

UECS3294

Advanced Web Application Development

**Assignment**

Project Title: Sport Indirect, Online Shopping Website

Prepared by:

|  |  |  |
| --- | --- | --- |
| No | Student Name | Student No. |
| 1 | Kim Bei Er | 2103893 |
| 2 | Lim Juan Hong | 2105435 |
| 3 | Desmond Soo Chun Wai | 2005081 |
| 4 | Aven Ding Xian Kai | 2002759 |

**.Marking Rubrics:**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PART** | | **Criteria** | **Marks Allocated** | **Comprehensive solutions and accurate with no errors** | **Complete with room for improvement** | **Mostly complete with missing minor solutions** | **Partially complete with errors or missing significant solutions** | **Demonstrated attempts to solve** | **No attempts** | **Marks Awarded** |
| **1** | **0.8** | **0.6** | **0.4** | **0.2** | **0** |
| CO1 | 1 | Database design & migrations | **10** |  |  |  |  |  |  |  |
| 2 | Models & relationships | **10** |  |  |  |  |  |  |  |
| 3 | CRUD Operations | **10** |  |  |  |  |  |  |  |
| 4 | Input validation | **10** |  |  |  |  |  |  |  |
| 5 | Relational queries and filters | **10** |  |  |  |  |  |  |  |
|  | **TOTAL** | | **50** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | Implementation of user authentication | **10** |  |  |  |  |  |  |  |
| CO2 | 2 | Implementation of user authorization logic | **10** |  |  |  |  |  |  |  |
|  | 3 | Implementation of cookies/session | **10** |  |  |  |  |  |  |  |
|  | 4 | Presentation | **20** |  |  |  |  |  |  |  |
|  | **TOTAL** | | **50** |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| **GRAND TOTAL** | |  | **100** |  |  |  |  |  |  |  |

**Project Title**

# Project Description

Write the details here.

# Database design and migrations

## Introduction

This section provides a comprehensive overview of the project's database design and migration process. It details the **Laravel migration system**, the **entity relationships**, and the **table structures** used to efficiently manage the database schema.

A well-structured **database design** is essential for ensuring **data integrity, efficiency, and scalability**. This project follows a **relational database model**, organizing data into tables with clearly defined relationships to prevent redundancy and maintain consistency. The design focuses on **normalization**, ensuring that data is stored efficiently and queries run optimally. The **Entity-Relationship Diagram (ERD)** further illustrates the connections between different tables, helping to visualize how various entities interact.

Laravel migrations serve as a **version control system for the database**, enabling structured and trackable modifications without directly altering the database via tools like **phpMyAdmin**. This ensures that the database schema remains **consistent across different environments**, such as development, testing, and production. Additionally, migrations facilitate seamless collaboration among developers by automating table creation and modifications through **PHP-based migration scripts**.

By leveraging migrations, the project benefits from **easier database modifications, rollback capabilities, and automated deployment**, reducing human error and ensuring a well-organized, maintainable database structure. This document explores the design, structure, and implementation of the database schema, providing insights into its functionality and how it supports the application's overall workflow.

