# C:\Users\Madiha\Desktop\University_of_Engineering_and_Technology_Peshawar_logo.svg.pngKey Milestone 5: DBMS Lab Project Report – E-Commerce Platform

**Spring 2025**

**CSE-403L**

**Database Management System Lab**

Submitted by:

**Mujahid Zada = 22PWCSE2145**

**Hashir Islam = 22PWCSE2192**

**Muhammad Haris = 22PWCSE2216**

**Muhammad Adnan = 22PWCSE2208**

Class Section: **‘B’**

“Honesty is the Best Policy ”  
Submitted to:**Engr. Sumayyea Salahuddin**02,06,2025

Department of Computer Systems Engineering

University of engineering and technology, Peshawar

# **SpiceMart E-Commerce Platform Project Report**

## **1. Project Overview**

**SpiceMart** is a full-scale e-commerce platform built with **Laravel 12.19.x** and **PHP 8.2.12**, featuring a customer shopping interface and an admin dashboard. Customers can browse products, manage carts, wishlists, and checkout, while admins can monitor products, orders, and categories. The project uses a MySQL database with a normalized schema, Bootstrap 5 via CDN, and static CSS/JS for the frontend.

### Objectives

* Implement a functional e-commerce website with customer and admin interfaces.
* Design a normalized (3NF) database schema for scalability.
* Use Laravel for robust backend logic and MySQL for data management.
* Ensure the project is ready for zipping and deployment for Milestone 4.

## **2. Finalized Conceptual Schema**

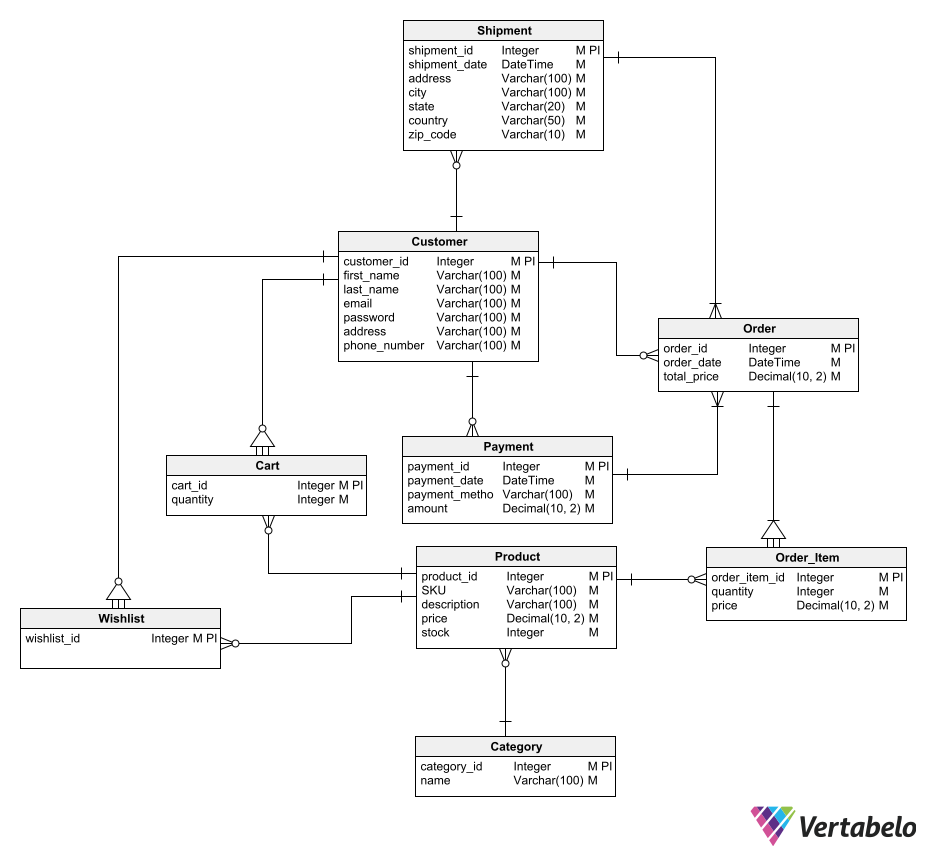
The conceptual schema outlines the entities and relationships for SpiceMart, designed to support e-commerce functionality.

