Caltech | Center for Technology & Management Education

Live Session 02: Admin Dashboard Web Pages
DB Structure for Admin

Powerd by simplilearn

Caltech | Center for Technology & Management Education

Develop the Web Pages in Angular for Admin Dashboard

You Already Know

Before we begin, let's recall what we have covered till now:

- Angular 
- Cucumber 

Recap

Angular

It is a platform and framework by Google used to create single-page web applications using HTML and TypeScript.

Cucumber

It is a testing tool that supports the BDD framework.

Recap

Project Planning with Agile

- Developed user stories for admin dashboard epic
- Developed web app for end user epic
- Planned sprints in JIRA

Develop Angular Project Structures

- Created a Web App Project for End User using Angular CLI

Sync the Projects with Git on GitHub

- Pushed the Angular Projects on GitHub using Git

A Day in the Life of a Full Stack Developer

After syncing Angular projects on Github, Bob wants to design, develop, and create a database structure for his web page.

Let me think about this. Which technologies should I use for these requirements?

A Day in the Life of a Full Stack Developer

After brainstorming a bit, Bob found a solution for his requirements.

Let me use HTML and CSS to design and develop web pages and MySQL to design the database structure and create tables inside the database.

In this lesson, we will learn how to design, develop, and create a database structure for web pages to help Bob complete his task effectively and quickly.

Learning Objectives

By the end of this lesson, you will be able to:

- Develop the web page templates for admin dashboard
- Develop the CSS for styling the web pages
- Apply Angular component templates
- Correlate the pages in Angular with routing

Learning Objectives

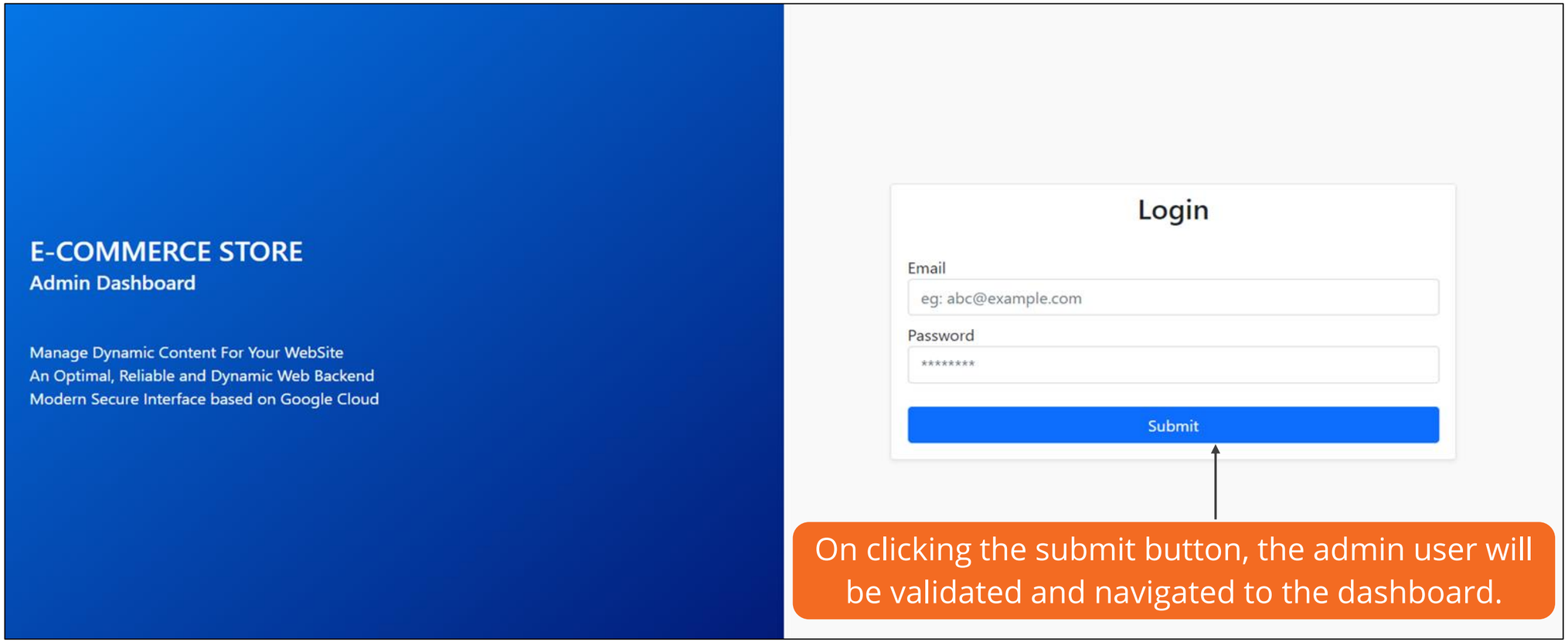
By the end of this lesson, you will be able to:

- Design a database for your project in MySQL
- Design a Web App Project for End-user using Angular CLI
- Work with constraints
- Create primary and foreign key relations within tables

Angular Component Templates

Web Page for Authentication Login Component

To begin, let’s develop a login page for admin to sign in and enter the Dashboard for various management flows.



