****

Academic year 2025-2026

**20CYS401**

**Secure Software Engineering**

Lab Record

*Submitted by*

P Thanusree

CH.EN.U4CYS22038

**Amrita Vishwa Vidyapeetham Chennai – 601 103, Tamil Nadu, India.**

**October 2025**

**Question:For the given application complete full secure devops lifecycle within single lab session**

**Application name: Feeddback submission portal**

**Suggested technology stack:Node.js+mongoDB**

**Focus area/expected output:cross site scripting prevention**

**1.Mini Software Requirements Specification (SRS)**

**1. Purpose**

The purpose of the Feedback Submission Portal is to provide a simple, secure, and reliable web-based application for users to submit feedback and for all visitors to view previously submitted comments. The primary focus of this project is to implement this functionality while ensuring the application is secure against common web vulnerabilities, specifically Stored Cross-Site Scripting (XSS).

**2. Functional Requirements**

The system shall:

* **FR1: Present Feedback Form:** Display a web page containing a form with input fields for a user's "Name" and "Feedback" text.
* **FR2: Allow Feedback Submission:** Allow a user to submit the completed form to the server.
* **FR3: Persist Feedback Data:** Store the submitted "Name" and "Feedback" text securely in a MongoDB database.
* **FR4: Display All Feedback:** Display a list of all previously submitted feedback entries (name and text) on the main web page for all users to view.
* **FR5: Validate User Input:** The server shall validate that both the "Name" and "Feedback" fields are not empty before processing a submission.

**3. Non-Functional Requirements**

* **NFR1: Security:** The application must be secure against Stored Cross-Site Scripting (XSS) attacks. All user-submitted content that is displayed on the page must be appropriately handled to prevent the execution of embedded scripts in the browsers of other users.
* **NFR2: Usability:** The user interface must be simple, clean, and intuitive, requiring no training for a user to submit feedback.
* **NFR3: Performance:** The main page, including the list of all submitted feedback, should load within 5 seconds under normal conditions.
* **NFR4: Technology Stack:** The application must be developed using the specified technology stack.

**4. Constraints**

* **C1: Backend Technology:** The server-side application logic must be implemented using Node.js with the Express.js framework.
* **C2: Database Technology:** The data must be stored in a MongoDB database.
* **C3: Timeframe:** The entire secure development lifecycle, from planning to deployment, must be completed within a single lab session.
* **C4: Deployment Environment:** The final containerized application must be deployed to a local Minikube Kubernetes cluster.

**5. Acceptance Criteria**

The project will be considered successful when the following criteria are met:

* **AC1: Successful Functional Loop:** A user can navigate to the web page, enter a name and feedback, click "Submit," and see their feedback appear correctly in the list on the same page after it reloads.
* **AC2: Successful XSS Prevention:** When a user submits feedback containing the string <script>alert('XSS')</script>, the script is **not** executed in the browser upon page load. The literal text of the script is either displayed safely on the page or the submission is blocked entirely with an alert.
* **AC3: Successful Negative Input Handling:** When a user attempts to submit the form with the "Feedback" field left empty, the server rejects the submission and presents a clear error message (e.g., "Name and feedback text cannot be empty."). The empty submission is not saved to the database.