🔐 SQL Injection Playground with Detection Engine

Name: Mohd Uzair

Internship Program: ElevateLabs

Date: September 1, 2025

1. Project Objective

This project demonstrates SQL Injection vulnerabilities, detection, and prevention through a hands-on educational platform. It includes a vulnerable web app, a detection engine, and a secure coding example.

2. Tools & Technologies

- **Backend:** Flask (Python)
- **Database:** SQLite
- **Frontend:** HTML/CSS
- **Detection:** Python script with request analysis
- **Logging:** File-based logs

3. Vulnerable Application

The app has two vulnerable endpoints:

- **Login Page:** Uses string formatting → allows authentication bypass
- **Search Page:** Direct input in SQL → allows data leakage

4. SQL Injection Examples

Login Bypass: Username: `admin'--` , Password: anything

Data Leak: Search: `' UNION SELECT sql, '' FROM sqlite_master--`

5. Detection Engine

The `detector.py` script sends payloads and analyzes:

- Response content (success/error)
- Response time (delay-based detection)
- Database errors

Logs are saved in `logs/sqli_logs.txt` .

6. Prevention: Secure Coding

The secure login uses parameterized queries:

```
cur.execute("SELECT * FROM users WHERE username=? AND password=?", (username, passwor
```

This prevents SQL injection by separating code from data.

7. Educational Value

This project helps learners understand:

- How SQLi works
- How to detect attacks
- How to prevent them using best practices

8. GitHub Repository

🔗 https://github.com/uzairkaaf/sqli-playground

The repo contains all source code, instructions, and this report.

9. Conclusion

This project successfully demonstrates real-world SQL injection risks and defenses. It highlights the importance of secure coding and proactive security testing.

10. How to Run

```
# Install dependencies
pip install -r requirements.txt

# Start the app
python app.py
# Visit http://localhost:5000

# Run detector
python detector.py
```

Internship Project | Cybersecurity | Generated using HTML → PDF