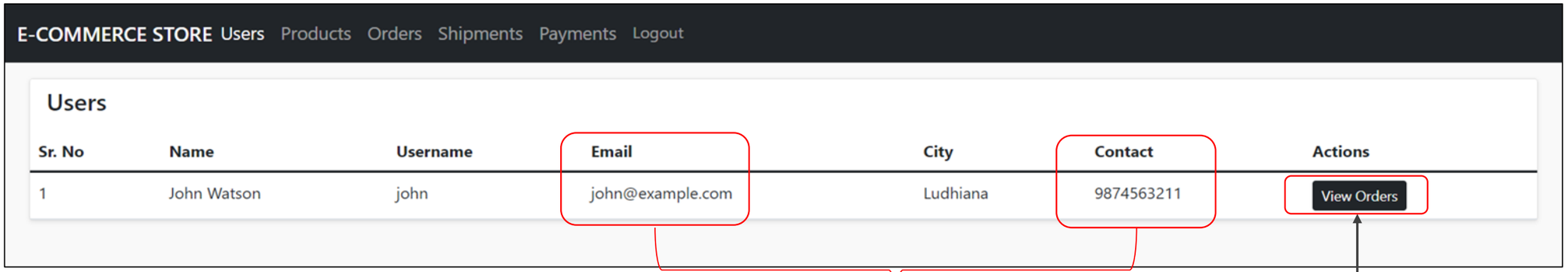
Web Page for Authentication Login Component

In the directory: src/app/pages/login/

Command	Use
login.component.css	CSS goes here to design the page and forms.
login.component.html	HTML Code is written in this template file.
login.component.ts	Logic will be written in typescript file.

Web Page for Users Component

On the Admin Dashboard, a navigation bar is designed to navigate between various modules. Users' component will have the list of all the users registered with the platform.



The search can be implemented through phone number or email ID.

The view order button shows the order history of the corresponding user.

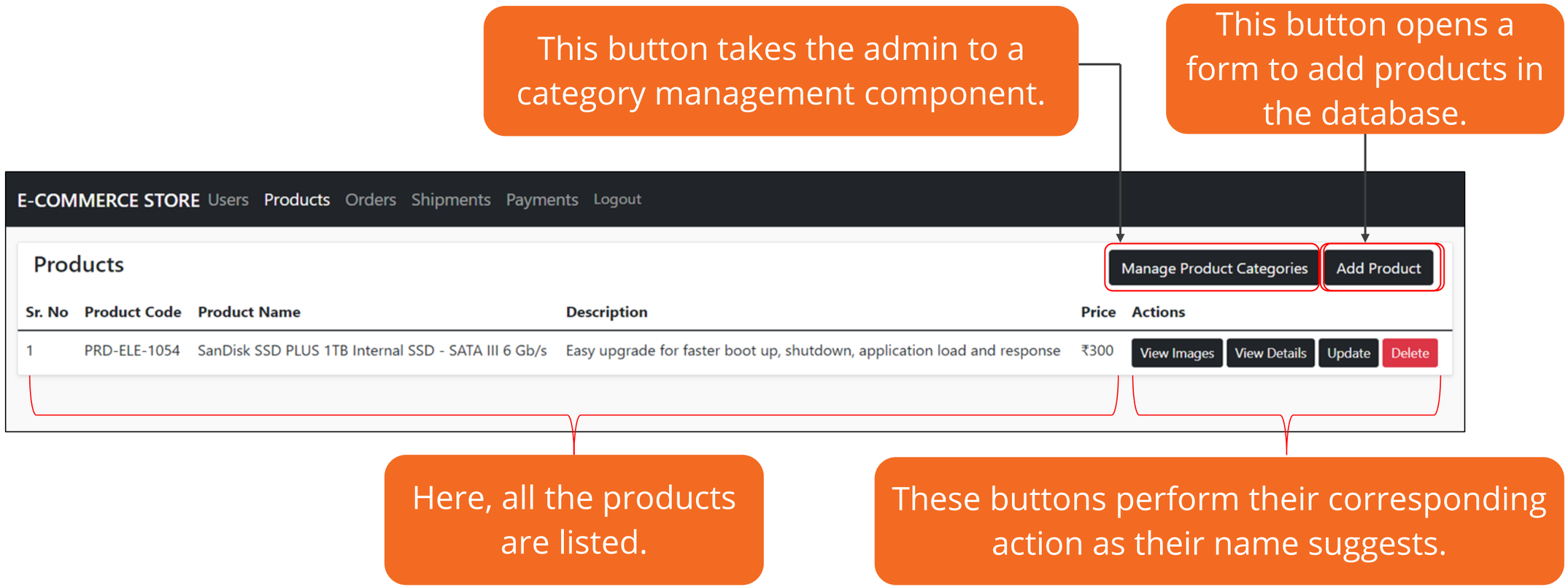
Web Page for Users Component

In the directory: src/app/pages/users/

Command	Use
users.component.css	CSS goes here to design the page and forms.
users.component.html	HTML Code is written in this template file.
users.component.ts	Logic will be written in typescript file.

Web Page for Products Component

To show the products along with categories, admin must add them from the dashboard.



Web Page for Products Component

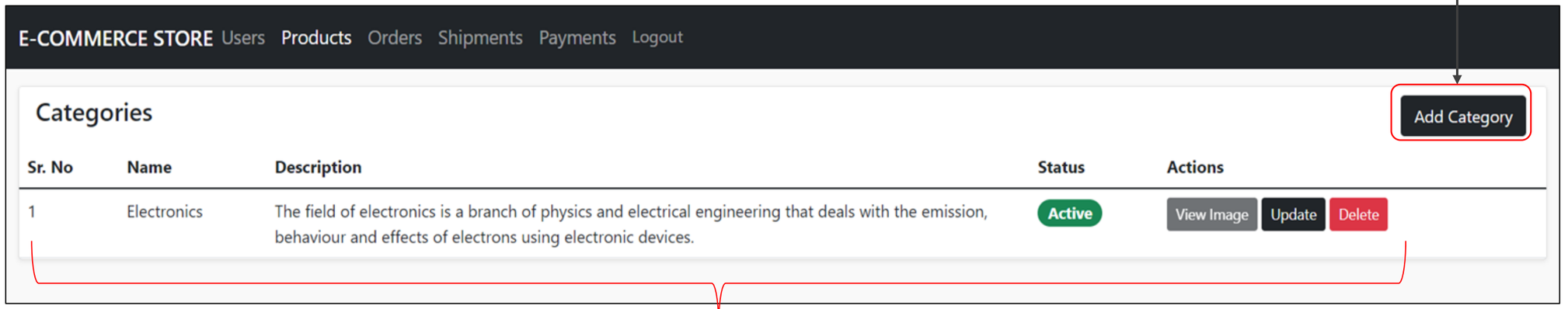
In the directory: src/app/pages/products/

Command	Use
products.component.css	CSS goes here to design the page and forms.
products.component.html	HTML Code is written in this template file.
products.component.ts	Logic will be written in typescript file.

