JOSE MARIA COLLEGE

Philippie-Japan Friendship Highway, Sasa, Davao City
COLLEGE OF INFORMATION TECHNOLOGY EDUCATION

Spotlight Products Web Application

IT 203 and IT ELECT 3 - Final Project

Submitted By:

Humphrey Myles C. Lozano

INTRODUCTION

1.1 Project Overview

Spotlight Products is a web application developed using the Laravel framework. Its primary objective is to provide local businesses and shops with a platform to promote their products. This can be gadgets, makeup, and more. The application also allows sellers to upload their products, subject to admin approval for security purposes. Once approved, the products are shown on the front page, featured for all sellers and buyers, and catered to.

1.2 Project Scope

The application includes features such as product promotion by businesses, seller uploads with pending admin approval, a search functionality, as well as an exclusive admin page for developers, a login page, a registration page, a buyers' page, a transaction page, and an "add to cart" functionality.

Given the time constraints, the implementation of the transaction page may be difficult and may become a scope creep in itself depending on the level of complexity. The same can also be said regarding the add-to-cart functionality.

1.3 Document Purpose

This document serves as a comprehensive guide to developing the Spotlight Products web application. It is intended for developers, project managers, and anyone involved in the development process and as a future reference.

REQUIREMENTS ANALYSIS

2.1 Functional Requirements

- User Registration and Login: Users would be able to create an account by providing necessary information such as username, email, and password. Upon registration, they can log in securely using their credentials to access the platform's features and services.

- Product Upload: Authorized users, typically sellers or vendors, should be able to upload details of their products to the platform. This includes product images, descriptions, pricing, and other relevant information.

- Admin Approval System: Uploaded products should go through an approval process by the admin before becoming visible to other users on the platform. The admin reviews the uploaded products to ensure they meet the platform's standards and guidelines.

- Product Promotion: The platform would have features to promote certain products, such as highlighting them on the homepage, offering discounts, or featuring them in promotional campaigns to increase visibility and sales.

- Search Functionality: Users should be able to search for products easily using keywords, filters, or categories. The search functionality helps users find specific products quickly among the vast inventory available on the platform.

- Add-to-cart functionality: Users would be able to add products they are interested in purchasing to their shopping cart. This feature allows users to continue browsing and adding more items before proceeding to checkout.

- Transaction Page: Once users are ready to make a purchase, they can proceed to the transaction page where they can review their selected items, choose payment and shipping options, and complete the purchase securely.

** - **uncertain**

2.2 Non-Functional Requirements

- Performance:
  The application should respond quickly to user interactions, and handle increased traffic without slowing down, and code and database queries should be optimized for speed.

- Security:
  User data and transactions should be encrypted and only authorized users should have access to the application. Different levels of access should be enforced based on user roles.

- Reliability:
  The application should be available with minimal downtime and it should recover gracefully from errors. Regular backups of data should be performed.

- Usability:
  The user interface should be easy to use and accessible to users with disabilities. Comprehensive documentation should be provided.

- Maintainability:
  The application should be modular, easy to update, and well-documented, adhering to standards. Version control should be used to track changes.

- Compatibility:
  The application should work well across different web browsers. It should be compatible with various devices, including desktops and mobile devices.

