, and updated 2 days ago. The "Latest repositories" section shows a repository "Vivalchemy/Jira_CS" last updated 2 days ago.

Task recommendations for you based on linked pull requests that need your attention. [More about work suggestions](#)

Nothing to show here, yet
You'll find work suggestions here when any pull requests need your attention.

[Give feedback](#)

Linked pull requests

Your team's linked pull requests from the last 30 days. [Learn more](#)

Search pull requests

Repository Author Reviewer Status

Summary	Status	Reviewers	Comments	Updated
CS-7 Completed the part frontend and backend → main #1 • Repository: sAnyo08/Jira_CS	MERGED	No reviewers	0	2 days ago

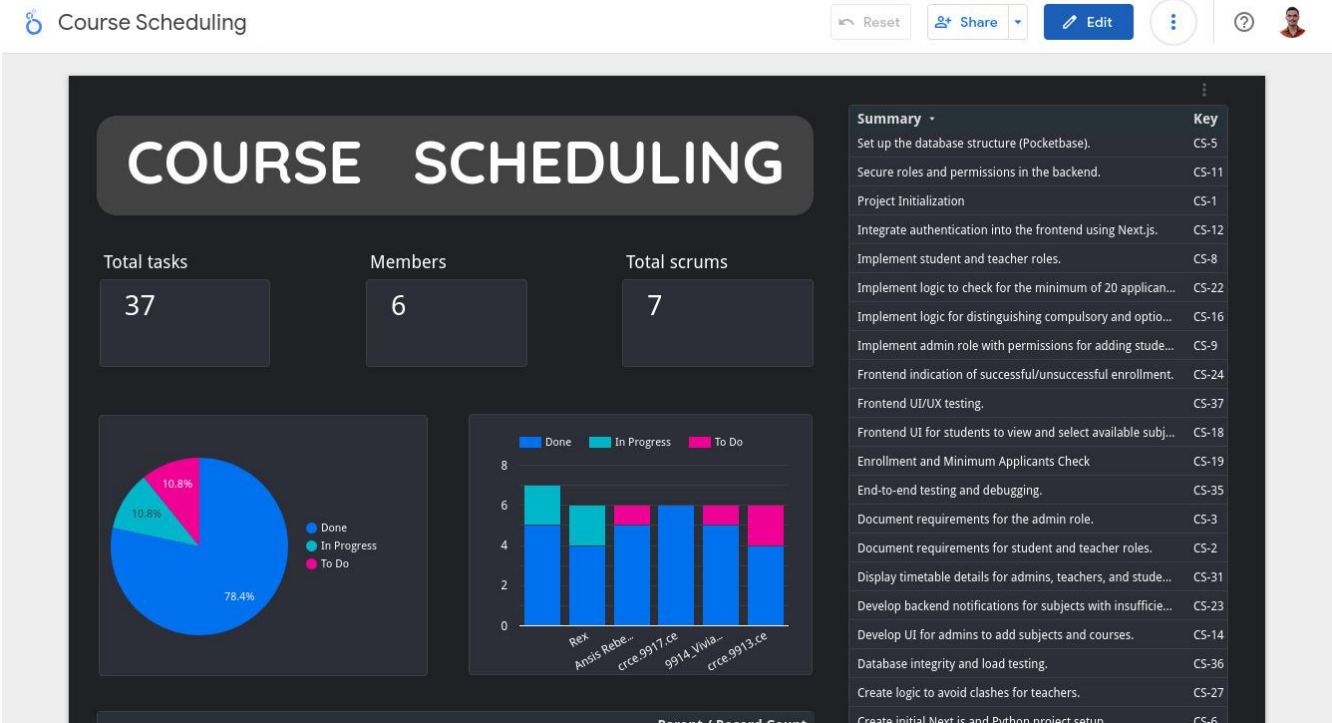
Latest repositories

The most recent repositories with commits, branches, or pull requests linked to this project.

Showing 1 of 1.

GitHub
Vivalchemy/Jira_CS
Last updated 2 days ago

- Describe your approach to using JIRA for quality assurance, focusing on tracking acceptance criteria and organizing test cases to ensure alignment with the system



requirements and project quality.

Rubrics :

COs Criteria		Needs		
Excellent (6/3)		Improvement (4/2)		
(10/5) Good (8/4) Satisfactory		Minimal (2/1)		
CSC502. 5	Management in JIRA	on		
	Clear, well organized structure, issues logged with detail	Adequate design, basic performance testing	Limited setup, lacks organization and detail	Lacks alignment and documentation
CSC502. 4	Well-defined test cases, comprehensive performance plan	Mostly aligned, adequate documentation	Minimal test case design, lacking key details	
CSC502. 5	Fully aligned with requirements, clear documentation	Basic structure, limited customization, basic issue tracking	Minimal alignment, lacks organization	
Project Setup and Defect Management in JIRA		Unclear	Basic design, some test cases missing	Issues
Test Case Design and Performance Testing	Mostly clear setup, some customization, issues logged	Partial alignment, missing some documentation	Minimal or missing test cases	
Acceptance Criteria				