SQL Injection Project on Auth Bypass & Credential Exfiltration

Ethics & Scope: This project documents testing performed **only** against an intentionally vulnerable training instance provided for ethical practice. Do **not** use these techniques on systems you do not own or lack explicit written permission to test.

Summary

- **Target**: Intentionally vulnerable login endpoint (training instance).
- **Goal**: Demonstrate SQL injection leading to **authentication bypass** and **exfiltration of stored credentials**.
- **Methods**:
  1. **Manual** exploitation via crafted payloads and wordlists.
  2. **Automated** exploitation using `sqlmap`.
- **Tooling**: Burp Suite (Proxy/Repeater/Intruder), `sqlmap`, Kali Linux payload wordlists (e.g., `SQL.txt`), browser.
- **Outcome**: Verified SQLi at login, bypassed auth, enumerated DB structure, and dumped user credential records (sanitized in report).

Skills Demonstrated

- Web app recon & traffic interception (Burp Proxy)
- Input tampering & payload testing (error-based, boolean-based, union-based)
- Automating detection/exploitation with `sqlmap`

High-Level Workflow

1. **Proxy setup** → route browser through Burp; capture baseline login request.
2. **Manual SQLi testing** → inject payloads in `username`/`password`; evaluate responses.
3. **Automated verification** → run `sqlmap` against the same request to confirm and enumerate.
4. **Evidence** → save key HTTP requests/responses and sanitized DB dumps.

5. **Reporting**

Repo Structure

```
.
├─ README.md
├─ report/
│ ├─ SQLi_Project_Report.pdf # exported report (or .md)
│ └─ evidence/
│ ├─ requests/
│ │ ├─ baseline_login.txt
│ │ └─ sqli_login_payloads.txt
│ ├─ screenshots/
│ │ ├─ 01_login_page.png
│ │ ├─ 02_burp_repeater.png
│ │ ├─ 03_auth_bypass.png
│ │ ├─ 04_sqlmap_detection.png
│ │ ├─ 05_sqlmap_dump.png
│ │ └─ 06_db_overview.png
│ └─ dumps/
│ ├─ dbs.txt
│ ├─ tables.txt
│ └─ users_sanitized.csv
└─ legal/
└─ authorization.md
```

Tools Used

- **Burp Suite** (Community edition): Proxy, Repeater, Intruder
- **sqlmap**: Automated SQLi detection/exploitation
- **Kali Linux**: Wordlists (`wfuzz`/`payloads/SQL.txt`), terminal utilities

Legal & Responsible Disclosure

All data shown in the report is **redacted/sanitized**. Follow your organization’s policy and relevant laws. Use parameterized queries, strict input validation, least privilege DB accounts, and WAF/monitoring.