Category Management Component

In this web page UI should be developed to manage the various categories for the products.

Add Category will open a modal to add the details in database.



This page lists all the categories, along with the action buttons to view, update, and delete the category.

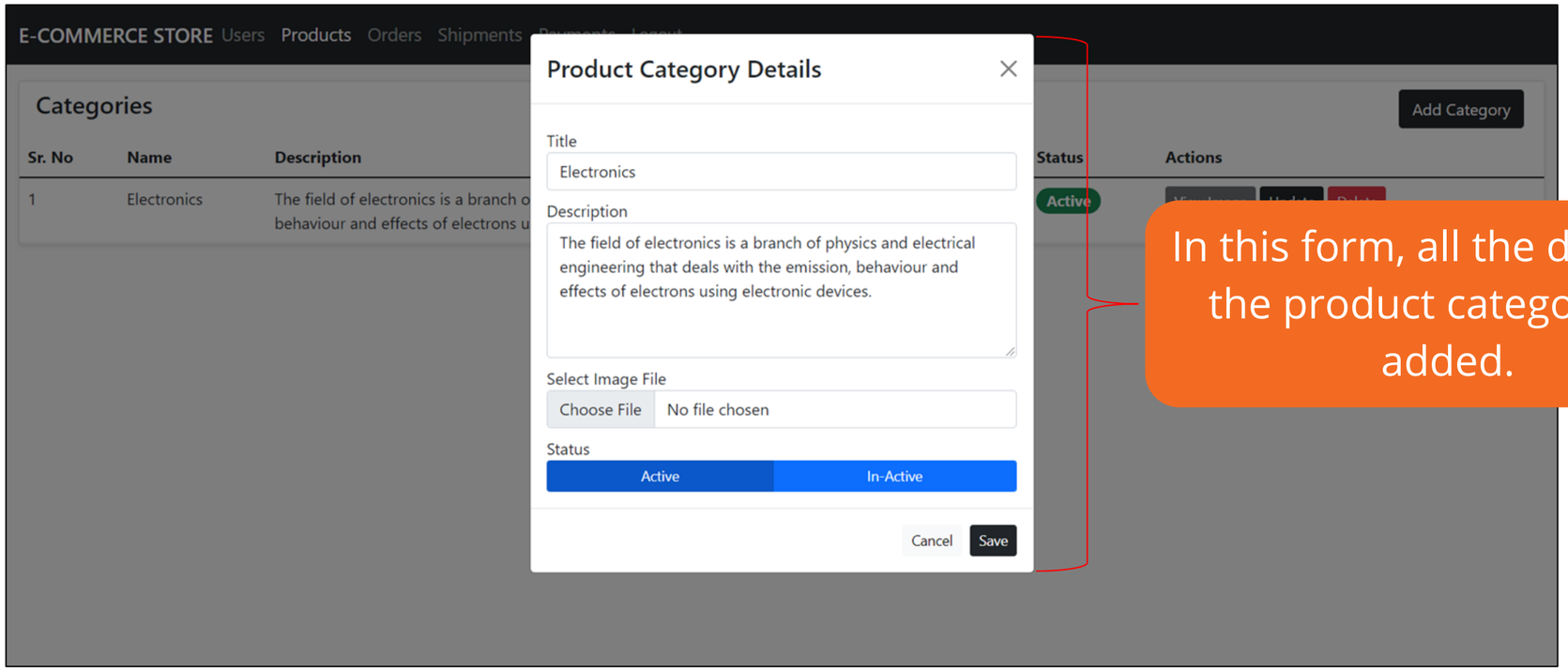
Category Management Component

In the directory: src/app/pages/categories/

Command	Use
categories.component.css	CSS goes here to design the page and forms.
categories.component.html	HTML Code is written in this template file.
categories.component.ts	Logic will be written in typescript file.

Category Management Component: Add Category Modal

- Use NgbModal service to develop the modal and add the category details.
- Create the modal for Product Category Details in the directory src/app/pages/modals/category.ts



Web Page for Product Component: Add Product Modal

On the Products Component Web Page, the Action Button is used to add the product in the database.