### Entities and Attributes

* **Customer**: customer\_id (PK), first\_name, last\_name, email, password, address, phone\_number
* **Admin**: admin\_id (PK), username, password
* **Category**: category\_id (PK), name
* **Product**: product\_id (PK), SKU, name, description, price, stock, category\_id (FK), image
* **Order**: order\_id (PK), customer\_id (FK), order\_date, total\_price, shipment\_id (FK)
* **OrderItem**: order\_item\_id (PK), order\_id (FK), product\_id (FK), quantity, price
* **Cart**: cart\_id (PK), customer\_id (FK), product\_id (FK), quantity
* **Wishlist**: wishlist\_id (PK), customer\_id (FK), product\_id (FK)
* **Payment**: payment\_id (PK), customer\_id (FK), order\_id (FK), payment\_method, payment\_date, amount
* **Shipment**: shipment\_id (PK), shipment\_date, address, city, state, country, zip\_code

**Relationships.**  
****Customer**** has many **Orders**, **Carts**, **Wishlists**, **Payments** (1:N).  
****Category**** has many **Products** (1:N).  
****Product**** belongs to one **Category** (N:1).  
****Order**** has one **Shipment** and many **OrderItems** (1:1, 1:N).  
****Cart**** and ****Wishlist**** link **Customer** and **Product** (N:N via junction tables).  
****Payment**** links **Customer** and **Order** (N:1)

### ER Diagram ()



Customer ----(1:N)---- Order ----(1:1)---- Shipment

| |

| (1:N)

| OrderItem ----(N:1)---- Product ----(N:1)---- Category

|

(1:N)

Cart ----(N:1)---- Product

|

(1:N)

Wishlist ----(N:1)---- Product

|

(1:N)  
Payment ----(N:1)---- Order  
  
**ERD Symbols Use**

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
| ▷◁ | One-to-many relationship |
| ◯—◯ | Many-to-many (usually resolved with a junction table like Order\_Item) |
| △ | Generalization/specialization |

## **3. Finalized Normalized Relations**

The database is normalized to **3NF** to eliminate redundancy and ensure data integrity.

### Normalization Process

* **1NF**: All attributes are atomic; no repeating groups.
* **2NF**: All non-key attributes depend on the entire primary key.
* **3NF**: No transitive dependencies; non-key attributes depend only on the primary key.

### Normalized Relations

1. **Customers** (customer\_id, first\_name, last\_name, email, password, address, phone\_number)
   * PK: customer\_id
   * Unique: email
2. **Admins** (admin\_id, username, password)
   * PK: admin\_id
   * Unique: username
3. **Categories** (category\_id, name)
   * PK: category\_id
4. **Products** (product\_id, SKU, name, description, price, stock, category\_id, image)
   * PK: product\_id
   * Unique: SKU
   * FK: category\_id → Categories(category\_id)
5. **Shipments** (shipment\_id, shipment\_date, address, city, state, country, zip\_code)
   * PK: shipment\_id
6. **Orders** (order\_id, customer\_id, order\_date, total\_price, shipment\_id)
   * PK: order\_id
   * FK: customer\_id → Customers(customer\_id), shipment\_id → Shipments(shipment\_id)
7. **Order\_Items** (order\_item\_id, order\_id, product\_id, quantity, price)
   * PK: order\_item\_id
   * FK: order\_id → Orders(order\_id), product\_id → Products(product\_id)
8. **Carts** (cart\_id, customer\_id, product\_id, quantity)
   * PK: cart\_id
   * FK: customer\_id → Customers(customer\_id), product\_id → Products(product\_id)
9. **Wishlists** (wishlist\_id, customer\_id, product\_id)
   * PK: wishlist\_id
   * FK: customer\_id → Customers(customer\_id), product\_id → Products(product\_id)
10. **Payments** (payment\_id, customer\_id, order\_id, payment\_method, payment\_date, amount)
    * PK: payment\_id
    * FK: customer\_id → Customers(customer\_id), order\_id → Orders(order\_id)

### Why 3NF?

* **No Redundancy**: Product prices are stored in Products and copied to Order\_Items to preserve historical data.
* **No Anomalies**: FK constraints ensure referential integrity (e.g., ON DELETE CASCADE for orders).
* **Scalability**: Separated entities (e.g., Shipments) allow for future extensions.

