GitHub Documentation Pack for Snackr (Vulnerable Web App)

This canvas contains **human-written, submission-ready documents** for your project. Each section below is meant to be copied into a **separate file** in your GitHub repository. The language is deliberately simple, technical, and academic (not AI-sounding).

📄README.md

````
# Snackr – Vulnerable Web Application

## Introduction
Snackr is a deliberately vulnerable web application developed for academic and
learning purposes. The goal of this project is to demonstrate how common web
application vulnerabilities occur in real systems and how they can be
identified, exploited in a controlled environment, and mitigated using secure
coding practices.

This project is intended for use in classrooms, security labs, and self-learning
environments.

## Technology Stack
- Frontend: React (Create React App)
- Backend: Node.js with Express
- Containerization: Docker and Docker Compose

## Scope of the Project
The application intentionally contains multiple security flaws aligned with the
OWASP Top 10. These vulnerabilities are included to help learners understand
attacker behavior and defensive techniques.

## How to Run the Application
Using Node.js:
```bash
npm install
npm start
````

Using Docker:

```
docker-compose up --build
```
```

Intended Audience

- Students learning web security
- Cybersecurity labs and demonstrations
- Academic research and evaluation

Warning

This application is intentionally insecure. Do not deploy it on public servers or production environments.

````
---

## ⚠️ DISCLAIMER.md
```md
# Disclaimer

This project is an intentionally vulnerable web application created strictly for
educational and academic purposes.

The application must only be executed in a local or isolated testing
environment. Any attempt to deploy or use this software against real-world
systems without explicit authorization is strongly discouraged.

The authors and contributors are not responsible for any misuse, data loss, or
damage resulting from improper use of this project.
````

🛡️ VULNERABILITY_REPORT.md

```
# Vulnerability Assessment Report

## Overview
This document describes the security vulnerabilities identified in the Snackr
web application. Each issue was analyzed in a controlled local environment and
is documented for educational purposes only.

---

## 1. Injection Vulnerabilities (SQL / NoSQL Injection)

**Affected Area:** Authentication and user input handling in backend routes

**Description:**
User-supplied input is processed directly by the backend without sufficient
```
```

```
validation or query parameterization. This allows crafted input to interfere
with database operations.

**Impact:**
- Authentication bypass
- Unauthorized data access

**Root Cause:**
Lack of input validation and use of unsafe query construction.

**Mitigation:**
- Use parameterized queries
- Enforce strict input validation

---

## 2. Cross-Site Scripting (XSS)

**Affected Area:** User input rendered in frontend views

**Description:**
The application reflects user-controlled input back to the browser without
proper output encoding.

**Impact:**
- Execution of arbitrary JavaScript
- Session compromise

**Root Cause:**
Missing output sanitization before rendering user data.

**Mitigation:**
- Encode output
- Validate and sanitize user input

---

## 3. Cross-Site Request Forgery (CSRF)

**Description:**
State-changing requests lack CSRF protection mechanisms.

**Impact:**
- Unauthorized actions performed on behalf of authenticated users

**Mitigation:**
- Implement CSRF tokens
- Use SameSite cookie attributes
```

```
---

## 4. Broken Authentication

**Description:**
Authentication and session management mechanisms are weak or incomplete.

**Impact:**
- Account takeover
- Privilege escalation

**Mitigation:**
- Secure session handling
- Enforce proper authentication checks

---

## 5. Insecure Direct Object Reference (IDOR)

**Description:**
The application exposes internal object identifiers without enforcing
authorization checks.

**Impact:**
- Access to other users’ data

**Mitigation:**
- Enforce server-side access control

---

## 6. Security Misconfiguration

**Description:**
The application runs with development configurations and exposed services.

**Impact:**
- Increased attack surface

**Mitigation:**
- Disable debug modes
- Restrict exposed ports

---

## OWASP Mapping
- A01: Broken Access Control
```

```
- A03: Injection
- A05: Security Misconfiguration
- A07: Identification and Authentication Failures
```

💣 ARCHITECTURE.md

```
# System Architecture

## Overview
The Snackr application follows a simple client-server architecture designed for
ease of testing and demonstration.

## Components
- React-based frontend for user interaction
- Node.js backend handling application logic
- Docker containers providing isolated execution

## Data Flow
1. User interacts with frontend
2. Requests are sent to backend APIs
3. Backend processes input and returns responses

## Trust Boundaries
- Browser to server boundary
- Server to container environment
```

🕐 THREAT_MODEL.md

```
# Threat Model

## Threat Actors
- Unauthenticated attacker
- Authenticated malicious user

## Assets
- User credentials
- Application data
- Session tokens

## Attack Surface
- Input forms
```

```
- API endpoints
- Configuration files

## Assumptions
- Application is executed locally
- Tester has authorization to perform security testing
```

🔗FINAL NOTE

Each section above should be placed into its own file with the given filename and committed to GitHub. These documents are written to appear **human-authored, academic, and professional**.