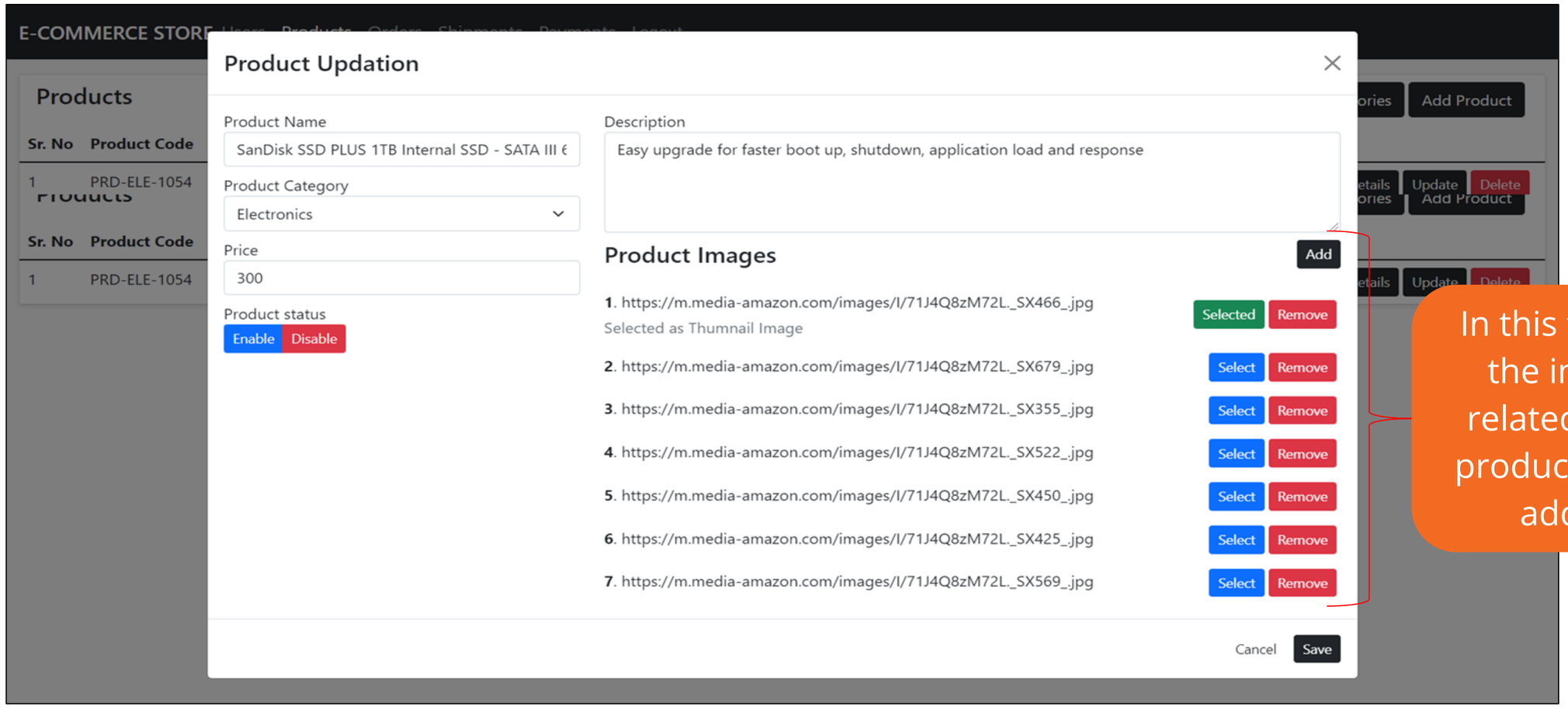
On clicking, it will open a modal, which typically is a UI in a dialog view. Here, user can add the details of a product in a form.

User can also select the category from a dropdown to be linked to the product.

Here, user can associate as many image as they wish to link to the product and these images must be shown as a thumbnail that can be managed.

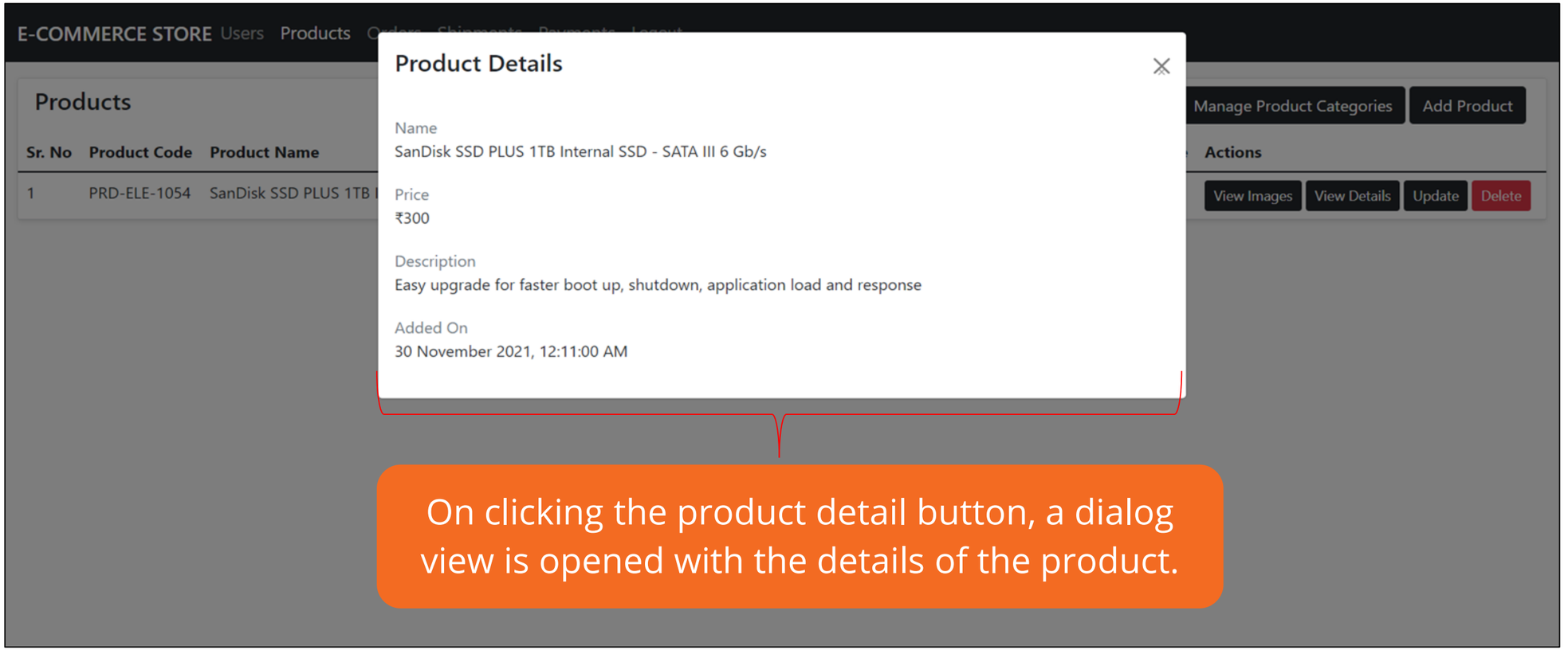
Web Page for Product Component: Add Product Modal