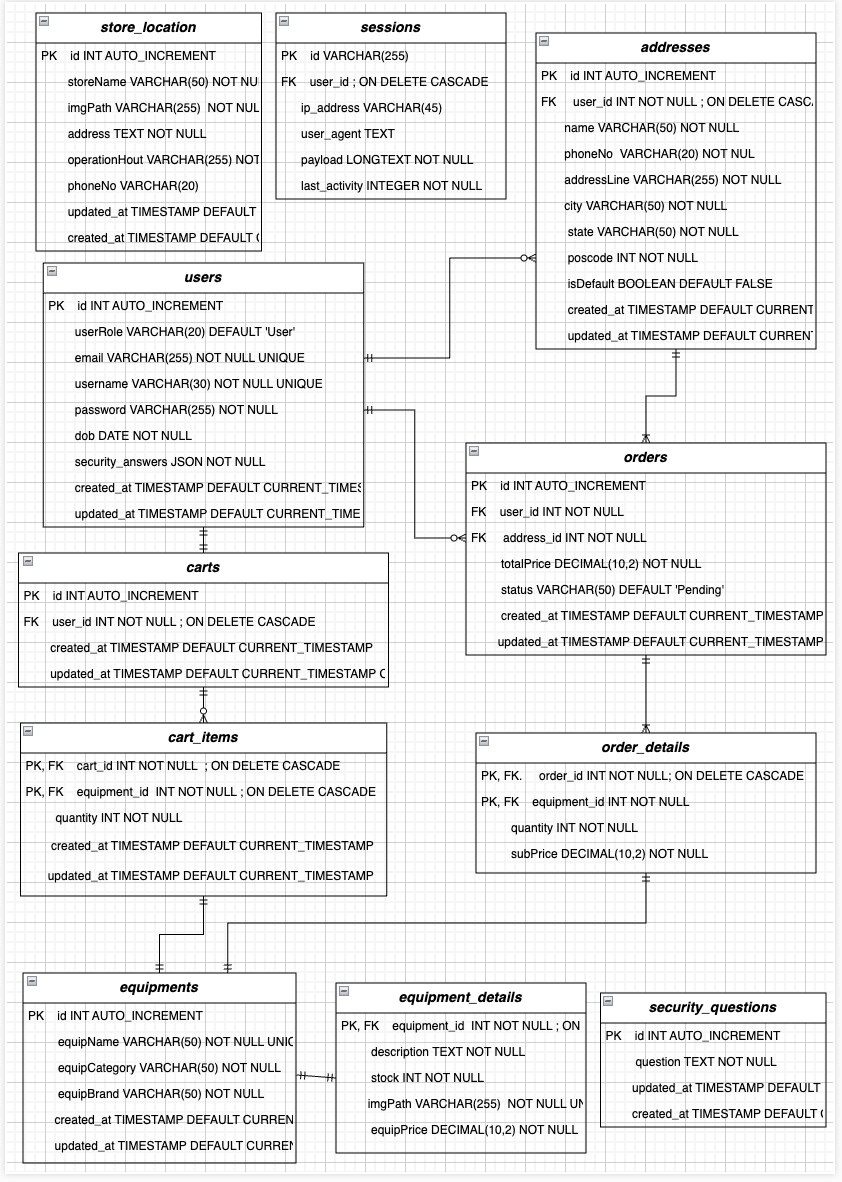
## Database Overview

The database is composed of 14 tables that used to store data regarding users, orders, products, store locations, carts, and security mechanisms. Plus, implementing **cache, migrations, and sessions** efficiently, the database maintains **performance, security, and flexibility**, ensuring a robust and scalable application. These components work together to streamline data management, optimize application performance, and provide a seamless user experience. The tables and their relationships are organized to guarantee the integrity of the data and the speed of retrieval.

List of Tables:

1. **users**
   * Stores user-related information, such as login credentials, roles, and security answers.
2. **security\_questions**
   * Contains predefined security questions for user authentication.
3. **sessions**
   * Stores active user session data, including IP addresses and user agents, to track login activity.
4. **addresses**
   * Stores multiple addresses associated with a user for order deliveries.
5. **store\_locations**
   * Manages store location details, including name, address, and operational hours.
6. **carts**
   * Stores shopping cart details for each user.
7. **cart\_items**
   * Tracks individual products added to a user’s cart.
8. **orders**
   * Maintains records of user orders.
9. **order\_details**
   * Stores detailed information about the products included in each order.
10. **equipments**
    * Stores available products, including product names and categories.
11. **equipment\_details**
    * Holds additional specifications for each product.
12. **migrations**
    * Tracks Laravel database migrations, ensuring consistent schema modifications.
13. **cache**
    * Stores temporary data to improve performance and reduce database load.
14. **cache\_locks**
    * Manages cache locks to prevent simultaneous access conflicts.

