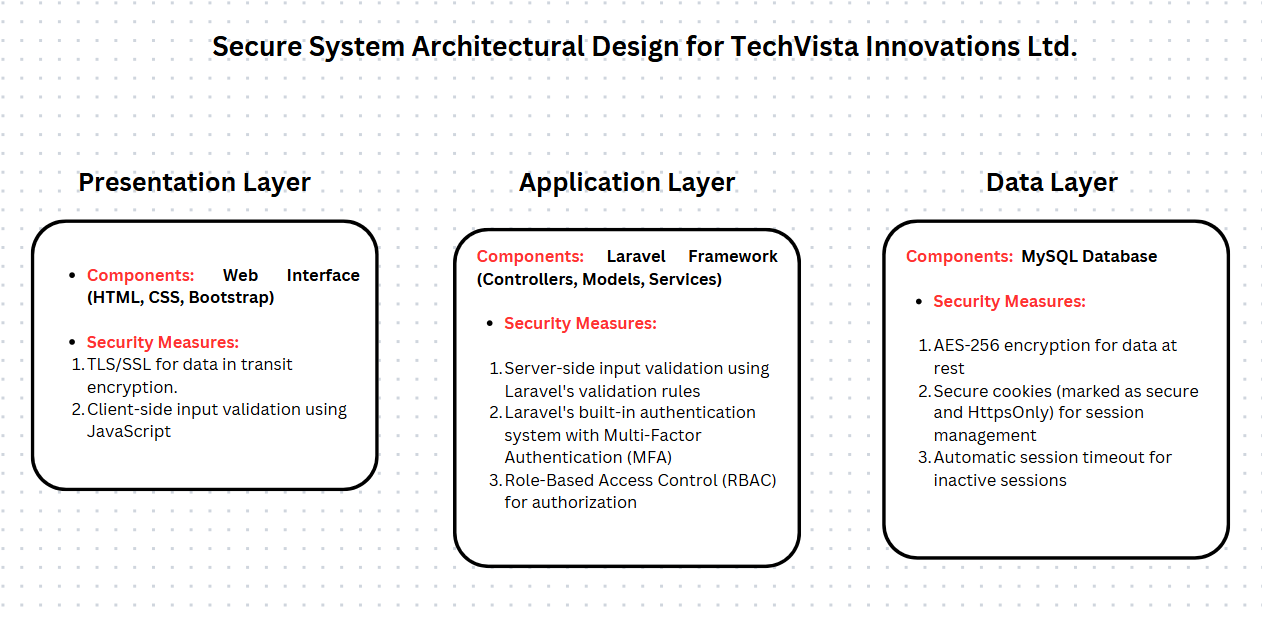
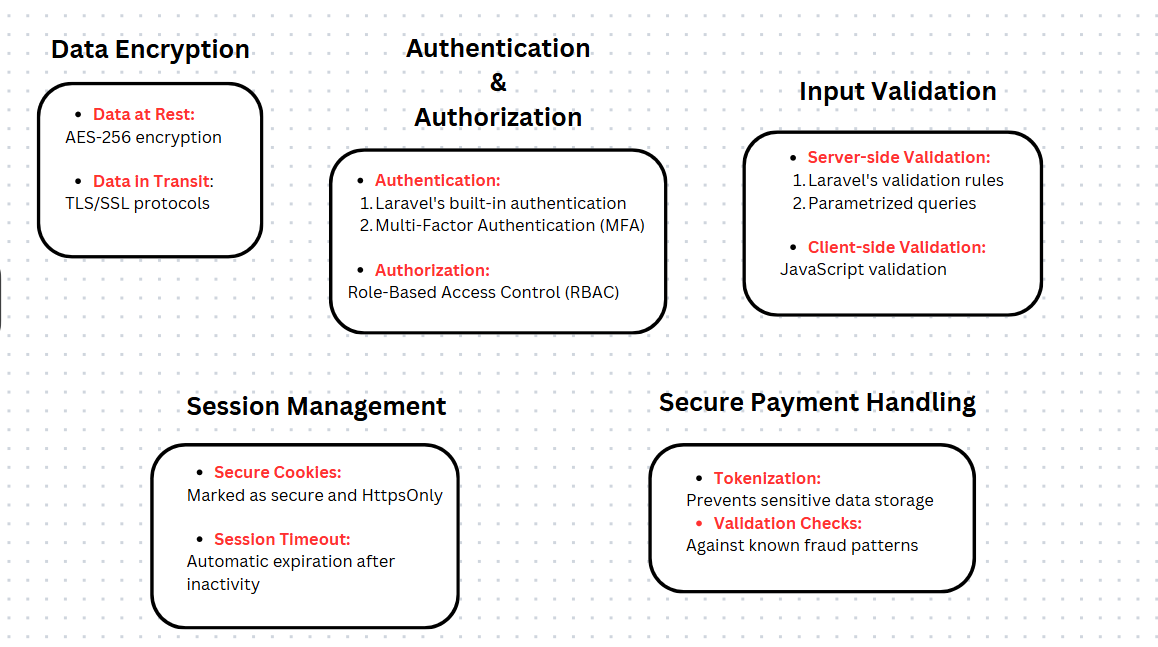
# Secure System Architectural Design

## Introduction

This document provides a detailed architectural design for the TechVista Innovations Ltd. web application. It highlights the security measures integrated into the application to ensure the protection of sensitive data and overall system integrity.





