Title: -Secure-Ride-Hailing Webiste

Welcome to the Secure-Ride-Hailing final project presentation for CY321. Meet our team and discover our security achievements.

Group Members:

Saad Ali 2022512

Arsalan Khan 2022115

Executive Summary

Our project delivers a user-friendly ride-hailing website with core features: ride booking, user authentication, and ride history. Payment handling is outsourced to reduce complexity and risk. Emphasis is placed on secure development, including encryption and protection against common threats like SQL injection and XSS.

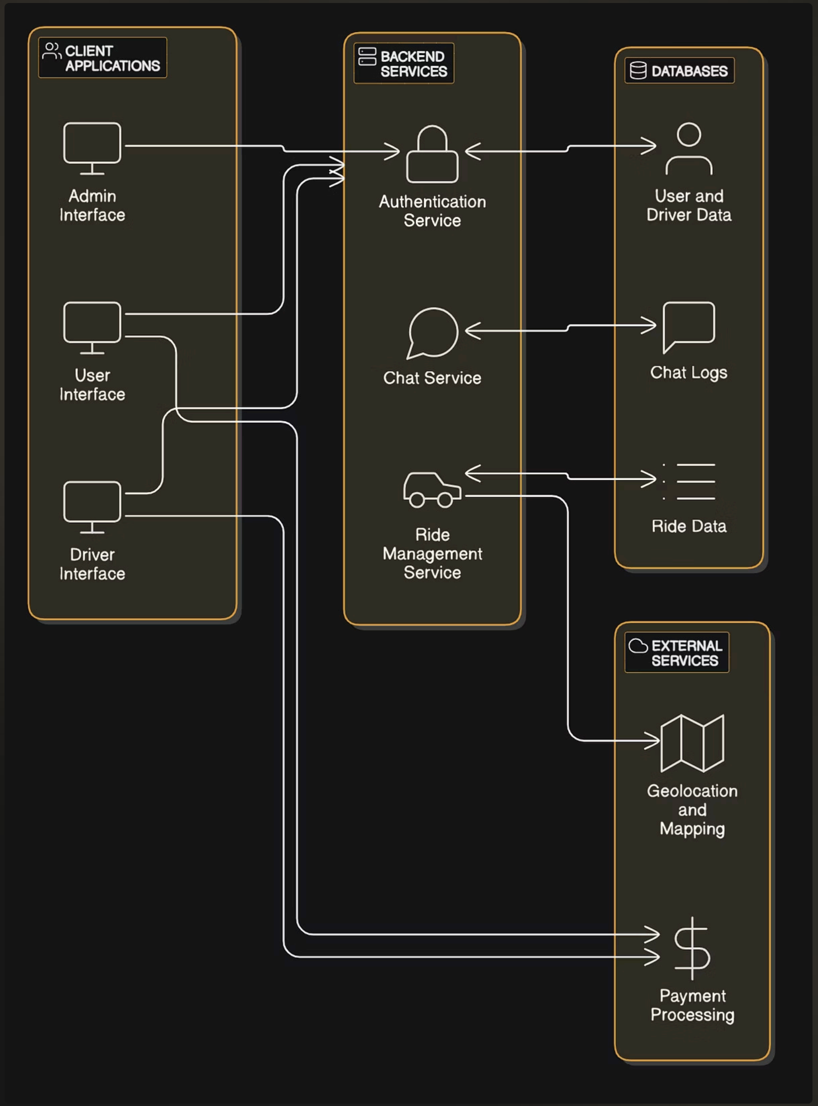
Architecture Overview: System Diagram

The app uses HTTPS between user and backend. JWT secures database access.

External payment API is isolated for

safety.