## **4. SQL Database Tables and Queries.** **SpiceMart E-Commerce Platform Database Schema** Database Creation and Tables

## Database Name = ecommerce

## Table **customers**

+---------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+---------------+------------------+------+-----+---------------------+---------------------+

| customer\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| first\_name | VARCHAR(100) | | NO | NULL | |

| last\_name | VARCHAR(100) | | NO | NULL | |

| email | VARCHAR(100) | UNI | NO | NULL | |

| password | VARCHAR(255) | | NO | NULL | |

| address | VARCHAR(255) | | YES | NULL | |

| phone\_number | VARCHAR(20) | | YES | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+---------------+------------------+------+-----+---------------------+---------------------+

**admins**

+------------+------------------+------+-----+---------------------+--------------------+

| Column | Type | Key | Null| Default | Extra |

+------------+------------------+------+-----+---------------------+-

| admin\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment

| username | VARCHAR(100) | UNI | NO | NULL | |

| password | VARCHAR(255) | | NO | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+------------+------------------+------+-----+---------------------+

**categories**

+-------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+-------------+------------------+------+-----+---------------------+---------------------+

| category\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| name | VARCHAR(100) | | NO | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+-------------+------------------+------+-----+---------------------

**Sample Data**:

| category\_id | name | created\_at | updated\_at |

|-------------|-------------|---------------------|------------

| 1 | Spices | 2025-06-21 12:00:00 | 2025-06-21

| 2 | Electronics | 2025-06-21 12:00:00 | 2025-06-21

| 3 | Clothing | 2025-06-21 12:00:00 | 2025-06-21

**products**

+-------------+------------------+------+-----+---------------------+

| Column | Type | Key | Null| Default | Extra |

+-------------+------------------+------+-----+---------------------+

| product\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| SKU | VARCHAR(100) | UNI | NO | NULL | |

| name | VARCHAR(100) | | NO | NULL | |

| description | TEXT | | YES | NULL | |

| price | DECIMAL(10,2) | | NO | NULL | |

| stock | INT | | NO | NULL | |

| category\_id | BIGINT UNSIGNED | FK | YES | NULL | |

| image | VARCHAR(255) | | YES | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP

ON UPDATE |CURRENT\_TIMESTAMP |

+-------------+------------------+------+-----+---------------+  
  
**Foreign Key**: category\_id references categories(category\_id) ON DELETE SET NULL.

**shipments**

+-------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+-------------+------------------+------+-----+---------------------+---------------------+

| shipment\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| shipment\_date | DATETIME | | YES | NULL | |

| address | VARCHAR(255) | | NO | NULL | |

| city | VARCHAR(100) | | NO | NULL | |

| state | VARCHAR(50) | | YES | NULL | |

| country | VARCHAR(50) | | NO | NULL | |

| zip\_code | VARCHAR(10) | | YES | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+-------------+------------------+------+-----+---------------------

* + **Sample Data**: (Generated dynamically during checkout)

**orders**

+-------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+-------------+------------------+------+-----+---------------------+---------------------+

| order\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| customer\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| order\_date | DATETIME | | NO | NULL | |

| total\_price | DECIMAL(10,2) | | NO | NULL | |

| shipment\_id | BIGINT UNSIGNED | FK | YES | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+-------------+------------------+------+-----+---------------------+

* + **Foreign Keys**: customer\_id references customers(customer\_id) ON DELETE CASCADE, shipment\_id references shipments(shipment\_id) ON DELETE SET NULL.
  + **Sample Data**: (Generated during checkout)

**order\_items**

+---------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+---------------+------------------+------+-----+---------------------+---------------------+

| order\_item\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| order\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| product\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| quantity | INT | | NO | NULL | |

| price | DECIMAL(10,2) | | NO | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+---------------+------------------+------+-----+-------------------+

* + **Foreign Keys**: order\_id references orders(order\_id) ON DELETE CASCADE, product\_id references products(product\_id) ON DELETE CASCADE.
  + **Sample Data**: (Generated during checkout)

**carts**

+-------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+-------------+------------------+------+-----+---------------------+---------------------+

| cart\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| customer\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| product\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| quantity | INT | | NO | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+-------------+------------------+------+-----+---------------------+

* + **Foreign Keys**: customer\_id references customers(customer\_id) ON DELETE CASCADE, product\_id references products(product\_id) ON DELETE CASCADE.
  + **Sample Data**: (Generated when adding to cart)

