Air University

(Final-Term Examination: Fall 2025)

Subject:	**Secure Software Design and Development Lab**	Total Marks:	100
Course Code:	**CY-256L**	Date:	
Class:	**BS-CYS(EVE)**	Time:	
Semester:	**IV**	Duration:	3 **Hours**
Section:	**B**	FM Name:	Fizza Nasir
HoD Signatures:	_____	FM Signatures:	_____

Note:

- **Add a sticky note with your name and roll number on each screenshot/snippet.**
- **Submit your work in a Word document (.doc/.docx).**
- **Attempt all questions.**
- **Add a smiley at the end to earn 2 bonus marks.**
- **Stay calm — you've got this!**

	Q. No. 1 (CLO 1)	**30 Marks**
a	**Static Code Analysis using Codacy** **Scenario:** You have joined the **software quality assurance team** for an **eCommerce Website project** developed using modern web technologies. The platform includes both **frontend and backend components**, with features such as user authentication, product listing, shopping cart, and order management. Your team lead has expressed concerns about the growing **technical debt**, inconsistent coding standards, and the presence of **untracked bugs and vulnerabilities** in the codebase. As part of the **secure development workflow**, you are required to conduct a **Static Code Analysis** using **Codacy** — a cloud-based tool that automates code reviews and highlights potential issues. A **zipped folder containing the project source code** has been provided to you. Your task is to follow the steps below: **Push the Project to GitHub** Unzip the folder and push the project to a new GitHub repository. Use the following commands in your terminal or Git Bash (adjust URLs and paths as needed): git init git add . git commit -m "Initial commit of Ecommerce project" git branch -M main git remote add origin https://github.com/your-username/ ecommerce-static-analysis.git git push -u origin main After pushing the project, log into Codacy at https://app.codacy.com, authorize your GitHub account, and add the newly created repository for analysis.	10

b	**Codacy Interface & Tabs Explanation** Once Codacy completes scanning your project, explore the following six tabs available in the interface: • Overview • Issues • Code Patterns • Coverage • Commits • Pull Requests For **each tab**, provide: • A brief explanation of what it shows • How the information helps in identifying code flaws, security risks, or poor practices	10
c	**Identify and Address Vulnerabilities** Codacy will highlight multiple issues and recommendations. Choose any **three** code issues or vulnerabilities and describe: • The type of issue (e.g., security, code duplication, complexity, maintainability) • The location/module in which it appears (e.g., messaging system, skill progress tracker, authentication) • A clear solution or fix, with reasoning behind your recommendation	10
	Q. No. 2 (CLO 2)	**30 Marks**
a	**Dynamic Code Analysis using OWASP ZAP** **Scenario:** You are a cybersecurity intern assigned to perform a **vulnerability assessment** of three business-critical web applications that are part of your organization's partner ecosystem. These applications are publicly accessible and handle sensitive user data. Your task is to conduct analysis using **OWASP ZAP (Zed Attack Proxy)** to identify security flaws. These applications are intentionally vulnerable and designed for educational security testing purposes. Provided URLs: • https://owasp.org/www-project-webgoat/ • https://google-gruyere.appspot.com/ • http://testasp.vulnweb.com You are required to perform Dynamic Application Security Testing (DAST) on **any one** of the three websites listed above using OWASP ZAP. **ZAP Interface Overview** Before conducting the scan, explore and explain the main components of the OWASP ZAP interface. Your answer must include the following sections:	15

	• Sites Panel • Request/Response Viewer • History Tab • Alerts Tab • Spider • Active Scan For each, briefly explain: • What the tab or section is used for • How it contributes to the vulnerability assessment process	
b	**Vulnerability Scanning Using ZAP** 1. Choose any website from the list and perform a basic vulnerability scan using ZAP's Spider and Active Scan features. 2. For each website: a. Run the scan and observe the output in the Alerts tab. b. Identify and explain any three vulnerabilities or issues discovered. For each: c. Mention the type of vulnerability (e.g., XSS, Insecure Cookies, Information Disclosure) 3. Provide a brief explanation of how the vulnerability works 4. Suggest a possible remediation or fix 5. Optionally (for better marks), refer to specific HTTP requests/responses where the vulnerability was detected using the Request/Response Viewer.	15
	Q. No. 3 (CLO 2)	**30 Marks**
a	**Secure Form Enhancement Visual Studio Code** **Scenario:** You are provided with a simple healthcare contact website called **QuickCare Contact Portal**. The site includes a form where users can enter their personal information and message. However, the form currently lacks **basic security practices**, making it vulnerable to **unfiltered input, auto-fill abuse**, and **scripting risks**. Your task is to identify and fix the following security issues in the code and enhance the user experience and functionality of the contact form using Visual Studio Code. **Update index.html – Add Validation and Placeholders** • Add appropriate placeholders for all input fields. • Use the correct input types (e.g., email, date) for better validation. • Add the required attribute to make inputs mandatory. • Add autocomplete="off" to fields like email and date of birth.	10

b	**Update style.css – Apply Styling Enhancements**
	In the style.css file, enhance the existing styles to make the form more visually appealing and user-friendly. Your tasks:
	• Add a light background color to the page (e.g., #f5f5f5) and a white container for the form.
	• Apply a box-shadow to the form container to make it stand out.
	• Style the Submit button with:
	  ○ A custom background color
	  ○ Rounded corners
	  ○ A hover effect (e.g., change background on hover)
	• Increase spacing between fields and labels for better readability.
		10

c	**Update script.js – Sanitize Inputs and Display Feedback** In script.js, implement **client-side validation** to: • Prevent submission if any field contains **script tags** or suspicious input (e.g., <, >). • Show an error message if invalid content is found. • Allow submission only if inputs are clean. **Note: Do not add or remove files. Make changes only in the provided files. Ensure your code runs correctly by previewing the form in your browser.**	10
	Q. No. 4 (CLO 2)	**10 Marks**
a	Draw diagram to explain the difference between the Docker and Virtaulization	10

********** End of Question Paper **********