1. User interacts via secure HTTPS
2. Flask app manages sessions and logic
3. SQLite stores data.
4. Payment via secure external gateway API



Architecture Overview: Security Layers

1 Perimeter

Rate limiting controls request frequency

2 Application

- Secure error handling
- Input sanitization for location

3 Data

Encryption applied to data at rest for confidentiality

Threat Model: IRUS Risk Components

Interfaces

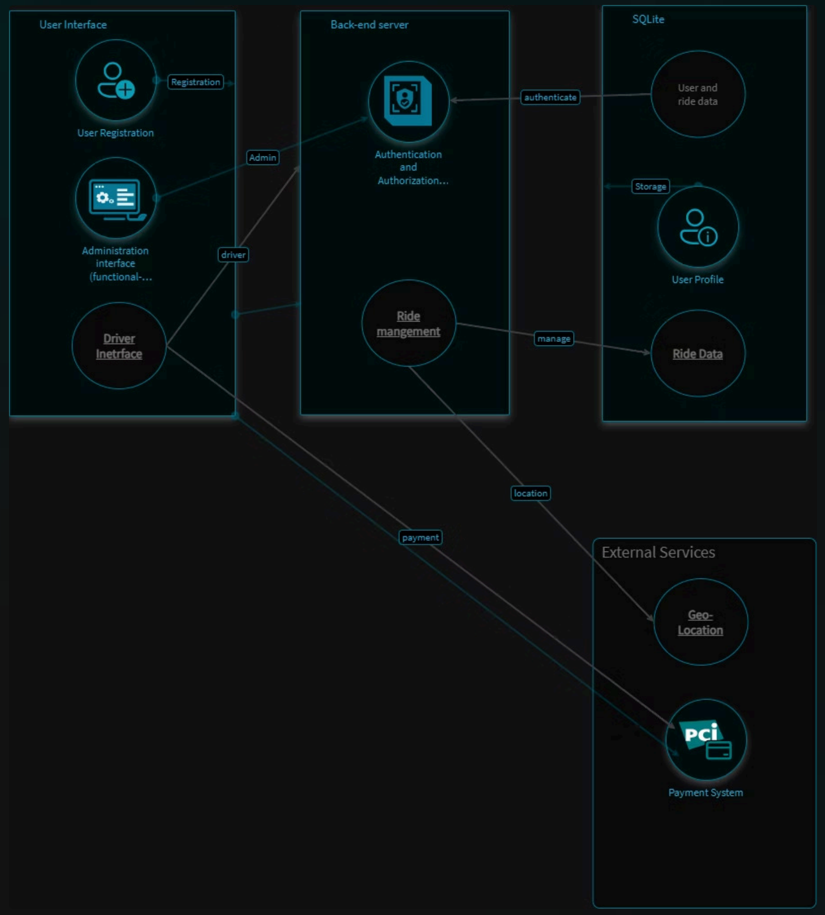
- Web
- Mobile API

Users

- Riders