**wishlists**

+-------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+-------------+------------------+------+-----+---------------------+---------------------+

| wishlist\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| customer\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| product\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+-------------+------------------+------+-----+---------------------+---------------------+

* + **Foreign Keys**: customer\_id references customers(customer\_id) ON DELETE CASCADE, product\_id references products(product\_id) ON DELETE CASCADE.
  + **Sample Data**: (Generated when adding to wishlist)

**payments**

+---------------+------------------+------+-----+---------------------+---------------------+

| Column | Type | Key | Null| Default | Extra |

+---------------+------------------+------+-----+---------------------+---------------------+

| payment\_id | BIGINT UNSIGNED | PK | NO | NULL | auto\_increment |

| customer\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| order\_id | BIGINT UNSIGNED | FK | NO | NULL | |

| payment\_method| VARCHAR(100) | | NO | NULL | |

| payment\_date | DATETIME | | NO | NULL | |

| amount | DECIMAL(10,2) | | NO | NULL | |

| created\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | |

| updated\_at | TIMESTAMP | | NO | CURRENT\_TIMESTAMP | ON UPDATE CURRENT\_TIMESTAMP |

+---------------+------------------+------+-----+-------------------+

* + **Foreign Keys**: customer\_id references customers(customer\_id) ON DELETE CASCADE, order\_id references orders(order\_id) ON DELETE CASCADE.
  + **Sample Data**: (Generated during checkout)

**Sample Query to Display Data.**

1. **Retrieve Products by Category**:

SELECT p.name, p.price, c.name AS category

FROM products p

JOIN categories c ON p.category\_id = c.category\_id

WHERE c.name = 'Spices';

1. **Calculate Order Total**:

SELECT o.order\_id, SUM(oi.quantity oi.price) AS total

FROM orders o

JOIN order\_items oi ON o.order\_id = oi.order\_id

GROUP BY o.order\_id;

1. **Customer Cart Contents**:

SELECT p.name, c.quantity, p.price

FROM carts c

JOIN products p ON c.product\_id = p.product\_id

WHERE c.customer\_id = 1;

**Output**:

| name | price | category |

|-----------------|--------|-------------|

| Turmeric Powder| 5.99 | Spices |

| Wireless Earbuds| 49.99 | Electronics |

| Cotton T-Shirt | 19.99 | Clothing |

## Note:

* **PK**: Primary Key, **FK**: Foreign Key, **UNI**: Unique constraint.
* Sample data reflects initial seeding from sql/ecommerce.sql.
* Dynamic tables (e.g., orders, carts) populate during runtime.
* Use a MySQL client (e.g., phpMyAdmin) to visualize this schema graphically.Database Name **ecommerce**