Given below is an image of the product update page:



Web Page for Product Component: Product Details Modal

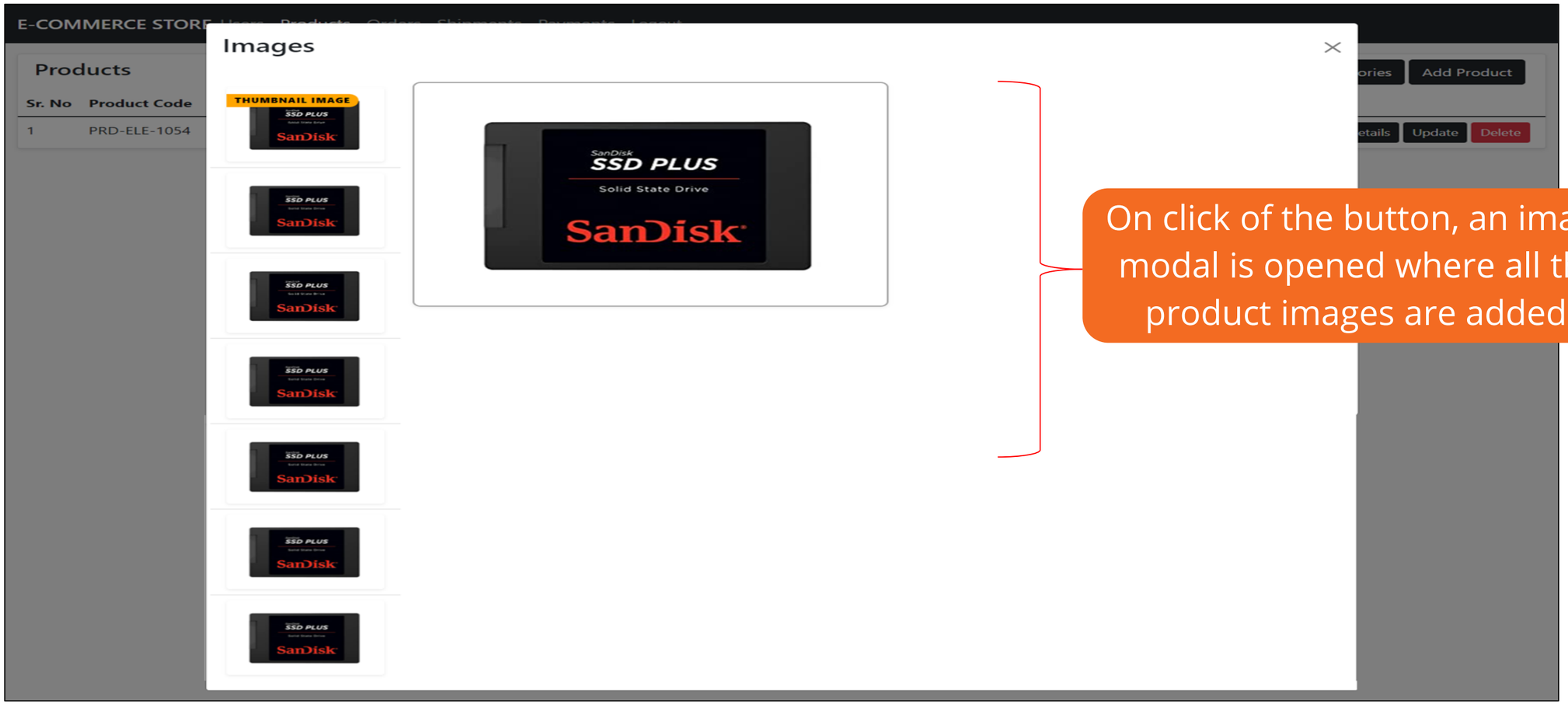
Create the modal for Adding Product Detail
In the directory: src/app/pages/modals/products.ts



On clicking the product detail button, a dialog view is opened with the details of the product.

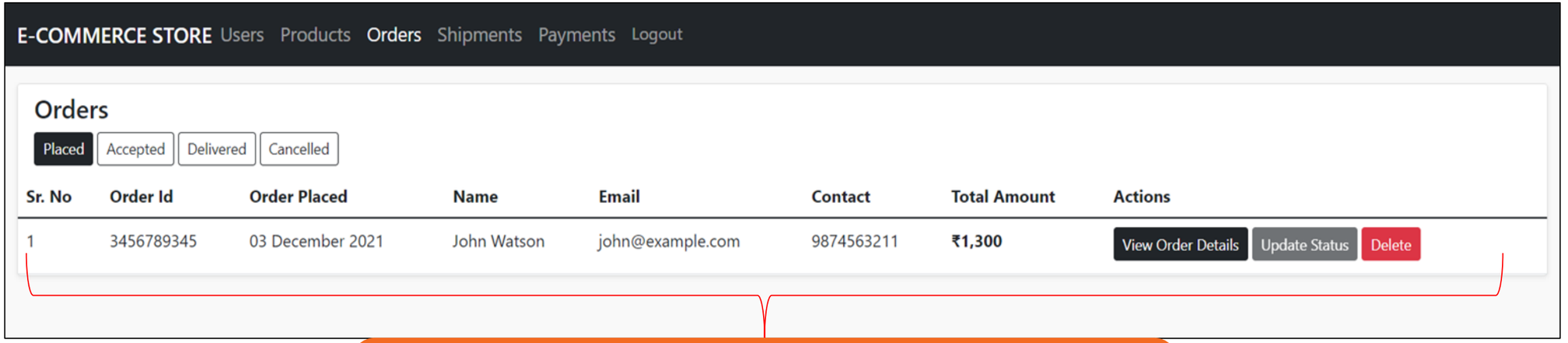
Web Page for Product Component: Product Images Modal

An UI element is added as a button in the product component to see the product images.



