

Test Automation and CI/CD for Web Application Project

11 Nov. 2024 Edit

Taeko Harada

Project Overview

- **Objective:** Develop a Todo List Application to experiment with CI/CD and test automation
- **Start Date:** 4 October 2024

About Application

- **Frontend:** Next.js (deployed on Vercel: <https://todo-fullstack-frontend-ten.vercel.app/>)
- **Backend:** Express.js (deployed on Heroku)
- **Database:** MySQL (integrated with the backend server)
- **Authentication:** JWT with HttpOnly cookies
- **Planned Addition:** Real-time communication (Socket.io)

GitHub URL

Frontend: <https://github.com/TaekoHarada/todo-fullstack-frontend.git>

Backend: <https://github.com/TaekoHarada/todo-fullstack-backend-mysql.git>

Docker Compose: <https://github.com/TaekoHarada/todo-fullstack-Docker.git>

App URL

<https://todo-fullstack-frontend-ten.vercel.app/>

Key Features

1. User authentication

2. ToDo management (add, update, delete, view)

Testing Strategy

- **Check List:** Determine the test objects
- **Test Coverage:** The test matrix is used to plan test cases and ensure that all features are covered.
- **Unit Testing:** Implemented with **Jest**
- **E2E Testing:** Implemented with **Cypress**
- **Environment:** Docker containers for consistent E2E testing across browsers.
- **Docker containers:** Three containers for the frontend, backend, and database, established using Docker Compose.

CI/CD Implementation

- **Platform:** GitHub Actions
- **Processes:** Automated build, unit test, and deployment (currently for the frontend only)

Current Status

- Core application functionality developed
- Test matrix, plans and test cases created
- Automation test code developed
- Docker compose file created
- CI/CD pipeline established by GitHub Actions
- Deploy on Heroku

Next Steps

1. Implementation of uncompleted test items identified from the checklist

- Session expiration
 - Form validation
 - Responsive support
 - Performance, security and accessibility under heavy load
 - Special characters, long text
 - Testing on different screen sizes and multiple browsers
2. Implement real-time communication features
 3. Enhance evaluation criteria by gaining a deeper understanding of Jest and Cypress.
 4. Optimize CI/CD pipeline by gaining a deeper understanding of GitHub Actions.
 5. Explore cloud deployment options, such as AWS

Conclusion

This project provided hands-on experience across the development lifecycle, from coding to testing and deployment. I learned to plan and implement tests while exploring tools like Jest and Cypress for comprehensive coverage. Setting up a CI/CD pipeline with GitHub Actions streamlined the workflow, and Docker ensured consistent testing. Overall, the project enhanced my understanding of test automation and CI/CD.

By creating a checklist, I was able to identify requirements that had not been tested.

Test Plan Materials

Check List

- Session expiration
- Form validation
- Responsive support
- Performance, security and accessibility under heavy load
- Special characters, long text
- Testing on different screen sizes and multiple browsers

Created a test matrix early on to cover all features comprehensively.

Test Matrix				
ID	Feature	Scenario	Expected Outcome	Priority
LOGIN-001	Login	Valid credentials	User is logged in and redirected to ToDo list page	High
LOGIN-002	Login	Invalid credentials	Error message displayed, login blocked	High
LOGIN-003	Login	Empty fields	User cannot log in, validation error displayed	Medium
TODO-001	To-do	Display ToDo list	Display existing todo list	High
TODO-002	To-do	Add new to-do	New to-do item is added to the list	High
TODO-003	To-do	Add with empty fields	User cannot add a new todo, validation error displayed	Medium
TODO-004	To-do	Delete a to-do	Item is removed from the list	High
TODO-005	To-do	Mark a to-do as completed	To-do is marked as completed	Medium
TODO-006	To-do	Mark a to-do as not-completed	To-do is marked as not-completed	Medium
LOGOUT-001	Logout	User clicks logout	User session is cleared, and login page is shown	High

Unit Test Plan

UnitTestPlan (Frontend) by Jest									
No	Function ID	Function	Scenario ID	Scenario	TestMatrix ID	Test Case	Expected Outcome		Result
01	LOGIN	User Login	000	Render login page	N/A	The login page has been rendered	All expected elements are rendered on the login page		Pass
			001	Successful login	LOGIN-001	User enters valid credentials	User is redirected to the home page.		Pass
			002	Invalid login	LOGIN-002	User enters invalid credentials	An error message is shown, and login fails.		Pass
			003	Empty input	LOGIN-003	User submits without input	An error message is shown asking for required fields.		Pass
02	VIEW_TASK	Viewing a List of Tasks	000	Render ToDo page	TODO-001	ToDo page has been rendered	All expected elements are rendered on the todo page		Pass
			001	View all tasks	TODO-001	ToDo page has been rendered and user views the list of all tasks	All expected elements are rendered on the todo page The list of tasks is displayed correctly		Pass
			002	View tasks when any tasks have not been added	TODO-001	User views tasks when no tasks have been added	A message indicating no tasks are available is displayed		Pass
			003	Redirect to login page when the user is not authorized	LOGIN-002	User have not login, or the authentication has expired. The ToDo page has been denied and user is navigated to login page.	Show console error and the login page has been reloaded.		Pass
03	ADD_TASK	Adding Tasks	000	Add a new task	TODO-002	User adds a new task with a valid title	The new task is added successfully		Pass
			001	Add a new task with a empty title	TODO-003	User tries to add a new task with an empty title	An error message is shown indicating the title is required		Pass
									Pass
04	DEL_TASK	Deleting Tasks	000	Delete an existing task	TODO-004	User deletes an existing task	The task is deleted successfully		Pass
			001	Delete a non-existent task	TODO-004	User tries to delete a non-existent task	An error message is shown indicating the task does not exist		Pass
									Pass
05	COMPLETE_TASK	Marking Tasks as Complete	000	Mark a not-completed task	TODO-005	User marks a task that is not yet completed	The task is marked as complete		Pass
			001	Mark a completed task	TODO-006	User tries to mark a task that is already completed	An error message is shown indicating the task is already completed		Pass

E2E Test Plan

E2E Test Plan			
Test Case	Test Matrix ID	Scenario	Result
Case 1	LOGIN-001	Login (Success)	Pass
	TODO-001	Display the existing todos	Pass
	TODO-002	Add a new todo	Pass
	TODO-005	Edit todo as completed	Pass
	TODO-004	Delete todo	Pass
	LOGOUT-001	Logout	Pass
Case 2	LOGIN-001	Login (Success)	Pass
	TODO-001	Display the existing todos	Pass
	TODO-003	Add with empty fields	Pass
Case 3	LOGIN-001	Login (Success)	Pass
	TODO-001	Display the existing todos	Pass
	TODO-002	Add a new todo	Pass
	TODO-005	Edit todo as completed	Pass
	TODO-006	Edit todo as not-completed	Pass
Case 4	LOGIN-002	Login with invalid password (Fail)	Pass
Case 5	LOGIN-003	Login with empty Id and Password (Fail)	Pass