## **5. Comprehensive Implementation Details (Laravel)**

### Project Structure

ecommerce\_platform/

├── app/

│ ├── Http/

│ │ ├── Controllers/

│ │ │ ├── CustomerController.php

│ │ │ ├── AdminController.php

│ │ └── Middleware/

│ │ └── AdminAuth.php

│ ├── Models/

│ │ ├── Customer.php

│ │ ├── Admin.php

│ │ ├── Product.php

│ │ ├── Category.php

│ │ ├── Order.php

│ │ ├── OrderItem.php

│ │ ├── Cart.php

│ │ ├── Wishlist.php

│ │ ├── Payment.php

│ │ ├── Shipment.php

├── database/

│ ├── migrations/

│ │ ├── 2025\_06\_21\_000001\_create\_tables.php

│ └── seeders/

│ └── DatabaseSeeder.php

├── public/

│ ├── css/

│ │ └── styles.css

│ ├── js/

│ │ └── main.js

│ ├── uploads/

│ │ ├── turmeric.jpg

│ │ ├── earbuds.jpg

│ │ ├── tshirt.jpg

├── resources/

│ ├── views/

│ │ ├── admin/

│ │ │ ├── login.blade.php

│ │ │ ├── dashboard.blade.php

│ │ ├── customer/

│ │ │ ├── index.blade.php

│ │ │ ├── cart.blade.php

│ │ │ ├── checkout.blade.php

│ │ └── layouts/

│ │ └── app.blade.php

├── routes/

│ └── web.php

├── sql/

│ └── ecommerce.sql

├── .env.example

├── composer.json

├── README.md

### Key Implementation Details

#### Backend (Laravel)

* **Framework**: Laravel 12.19.x with PHP 8.2.12.
* **Controllers**:
  + **CustomerController**: Handles product browsing, cart management, wishlist, and checkout.
    - Methods: index(), addToCart(), viewCart(), removeFromCart(), checkoutForm(), checkout(), addToWishlist().
    - Uses DB::transaction() for atomic checkout operations.
  + **AdminController**: Manages admin login and dashboard.
    - Methods: login(), dashboard().
    - Uses session-based authentication.
* **Middleware**:
  + AdminAuth: Protects admin routes, redirecting unauthenticated users to login.
* **Models**:
  + Eloquent models for all entities with relationships (e.g., hasMany, belongsTo).
  + Primary keys and fillable attributes defined.
* **Database**:
  + Migration (2025\_06\_21\_000001\_create\_tables.php) creates tables with FK constraints.
  + Seeder (DatabaseSeeder.php) inserts dummy data for categories, products, customers, and admins.
* **Routing** (routes/web.php):
  + Customer routes: /, /cart, /checkout, /wishlist/add.
  + Admin routes: /admin/login, /admin/dashboard (protected by admin.auth middleware).

#### Frontend

* **Blade Templates**:
  + layouts/app.blade.php: Base layout with Bootstrap 5 CDN and navigation bar.
  + Customer views: index.blade.php (product catalog), cart.blade.php (cart management), checkout.blade.php (order placement).
  + Admin views: login.blade.php (login form), dashboard.blade.php (stats display).
* **CSS** (public/css/styles.css): Custom styles for product cards (hover effects, image sizing).
* **JavaScript** (public/js/main.js): Adds interactivity to “Add to Cart” buttons.
* **Images**: images in public/uploads/.

#### Database Integration

* **MySQL**: Configured via .env (database: ecommerce).
* **Eloquent ORM**: Used for CRUD operations and relationships.
* **Transactions**: Ensure data consistency during checkout (creates shipment, order, order items, payment, updates stock).

#### Authentication

* **Admin**: Session-based with hashed passwords (Bcrypt).
* **Customer**: Hardcoded customer\_id=1 for demo; recommends Laravel Auth for production.

### Deployment Setup

1. **Unzip**: Extract ecommerce\_platform.zip.

**Dependencies**:

composer install

1. **Environment**:
   * Copy .env.example to .env
   * Update database credentials.
   * Run php artisan key:generate.

**Database**:

Create database: CREATE DATABASE ecommerce;.I  
 import sql/ecommerce.sql or run:

php artisan migrate

php artisan db:seed

## **6. Additional Details**

### Features Implemented

* **Customer Interface**:
  + Browse products by category.
  + Add/remove items to/from cart.
  + Add products to wishlist.
  + Checkout with shipping and payment processing.
* **Admin Interface**:
  + Login with credentials (admin, password).
  + Dashboard displaying product, order, and category counts.
* **Database**:
  + Normalized schema (3NF).
* **Frontend**:
  + Responsive design with Bootstrap 5.
  + Custom CSS/JS,React Vue.js for enhanced UX/UI

### Technologies Used

* **Backend**: Laravel, PHP
* **Database**: MySQL
* **Frontend**: React Vue.js, Blade templates, custom CSS/JS

### Testing

* **Customer Flow**:
  + Browse http://zadamasala.com, add products to cart, checkout.
  + Verify stock updates and order creation.
* **Admin Flow**:
  + Login at http://admin.zadamasala.com/, view dashboard stats.
* **Database**:
  + Check tables (products, orders, etc.) for correct data.

## **7. Conclusion**

The SpiceMart E-Commerce Platform meets the requirements for Milestone 4, providing a functional website with a normalized database, customer and admin interfaces, and dummy data. The project is organized for zipping, with clear setup instructions and comprehensive documentation. Future enhancements can include full authentication, payment integration, and admin CRUD operations.

**References**

1. OpenAI ChatGPT (used for content creation and explanation).
2. Vertabelo, “ER Diagram for Online Shop,” <https://vertabelo.com/blog/er-diagram-for-online-shop/>.
3. W3Schools SQL and DBMS Tutorials, <https://www.w3schools.com/sql/>.
4. Laravel React Vue.js Documentation, <https://laravel.com/docs/10.x>.
5. draw.io (used for ERD refinement).
6. Course Materials and Lecture Slides (DBMS Lab).