Web Page for Orders Component: View Orders

In the End User Web App, users will place order from their account. The same list of orders will be displayed in the Admin Panel for various users.



In this form, the order status can be modified with the help of update status button.

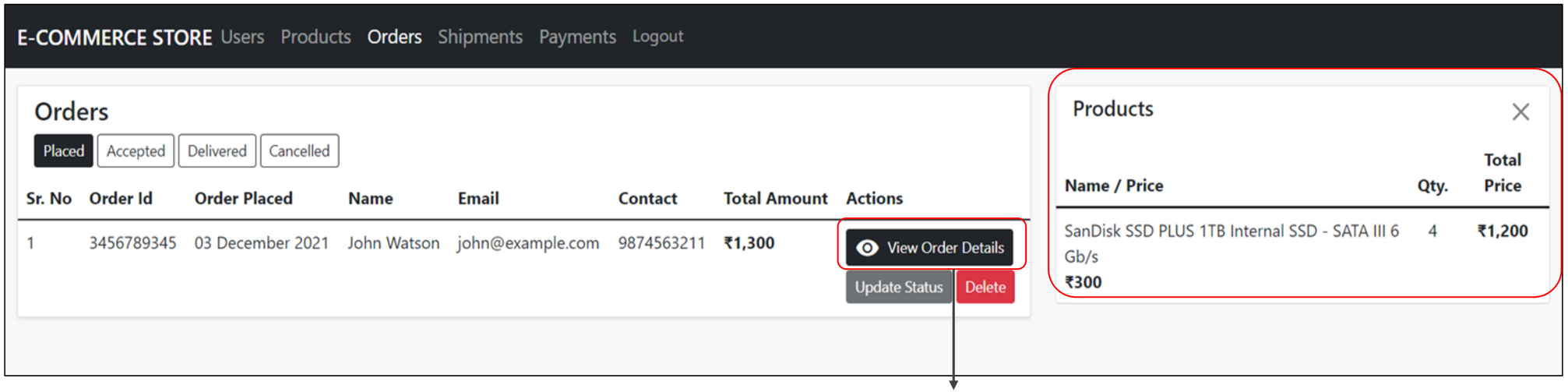
Web Page for Orders Component: View Orders

In the directory: src/app/pages/orders/

Command	Use
orders.component.css	CSS goes here to design the page and forms.
orders.component.html	HTML Code is written in this template file.
orders.component.ts	Logic will be written in typescript file.

Web Page for Orders Component: View Products in an Order

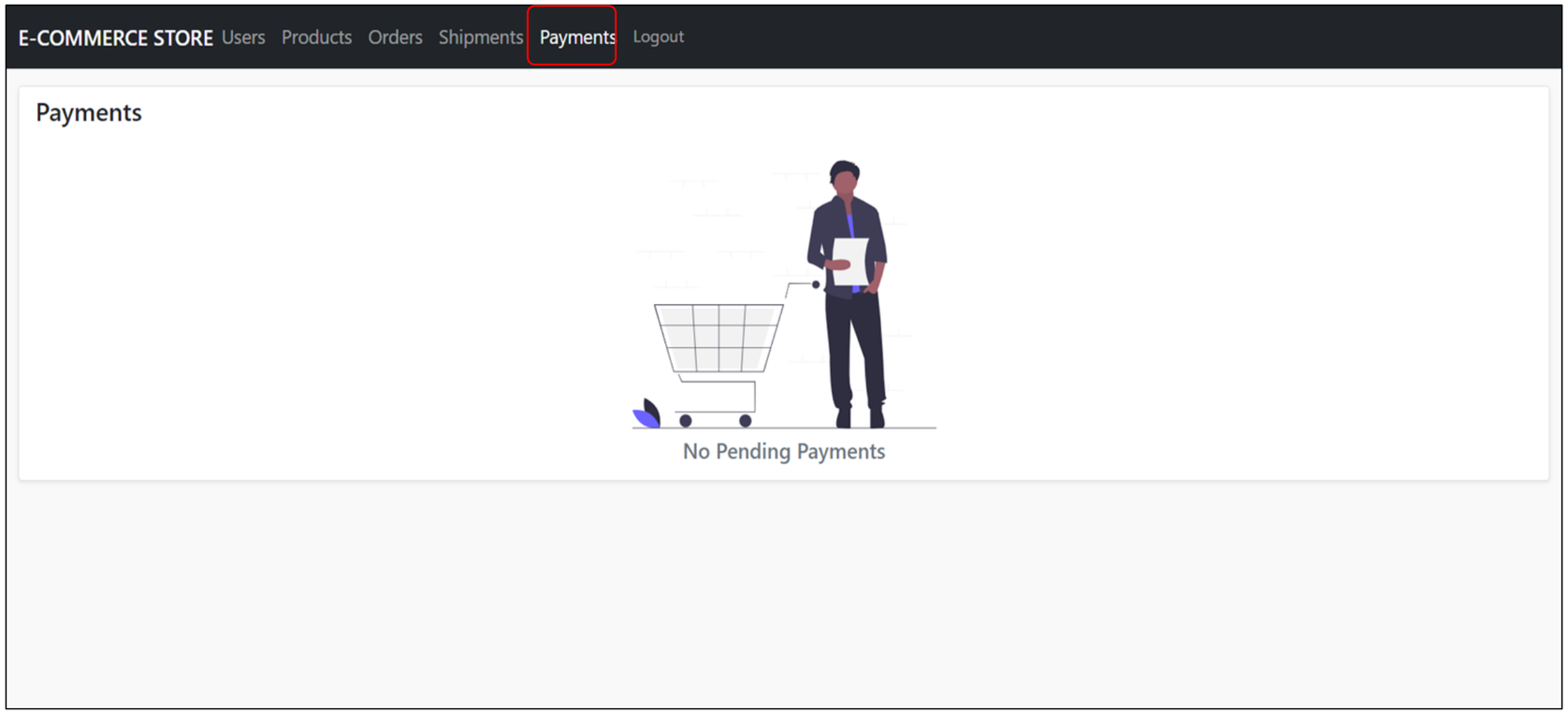
An action button on the orders page is View Order Details.