- Drivers

Services

- Authentication
- Payments



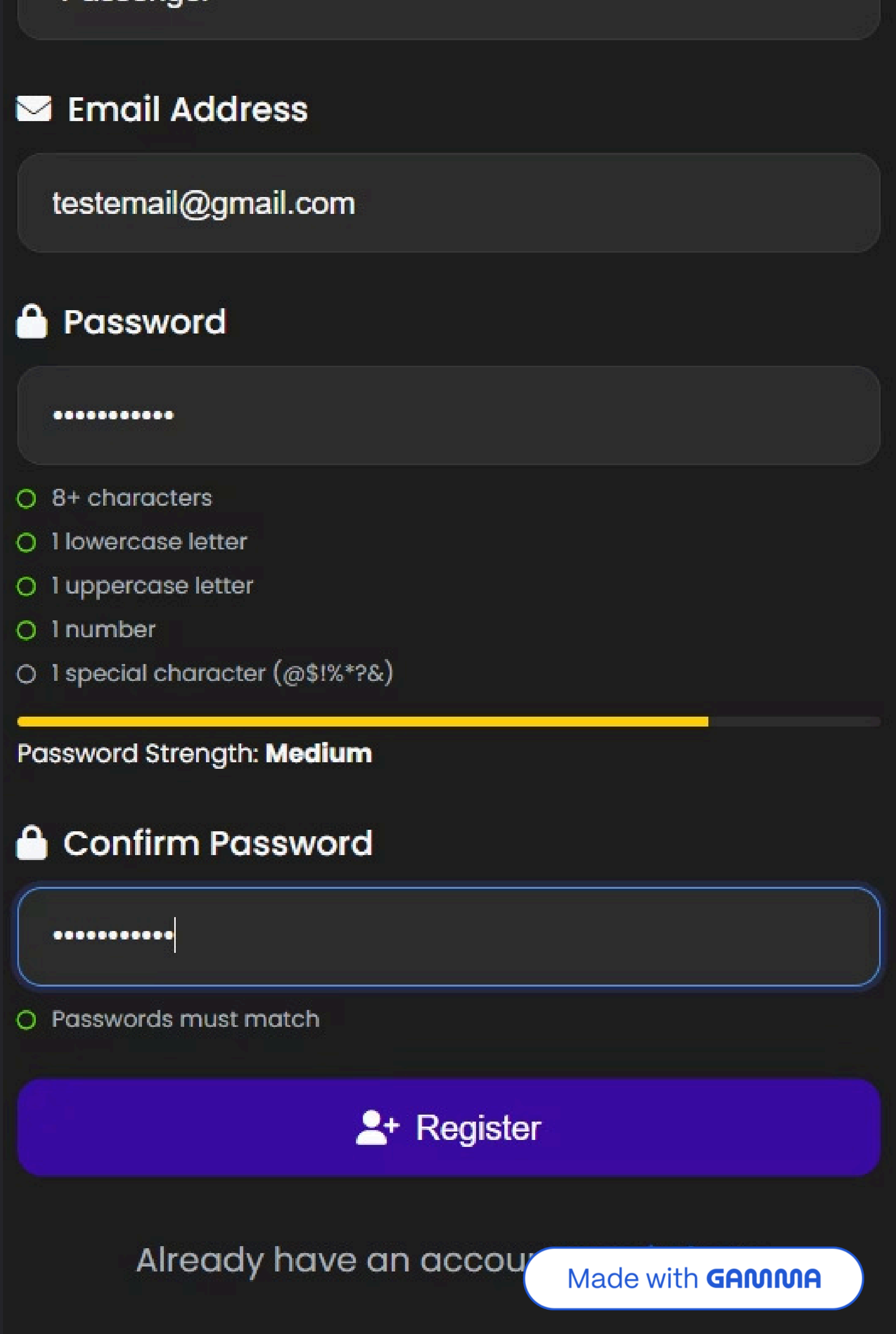
Implemented Security Features Overview

Core Protections

- Password complexity with special characters
- Session timeout after 30 minutes inactivity
- Rate limiting: 5 login attempts per minute

Technical Controls

- Outsourced payments



Email Address

testemail@gmail.com

Password

••••••••••••

○ 8+ characters
○ 1 lowercase letter
○ 1 uppercase letter
○ 1 number
○ 1 special character (@$!%*?&)