## **Entity-Relationship Diagram (ERD)**



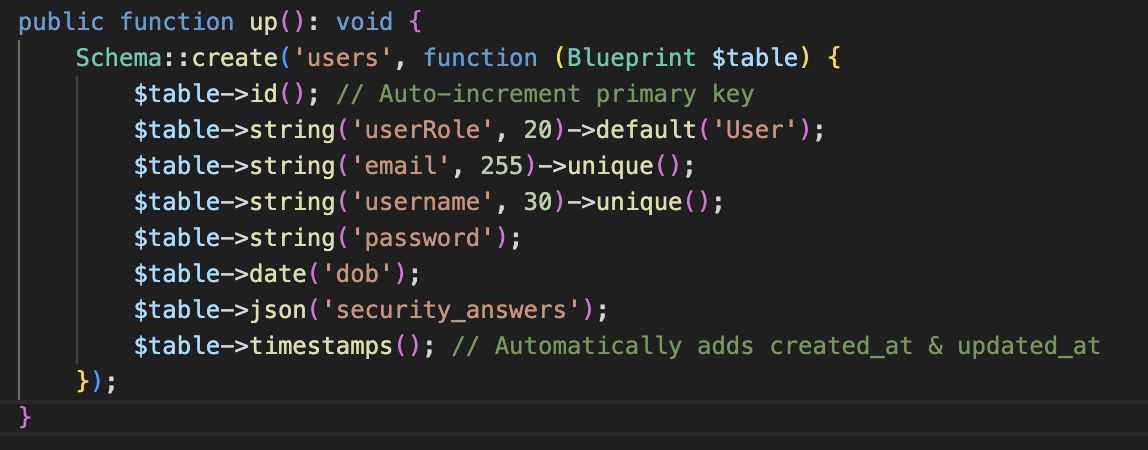
The Entity-Relationship Diagram (ERD) demonstrates the ways in which the database's tables are interconnected to ensure the consistency and integrity of the data. Relationships with numerous entities are established by the users table, which serves as the foundation of the design. Each user has the ability to maintain multiple addresses, which allows for the management of various shipping or billing locations. Additionally, users have the ability to generate orders that are associated with both the users and addresses tables, guaranteeing that each order is directly linked to a specific customer and delivery location.

On the other hand, the orders table maintains a one-to-many relationship with order\_details, which manages the individual items in each order. To retrieve product’s basic information, the products table is consulted. Additionally, users have the ability to add products to their carts, which are comprised of multiple cart\_items, thereby establishing another one-to-many relationship. The product\_details table is a one-to-one relationship that further refines the products table by including supplementary attributes such as stock and pricing.

