Assignment: Build and Analyze a Vulnerable Web Application

Objective

Design and implement a **simple, minimal web application** that contains **at least two vulnerabilities** covered in lab sessions (e.g., SQL Injection, XSS, Command Injection, File Inclusion). The focus is on the **backend logic**. The frontend can be basic (e.g., raw HTML). The app must be **containerized using Docker** to allow consistent building and testing across different systems.

Requirements

Web Application

- Create a basic web app (PHP, Flask, Node.js, etc.).
- Include **at least two vulnerabilities**.
- Use **Docker** for deployment.

Report Contents

Prepare a report that includes the following:

1. **Web Application Features**
   Describe the purpose of your app. What features does it offer?
2. **Identifying Vulnerabilities**
   Which vulnerabilities are present? Where are they located in the code?
3. **Black-Box Testing**
   If you do **not have access to the source code**, how would you identify potential vulnerabilities?
   - What tools or techniques would you use?
   - What signs or responses from the application help reveal vulnerabilities?
4. **How to Exploit**
   Provide step-by-step exploitation instructions for each vulnerability. Include:
   - Payloads

- Screenshots or video proofs

5. **Root Cause**
   Why does each vulnerability exist? What is the technical mistake behind it?

6. **Proposed Mitigations**
   What steps can be taken to fix or prevent each vulnerability?

Deliverables

- **Your GitHub repository link**, which must contain:
  - Source code of the web application.
  - Dockerfile for building and running the application.
  - Report in PDF format, including screenshots.
  - (Optional) A video demonstrating proof-of-concept (PoC) exploitation.

Rubric (Total: 100 points)

Criteria	Description	Points
Web Application (40 pts)		
–	At least 2 vulnerabilities implemented	20
–	Functional and working app	10
–	Proper use of Docker	10
Report Quality (60 pts)		
–	Clear feature explanation	10
–	Vulnerabilities explained	10
–	Black-box testing insights	5
–	Exploitation payloads and procedure	10
–	Screenshots or video included	10
–	Root cause analysis	10
–	Proposed mitigations	10
Bonus (Optional)		
–	One extra vulnerability	+5
–	Mitigation implemented in code	+5