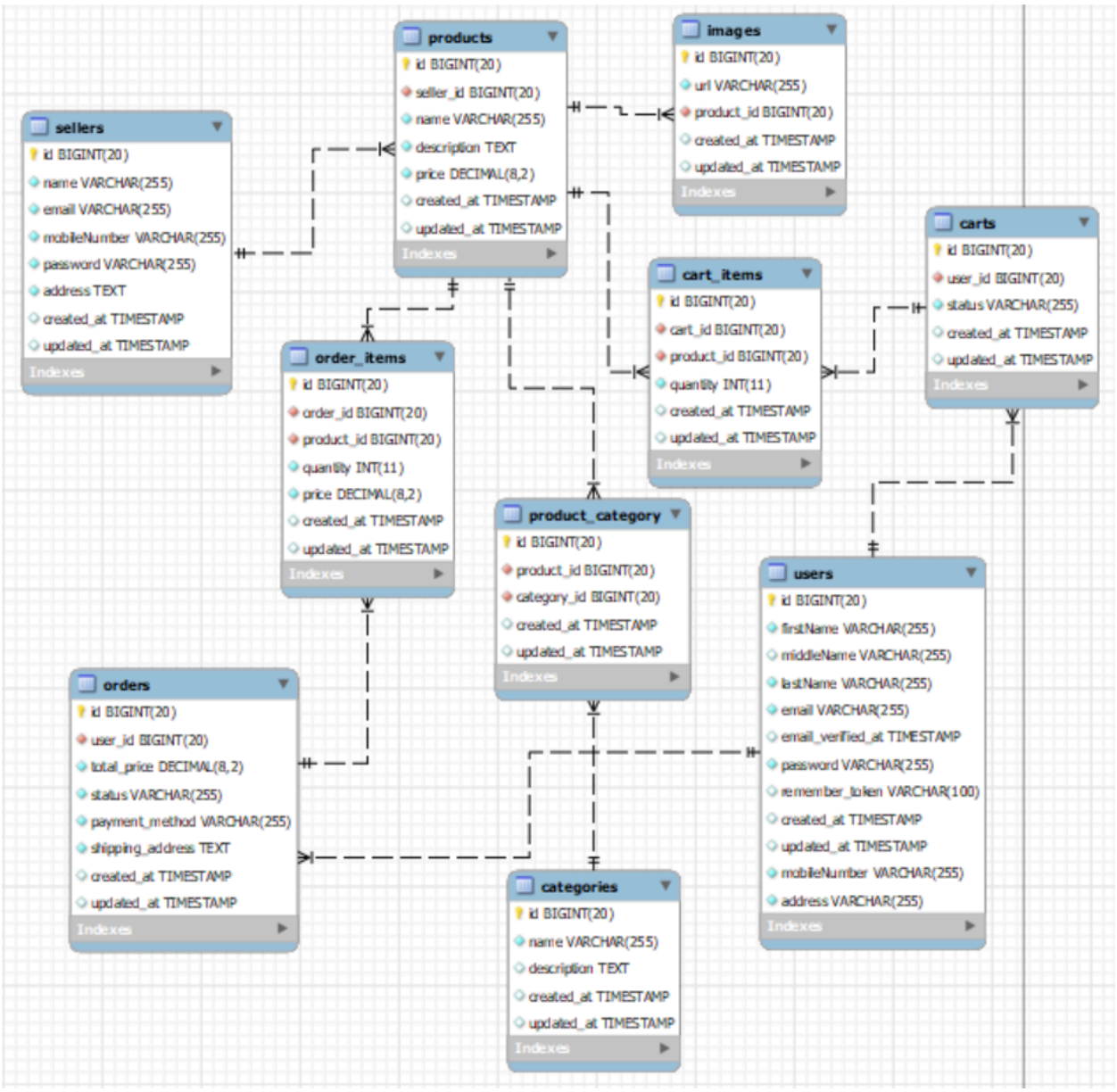
DESIGN

3.1 System Architecture

The application follows the MVC (Model-View-Controller) architecture inherent in Laravel applications. The system follows the client-server architecture where the client side of the application is what the user interacts with (views/blade.php) and the server side of the application is where the main logic resides (models, databases, controllers, middleware, routes, etc.). It also follows the Request-Response architecture wherein the client sends a request and the server responds to that request.

3.2 Database Design

The database schema includes tables for Users, Products, Buyers, Admins, and many in-between tables. Relationships are defined as per the requirements of the application.



Note about admin tables not being related to other tables:

By not establishing relationships at the schema level, the system gains flexibility. It allows for easier modifications and expansions in the future without being constrained by predefined relationships. Admin-related functionality can evolve independently without affecting other parts of the system.

JOSE MARIA COLLEGE

Philippie-Japan Friendship Highway, Sasa, Davao City
COLLEGE OF INFORMATION TECHNOLOGY EDUCATION

DATA DICTIONARY:

Table: users				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	User ID, Auto generated
firstName	string	50	Not null	First name of the user
middleName	string	50	Nullable	Middle name of the user
lastName	string	50	Not null	Last name of the user
email	string	100	Unique, Not null	Email ID of the user
email_verified_at	timestamp		Nullable	Email verification timestamp
password	string	255	Not null	Login password for the user
remember_token	string	100	Nullable	Token for remembering the user
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
mobileNumber	string	15	Not null	Mobile number of the user
address	string	255	Not null	Address of the user
Table: products				
Field Name	Data Type	Field Length	Constraint	Description

id	bigint		Primary key	Product ID, Auto generated
name	string	100	Not null	Name of the product
description	text		Not null	Description of the product
price	decimal	8, 2	Not null	Price of the product
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
seller_id	bigint		Foreign key	Foreign key referencing sellers.id
Table: admins				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Admin ID, Auto generated
name	string	50	Not null	Name of the admin
password	string	255	Not null	Login password for the admin
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: images				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Image ID, Auto generated

url	string	255	Not null	URL of the image
product_id	bigint		Foreign key	Foreign key referencing products.id
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: sellers				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Seller ID, Auto generated
name	string	50	Not null	Name of the seller
email	string	100	Unique, Not null	Email ID of the seller
mobileNumber	string	15	Unique, Not null	Mobile number of the seller
password	string	255	Not null	Login password for the seller
address	text		Not null	Address of the seller
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: orders				
Field Name	Data Type	Field Length	Constraint	Description

id	bigint		Primary key	Order ID, Auto generated
user_id	bigint		Foreign key	Foreign key referencing users.id
total_price	decimal	8, 2	Not null	Total price of the order
status	string	50	Not null	Status of the order
payment_method	string	50	Not null	Payment method used
shipping_address	text		Not null	Shipping address for the order
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: order_items				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Order item ID, Auto generated
order_id	bigint		Foreign key	Foreign key referencing orders.id
product_id	bigint		Foreign key	Foreign key referencing products.id
quantity	integer		Not null	Quantity of the product ordered

price	decimal	8, 2	Not null	Price of the product ordered
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: carts				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Cart ID, Auto generated
user_id	bigint		Foreign key	Foreign key referencing users.id
status	string	50	Not null	Status of the cart
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: cart_items				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Cart item ID, Auto generated
cart_id	bigint		Foreign key	Foreign key referencing carts.id
product_id	bigint		Foreign key	Foreign key referencing products.id

quantity	integer		Not null	Quantity of the product in the cart
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: categories				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Category ID, Auto generated
name	string	50	Not null	Name of the category
description	text		Nullable	Description of the category
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp
Table: product_category				
Field Name	Data Type	Field Length	Constraint	Description
id	bigint		Primary key	Record ID, Auto generated
product_id	bigint		Foreign key	Foreign key referencing products.id

category_id	bigint		Foreign key	Foreign key referencing categories.id
created_at	timestamp		Not null	Record creation timestamp
updated_at	timestamp		Not null	Record update timestamp

IMPLEMENTATION

4.1 Technologies Used

Frontend:

- HTML/CSS/JavaScript
- Bootstrap
- Laravel Framework

Backend:

- XAMPP (PHP and MySQL)

Version Control:

- Github

4.2 Development Environment Setup

Instructions for setting up the Laravel development environment can be found on the Laravel documentation page.

4.3 Coding Standards

The project follows the PSR-12: Extended Coding Style PHP coding standard. The utilization of the artisan command-line interface included with Laravel makes for the easy creation of controllers, models, and migration of tables.

4.4 Conclusion

To conclude, this proposal entails benefits for those involved in the development of said system:

- Project Submitter:
  - Gain valuable hands-on experience in web application development using modern technologies like Laravel.
  - Showcase my skills and capabilities through the successful completion of a comprehensive project.
  - Demonstrate my ability to analyze requirements, design robust systems, and implement functional and non-functional features.
- Developers:
  - Enhance technical proficiency by working with the Laravel framework and other technologies used in the project.
  - Collaborate effectively in a team environment, gaining experience in version control and coding standards adherence.
  - Acquire practical knowledge in system architecture, database design, and implementation of key functionalities.
- Administrators:

- Streamline the management of products, users, and transactions through an intuitive admin interface.
- Ensure data security and integrity with role-based access control and encryption mechanisms.
- Facilitate seamless operations through regular backups and fault tolerance measures.

- Sellers:
  - Leverage the platform to showcase products and reach a wider audience of potential buyers.
  - Simplify the process of product upload and approval, enhancing efficiency and convenience.
  - Benefit from promotional features and increased visibility, leading to potential sales growth.
- Buyers:
  - Enjoy a user-friendly interface with easy navigation and search functionality, improving the overall shopping experience.
  - Access a diverse range of products from local businesses and shops, promoting community engagement and support.
  - Engage in secure transactions with confidence, knowing that their data is protected and transactions are monitored.