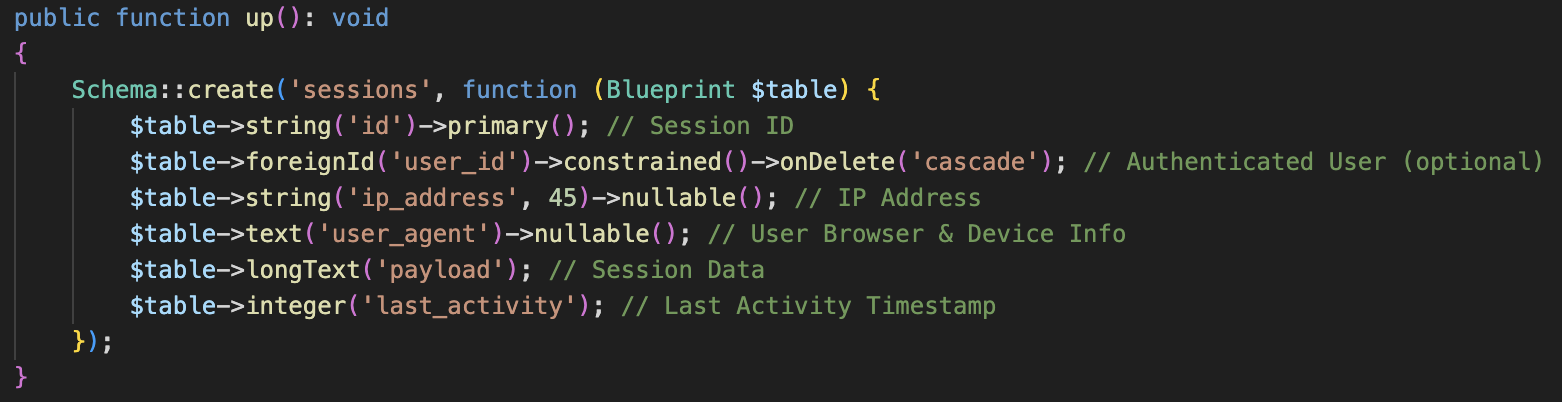
At the same time, the store\_locations table functions autonomously, storing information regarding physical stores. Users' active sessions are maintained by the sessions table, while account management is improved by the security\_questions table. This mechanism is particularly useful for **password recovery**, as users must provide the correct answer to regain access to their accounts. The cache and migrations tables also facilitate database versioning and system performance. A well-structured relational database is the result of the collective relationships, which guarantee efficient operations and seamless data management.