## System Architecture Overview

The system architecture for TechVista Innovations Ltd. web application follows a multi-tier architecture, comprising of the following layers:

- \*\*Presentation Layer:\*\* Responsible for handling user interactions and presenting information to users. This layer includes the web interface built using HTML, CSS, and Bootstrap.

- \*\*Application Layer:\*\* Contains the business logic of the application. This layer is developed using Laravel, a PHP framework, and includes various controllers, models, and services.

- \*\*Data Layer:\*\* Manages data storage and retrieval. This layer consists of a MySQL database that stores all application data securely.

## Security Measures

### Data Encryption

To ensure the confidentiality and integrity of sensitive data, the following encryption mechanisms are implemented:

- \*\*Data at Rest:\*\* All sensitive data stored in the MySQL database is encrypted using AES-256 encryption. This includes user passwords, payment information, and personal details.

- \*\*Data in Transit:\*\* All data transmitted between the client and server is encrypted using TLS/SSL protocols. This prevents eavesdropping and man-in-the-middle attacks.

### Authentication and Authorization

To control access to the application and its functionalities, the following mechanisms are implemented:

- \*\*Authentication:\*\* The application uses Laravel's built-in authentication system to manage user logins. Multi-factor authentication (MFA) is implemented to provide an additional layer of security.

- \*\*Authorization:\*\* Role-based access control (RBAC) is implemented to ensure that users can only access resources and perform actions that they are authorized for. This minimizes the risk of unauthorized access to sensitive information.

### Input Validation

To prevent common web application vulnerabilities such as SQL injection and Cross-Site Scripting (XSS), the following input validation mechanisms are implemented:

- \*\*Server-side Validation:\*\* All user inputs are validated on the server side using Laravel's validation rules. This ensures that only valid data is processed by the application.

- \*\*Client-side Validation:\*\* Additional validation is performed on the client side using JavaScript to provide immediate feedback to users and reduce server load.

### Session Management

To protect user sessions from hijacking or fixation attacks, the following session management practices are implemented:

- \*\*Secure Cookies:\*\* Session cookies are marked as secure and HttpOnly to prevent unauthorized access.

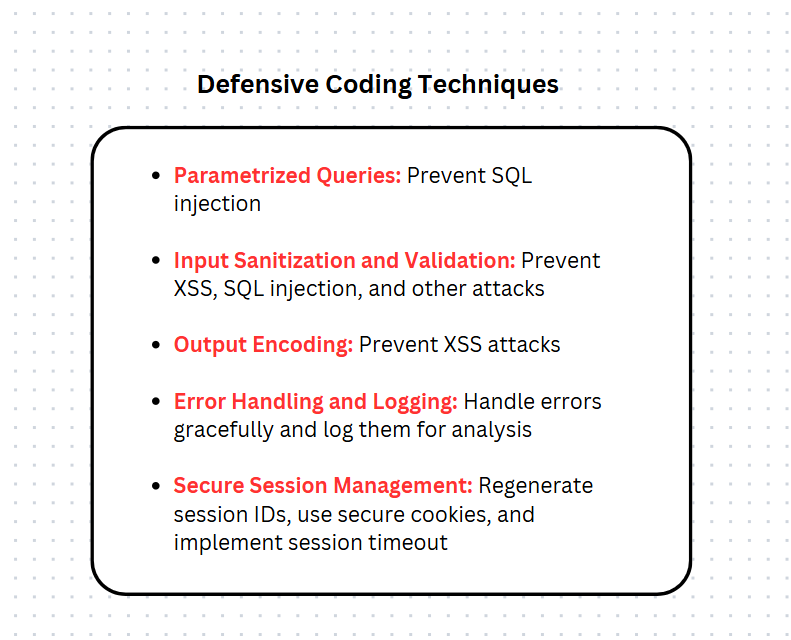
- \*\*Session Timeout:\*\* User sessions automatically expire after a period of inactivity to minimize the risk of unauthorized access.

### Secure Payment Handling

To ensure the security of financial transactions, the following measures are implemented:

- \*\*Tokenization:\*\* Payment information is tokenized to prevent sensitive data from being stored on the server.

- \*\*Validation Checks:\*\* All payment transactions are validated against known fraud patterns to detect and prevent fraudulent activities.



## Conclusion

The architectural design outlined in this document integrates comprehensive security measures to protect the TechVista Innovations Ltd. web application from various threats. By adhering to best practices for secure coding, encryption, and access control, we aim to deliver a robust and secure solution for our clients.