Password Strength: **Medium**

Confirm Password

••••••••••••

○ Passwords must match

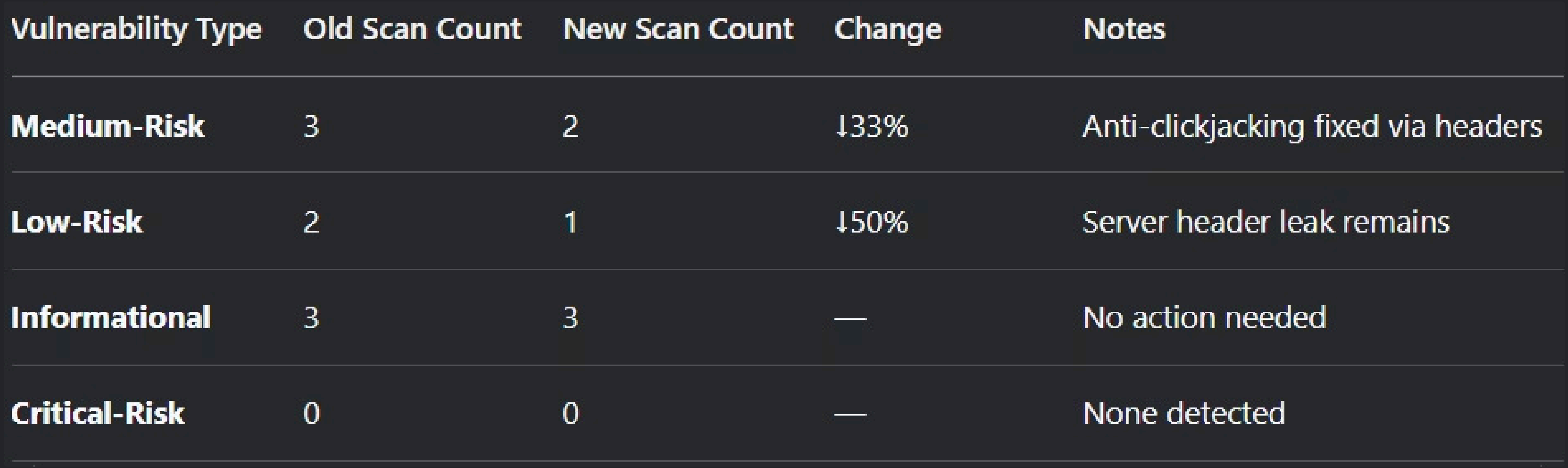
Already have an accou

Security Features

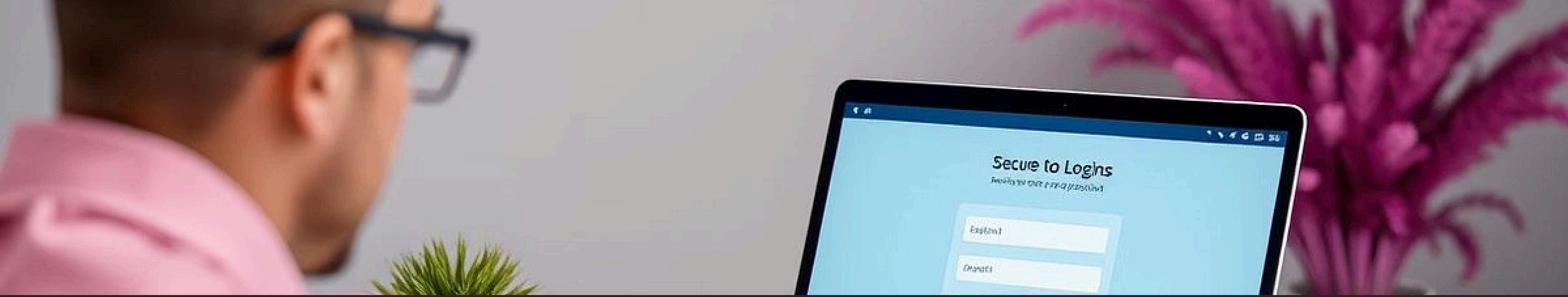
Authentication Security	Data Security	Application Security
• Password hashing using Werkzeug's PBKDF2-SHA256 • Session-based authentication with secret key	• Input validation for ride bookings • Secure password storage	• Secure error handling • Input sanitization for locations

Testing Results: ZAP Scan & Coverage

ZAP Scan Comparison



Vulnerability Type	Old Scan Count	New Scan Count	Change	Notes
Medium-Risk	3	2	↓33%	Anti-clickjacking fixed via headers
Low-Risk	2	1	↓50%	Server header leak remains
Informational	3	3	—	No action needed
Critical-Risk	0	0	—	None detected



Live Demo Script Overview

The End