## Table Structures and Migrations

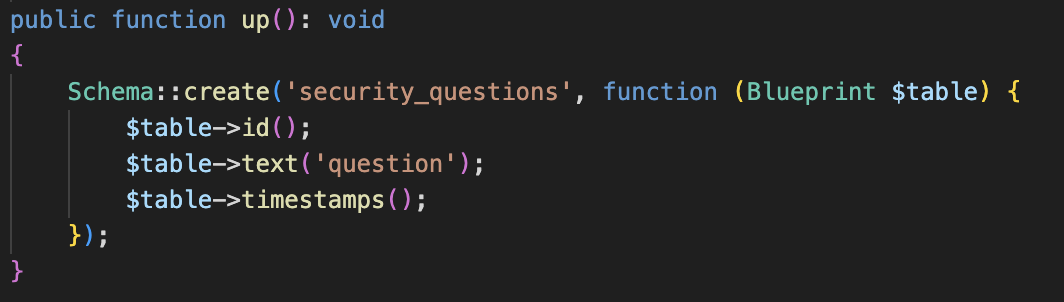
### Users Table



### Sessions Table



### Security\_questions Table



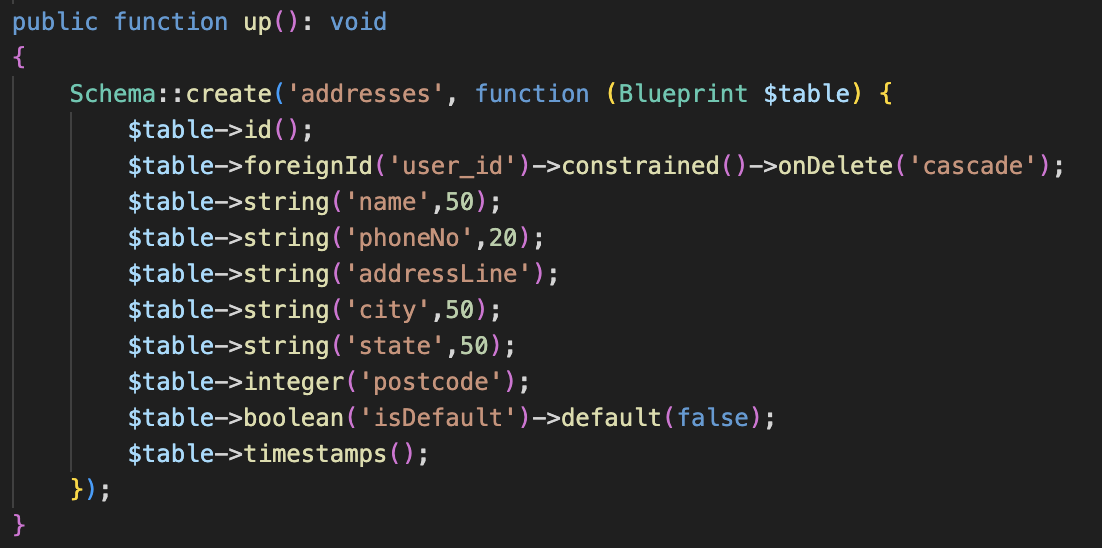
### Carts Table



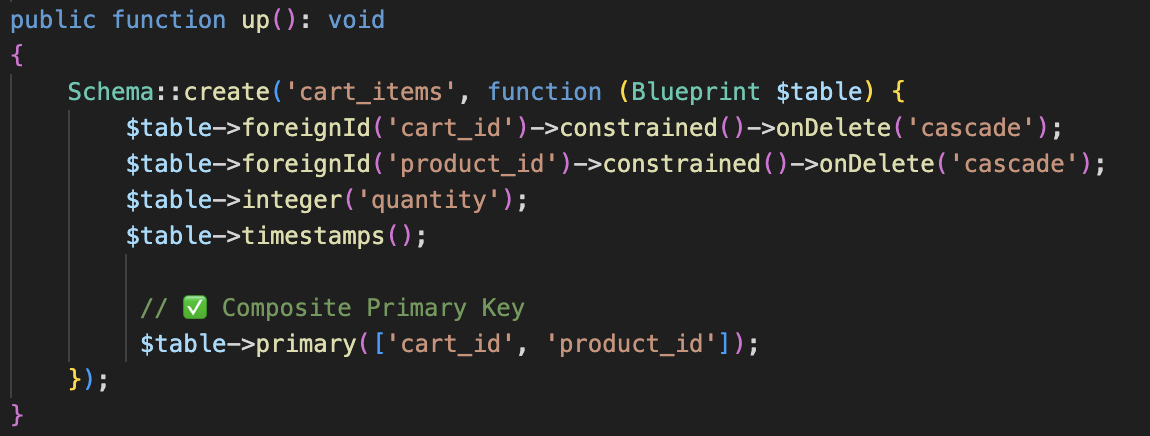
### Products Table



### Addresses Table



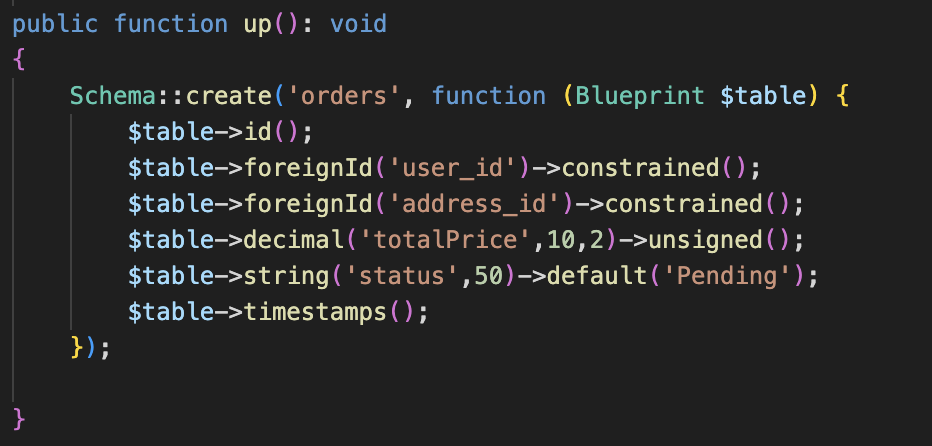
### Cart\_items Table



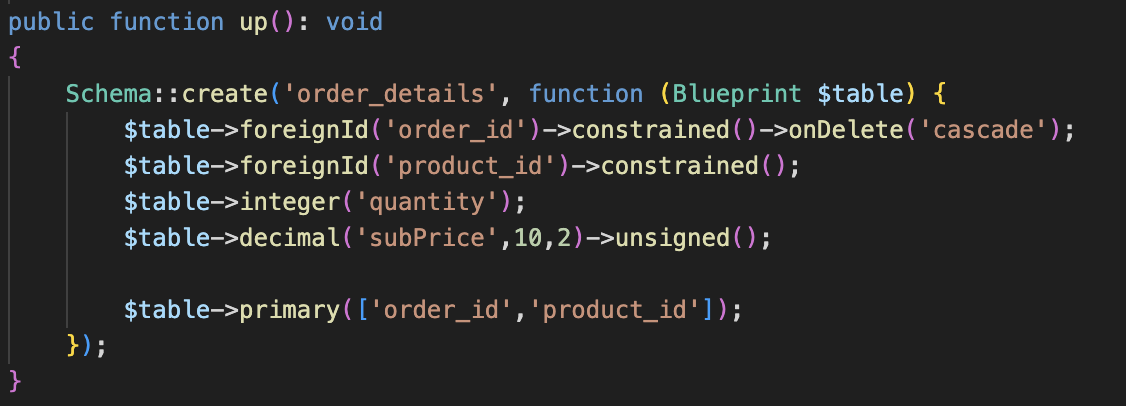
### Product\_details Table



### Orders Table



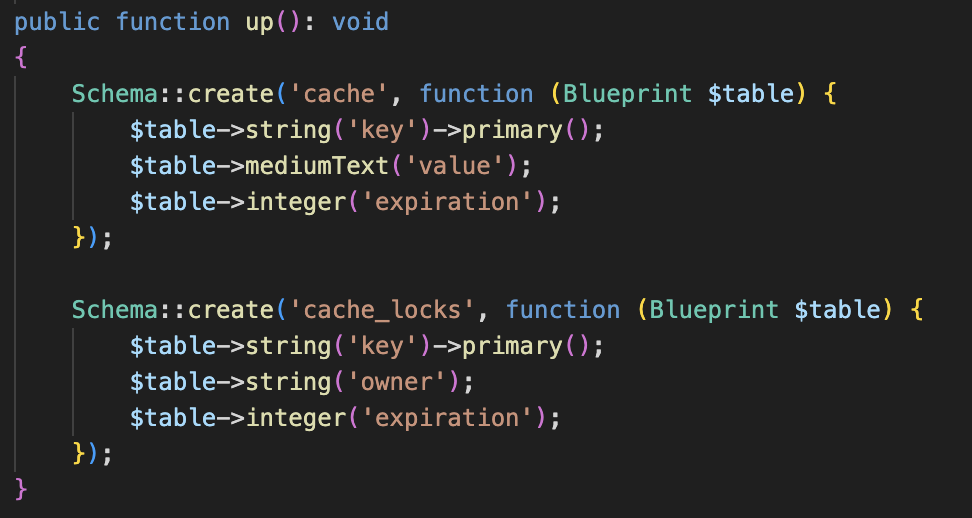
### Order\_details Table



### Store\_locations Table



### Cache and cache\_locks Table



# Models and relationships

## Introduction

Laravel's Eloquent ORM (Object-Relational Mapping) is used to interact with the database tables. Each table has a corresponding model that defines its structure and relationships. This section outlines the models and their relationships based on the database schema.

## Models

### Models

### Summary

# CRUD Operations

Write the details here.

## Sub-section

Write the details here.

# Input validation

Write the details here.

## Sub-section

Write the details here.

# Relational queries and search filters

Write the details here.

## Sub-section

Write the details here.

# Authentication logic

Write the details here.

## Sub-section

Write the details here.

# Authorization logic

Write the details here.

## Sub-section

Write the details here.

# Cookies and session implementation

Write the details here.

## Sub-section

Write the details here.