By clicking on it, the details of an order, including the product list and total amount get displayed.

Web Page for Payments Component

Develop Payments Page to check the payment methods and status of the payments or transactions made by the End Users for the Orders.



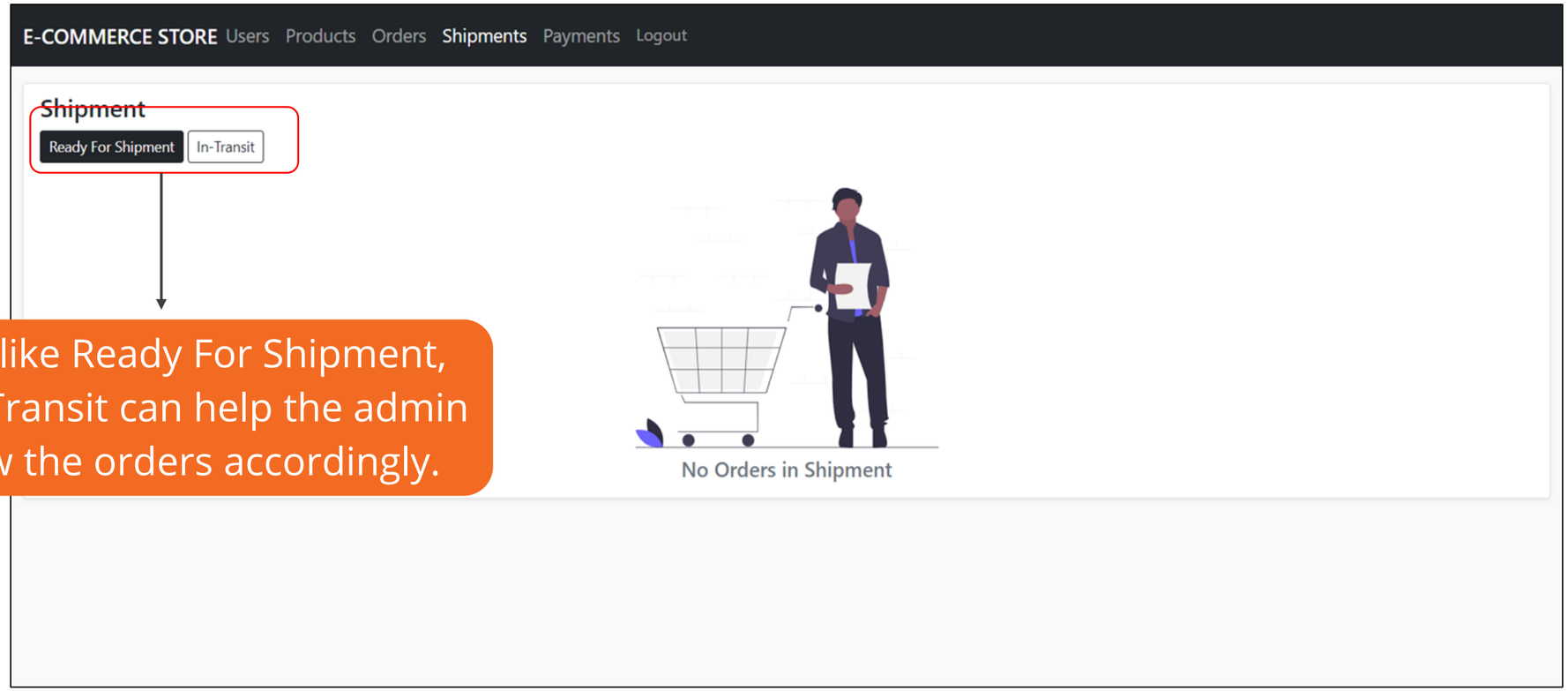
Web Page for Payments Component

In the directory: src/app/pages/payments/

Command	Use
payments.component.css	CSS goes here to design the page and forms.
payments.component.html	HTML Code is written in this template file.
payments.component.ts	Logic will be written in typescript file.

Web Page for Shipment Component

The shipments page helps the Admin to track the orders that are ready for shipment and in transit.



Filters like Ready For Shipment, and In-Transit can help the admin to view the orders accordingly.

Web Page for Shipment Component

In the directory: src/app/pages/shipments/

Command	Use
shipments.component.css	CSS goes here to design the page and forms.
shipments.component.html	HTML Code is written in this template file.
shipments.component.ts	Logic will be written in typescript file.

Caltech | Center for Technology & Management Education

DB Structure for Admin

Powered by simplilearn

Creating Database

Creating Database in MySQL

In MySQL, CLI uses these commands to create and work with the database:

Create database e-commerce

This command creates the database e-commerce.

Use database e-commerce

This command changes the current working selection of the database to e-commerce.

Show tables

This command will list all the tables in the database. It will return an empty set if no table is available.

