****

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**

**8.Code Refactoring**

Identify **one security improvement** (e.g., input validation, hashing passwords) and show code before & after.

Input validation is one security improvement:

**Before Code Snippet (No Input Validation)**

This is the original app.post('/submit', ...) route handler. Notice that it takes the name and feedbackText directly from the user's request and, after a basic check to see if they are empty, saves them to the database without inspecting the content for malicious code.

// BEFORE REFACTORING

// Route to handle form submission

app.post('/submit', async (req, res) => {

const { name, feedbackText } = req.body;

if (!name || !feedbackText) {

return res.status(400).send('Name and feedback text cannot be empty.');

}

// VULNERABILITY: User input is saved directly to the database

// without checking for malicious content like script tags.

const newFeedback = new Feedback({ name, feedbackText });

try {

await newFeedback.save();

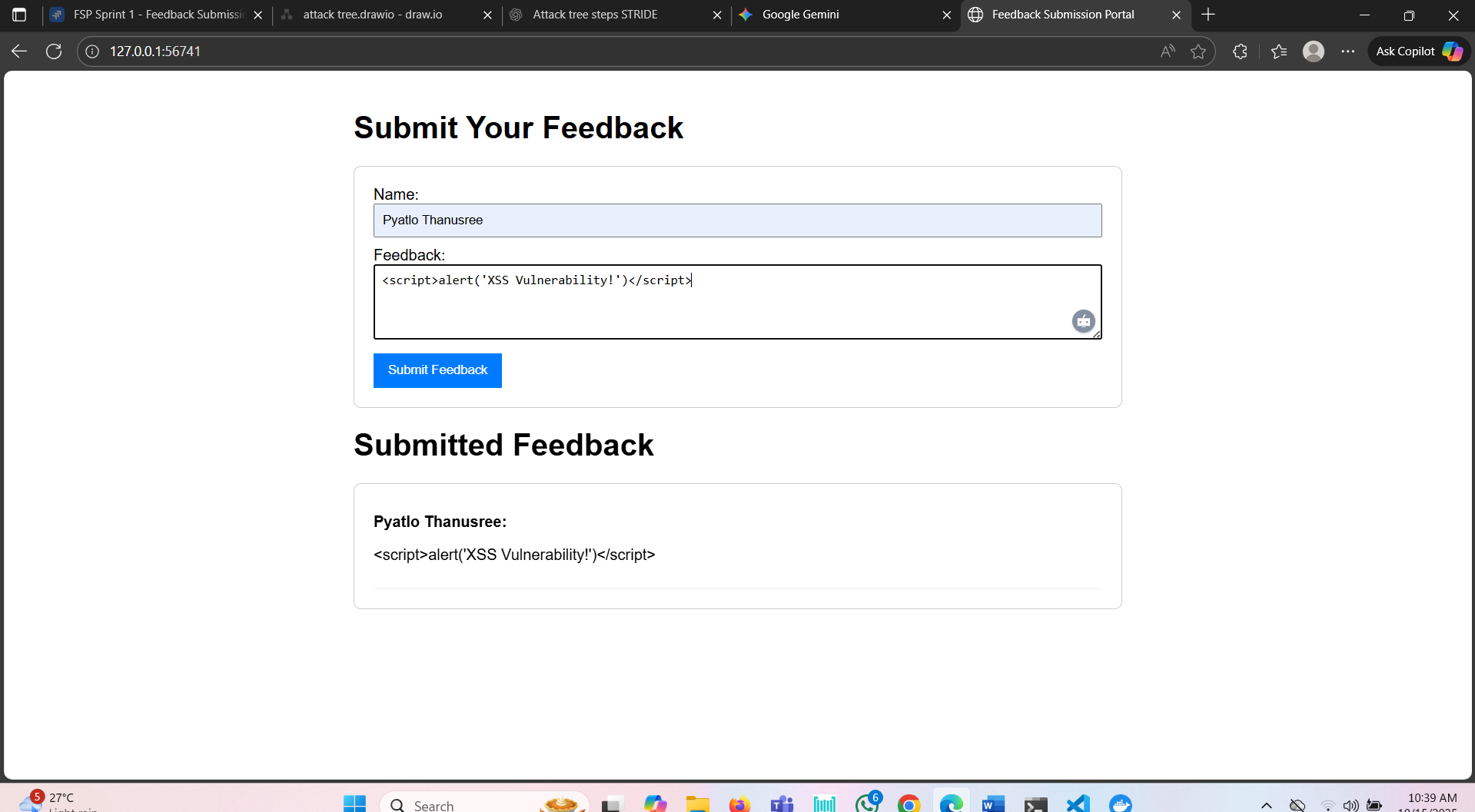
res.redirect('/');

} catch (err) {

res.status(500).send('Error saving feedback.');

}

});



**After Code Snippet (With Input Validation)**

This is the new, refactored app.post('/submit', ...) route handler. It now includes an if/else block that acts as a security gate. It inspects the feedbackText for the presence of <script>. If it's found, the submission is blocked, and a response is sent to trigger an alert. Otherwise, the data is saved as normal.

// AFTER REFACTORING

// MODIFIED Route to handle form submission with Input Validation

app.post('/submit', async (req, res) => {

const { name, feedbackText } = req.body;

if (!name || !feedbackText) {

return res.status(400).send('Name and feedback text cannot be empty.');

}

// --- SECURITY IMPROVEMENT: INPUT VALIDATION ADDED ---

// Actively check if the feedback text contains a script tag.

if (feedbackText.toLowerCase().includes('<script>')) {

// BLOCK the submission and send a response that triggers an alert.

res.status(400).send(`

<html>

<body>

<script>

alert("Submission blocked: Malicious script detected!");

window.location.href = "/";

</script>

</body>

</html>

`);

} else {

// If the input is clean, proceed to save it.

const newFeedback = new Feedback({ name, feedbackText });

try {

await newFeedback.save();

res.redirect('/');

} catch (err) {

res.status(500).send('Error saving feedback.');

}

}

});