Creating Tables

Prerequisites

- Make sure the database is selected as e-commerce.
- User can execute the command to select db for e-commerce as:

```
use database ecommerce;
```

Generating Admin's Table for Login

In order to create a table for admin, use columns like admin Id, email, and password.

```
CREATE TABLE ADMINS
  adminId               INTEGER NOT NULL PRIMARY KEY
AUTO_INCREMENT,
  email                 VARCHAR(50) NOT NULL,
  password              VARCHAR(50) NOT NULL,
  fullName              VARCHAR(255) NOT NULL,
  loginType             INTEGER DEFAULT 1,
  addedOn               DATETIME DEFAULT CURRENT_TIMESTAMP
```

Generating User's Table

In order to create the table for users, use columns like user id, email, password, etc.

```
CREATE TABLE USERS(
  userId                INTEGER NOT NULL PRIMARY KEY
AUTO_INCREMENT,
  email                 VARCHAR(50) NOT NULL,
  password              VARCHAR(50) NOT NULL,
  fullName              VARCHAR(255) NOT NULL,
  street                VARCHAR(50) DEFAULT NULL,
  city                  VARCHAR(50) DEFAULT NULL,
  state                 VARCHAR(50) DEFAULT NULL,
  country               VARCHAR(50) DEFAULT NULL,
  pincode               INTEGER,
  image                 VARCHAR(1000),
  contact               BIGINT,
  addedOn               DATETIME DEFAULT CURRENT_TIMESTAMP
```

Generating Product Categories Table

In order to create a table for categories, use columns like category id, category Name, etc.

```
CREATE TABLE CATEGORIES (
  categoryId            INTEGER NOT NULL PRIMARY KEY
AUTO_INCREMENT,
  categoryName          VARCHAR(255) NOT NULL,
  categoryDescription   VARCHAR(255),
  categoryImageUrl      VARCHAR(500),
  active                INTEGER DEFAULT 0,
  addedOn               DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Generating Product's Table

In order to create table for the product, use columns like product id, product title, etc.

```
CREATE TABLE PRODUCTS(
  productId             INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  productTitle          VARCHAR(500) NOT NULL,
  productDescription    VARCHAR(500) NOT NULL,
  productCode           VARCHAR(500) NOT NULL,
  categoryId            INTEGER,
  images                VARCHAR(1000),
  thumbnailImage        INTEGER DEFAULT 0,
  price                 INTEGER DEFAULT 0,
  addedOn               DATETIME DEFAULT CURRENT_TIMESTAMP,
  rating                INTEGER NOT NULL,
  FOREIGN KEY (categoryId) REFERENCES CATEGORIES(categoryId)
);
```

Generating Orders' Table

In order to create table for orders, use columns like order id, order date, etc.

```
CREATE TABLE ORDERS(
  orderId               INTEGER NOT NULL PRIMARY KEY,
  orderDate             DATETIME DEFAULT CURRENT_TIMESTAMP,
  orderStatus           VARCHAR(50) NOT NULL,
  totalItems            INTEGER NOT NULL,
  itemsSubTotal         INTEGER NOT NULL,
  shipmentCharges       INTEGER NOT NULL,
  totalAmount           INTEGER NOT NULL,
  paymentStatus         INTEGER DEFAULT 0,
  paymentStatusTitle    VARCHAR(255),
  paymentMethod         INTEGER,
  paymentMethodTitle    VARCHAR(255) NOT NULL,
  userId                INTEGER NOT NULL,
  name                  VARCHAR(255) NOT NULL,
  email                 VARCHAR(255) NOT NULL,
  contact               BIGINT NOT NULL,
  address               VARCHAR(500) NOT NULL,
  FOREIGN KEY (userId) REFERENCES USERS(userId)
);
```

Generating Shipments' Table

In order to create table for shipments, use columns like shipment id, order id, etc.

```
CREATE TABLE SHIPMENTS(
  shipmentId           INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  orderId              INTEGER,
  shipmentStatus       INTEGER,
  shipmentTitle        VARCHAR(255),
  shipmentDate         DATETIME DEFAULT CURRENT_TIMESTAMP,
  expectedDeliveryDate DATETIME,
  shipmentMethod       VARCHAR(255),
  shipmentCompany      VARCHAR(255),
  FOREIGN KEY (orderId) REFERENCES ORDERS(orderId)
);
```

Generating Order Items' Table

In order to create table for order items, use columns like order item id, order id, etc.

```
CREATE TABLE ORDERITEMS(
  orderItemId          INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
  orderId              INTEGER,
  productId            INTEGER,
  productCode          VARCHAR(255) NOT NULL,
  productImg           VARCHAR(255) NOT NULL,
  productTitle         VARCHAR(255) NOT NULL,
  productDescription   VARCHAR(255) NOT NULL,
  productCategory      VARCHAR(255) NOT NULL,
  price                INTEGER NOT NULL,
  quantity             INTEGER NOT NULL,
  totalPrice           INTEGER NOT NULL,
  FOREIGN KEY (orderId) REFERENCES ORDERS(orderId),
  FOREIGN KEY (productId) REFERENCES PRODUCTS(productId)
);
```

Key Takeaways

- HTML and CSS are used for the development of Admin Dashboard.
- Angular templates are used for the web page development.
- Various Angular CLI commands such as ng serve are used to view the web app.
- UI is developed on the web page for admin users to manage the various categories for the products.

Key Takeaways

- The Action Button in the products component web page adds the product to the database.
- In the end-user web app, users can place an order from their account, and the same list of orders is displayed in the admin panel for updating order status by an admin user.
- The shipments page helps the Admin track the orders that are ready for shipment and in transit.
- Database is selected as e-commerce to create tables.

Key Takeaways

- In MySQL, CLI uses commands to create and work with the database.
- The **create database e-commerce** command is used to create e-commerce database.
- The **use database e-commerce** command is used to select the e-commerce database as working database.
- The **show tables** command is used to view the list of all the tables in the database.

Before the Next Class

Since you have completed this session successfully, for our next discussion you should:

- Review the OOPS with Java
- Review the JDBC Fundamentals
- Revise Servlet API
- Review how to use HTTP Client API in Angular
- Explore and work with JSON
- Check working conditions of Eclipse EE edition

What's Next?

Now, we have finished our DB structure for the Admin dashboard. In our next discussion we will:

- Design and develop web pages for the end users web app project
- Work on Authentication Page, Users Profile Page, Orders Page, and others to create the UI for the project
- Work on deciding the various attributes for the tables in MySQL
- Create the structure of tables in MySQL which will store data from the web pages
- Create dynamic web project using Enterprise Edition of Java in Eclipse
- Create Servlets and JDBC connections