

MỤC TIÊU:

Kết thúc bài thực hành này bạn có khả năng

- ✓ Cài đặt và sử dụng mySql trong nodejs
- ✓ Truy vấn dữ liệu mySql với nodejs
- ✓ Thao tác trên mySQL

PHẦN I

Bài 1 (1 điểm)

Thiết kế cơ sở dữ liệu và giao diện layout cho website

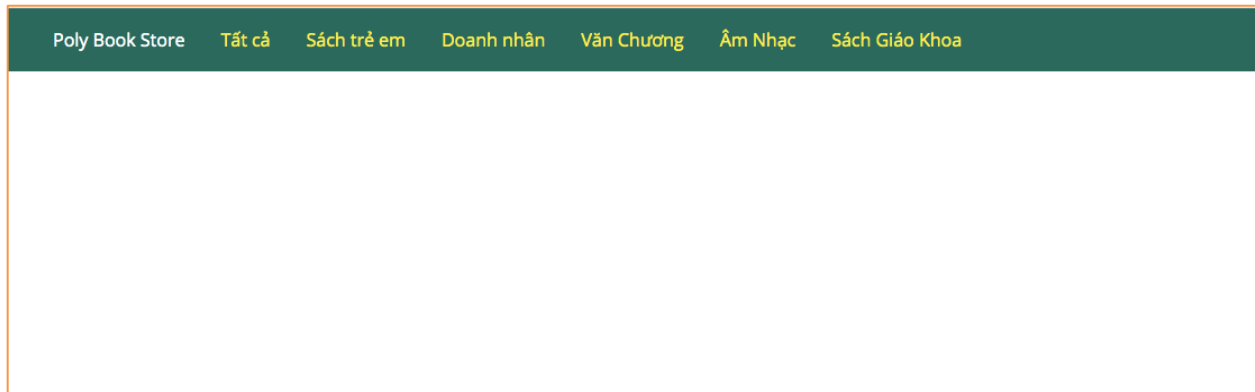
1. Thiết kế cơ sở dữ liệu

Book catalog	
🔑	idCategory : int(11)
📄	nameCategory : varchar(200)
📅	dateCategory : timestamp
📄	dateEditCategory : varchar(50)

Book products	
🔑	idProduct : int(11)
📄	nameProduct : varchar(200)
📄	sortDescription : varchar(300)
📄	images : varchar(200)
📄	description : varchar(20000)
📅	dateProduct : timestamp
#	priceProduct : double
📄	authorProduct : varchar(300)
#	idCategory : int(11)

Book tbluser	
🔑	username : varchar(50)
📄	password : varchar(300)
#	rule : int(11)

2. Thiết kế giao diện layout

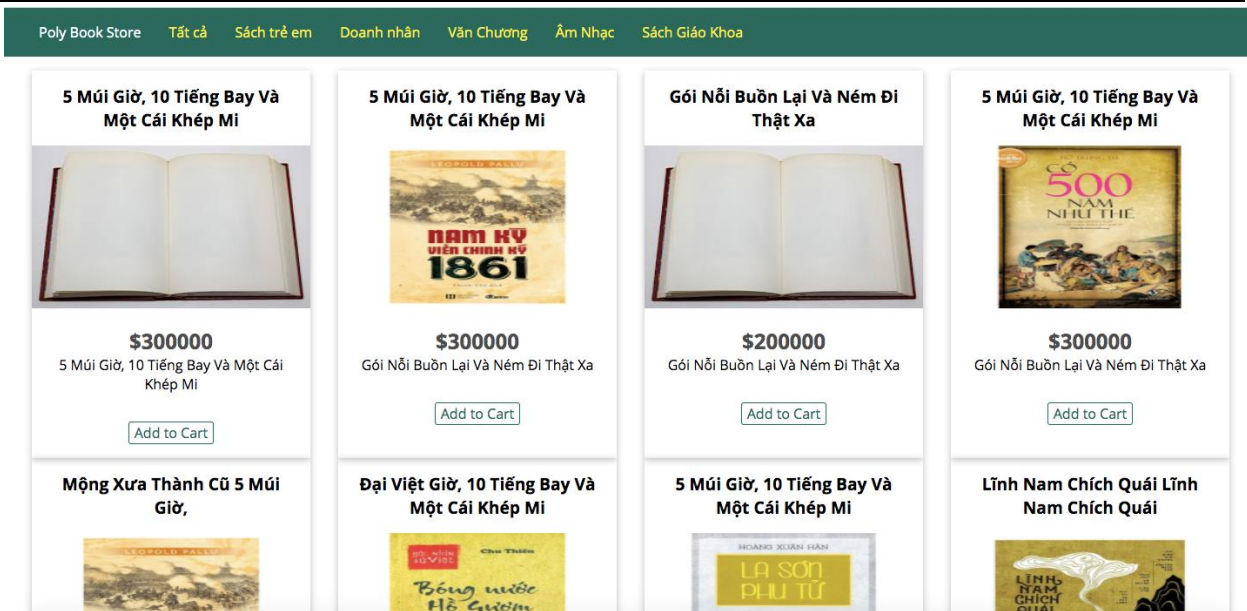


3. Tổ chức code

```
<body>
  <header class="main-header">
    <nav class="main-header__nav">
      <ul class="main-header__item-list">
        <li class="main-header__item"><a href="/">Poly Book Store</a></li>
        <% for(let cate of catalogs){ %>
          <li class="main-header__item"><a class="active" href="/shop"><%=
cate.nameCategory %></a></li>
        <%}%>
      </ul>
    </nav>
  </header>
  <main align="center">
    <!--main content-->
  </main>
</body>
```

Bài 2 (2 điểm)

Xây dựng trang Home->All



Bước 1: Thực hiện route để điều hướng và đọc dữ liệu cho trang home

```
app.get("/",(req,res)=>{
  let sql = `SELECT * FROM products;select * from catalog`;
  //let sql="select * from catalog";
  db.query(sql, function(err, data) {
    if (err) throw err;
    res.render('shop',{catalogs:data[1],products:data[0]});
  })
})
```

Bước 2: Hiện thị danh sách sản phẩm ra giao diện

```
<main align="center">
  <div class="grid">
    <% for (let product of products) { %>
      <article class="card product-item">
        <header class="card__header">
          <h1 class="product__title"><%= product.nameProduct %></h1>
        </header>
        <div class="card__image">
          
        </div>
        <div class="card__content">
          <h2 class="product__price">$<%= product.priceProduct %></h2>
          <p class="product__description"><%=product.sortDescription %></p>
        </div>
        <div class="card__actions">
          <button class="btn">Add to Cart</button>
        </div>
      </article>
    <% } %>
  </div>
```

</main>

Bài 3 (2 điểm)

Trang sách theo loại. Khi người dùng chọn loại sách trên menu thì website hiển thị sách tương ứng với loại sách được chọn

Bước 1: Cập nhật liên kết của menu

```
<ul class="main-header__item-list">
  <li class="main-header__item"><a href="/shop">Poly Book Store</a></li>
  <% for(let cate of catalogs){ %>
    <li class="main-header__item"><a class="active" href="/shop/<%= cate.idCategory
%>"><%= cate.nameCategory %></a></li>
  <%}%>
</ul>
```

Bước 1: route cho trường hợp /shop/id

//sản phẩm theo loại

```
app.get("/shop/:catId",(req,res)=>{
  let catId=req.params.catId;
  let sql = `SELECT * FROM products where idCategory=${catId};select * from catalog`;
  //let sql="select * from catalog";
  db.query(sql, function(err, data) {
    if (err) throw err;
    res.render('shop',{catalogs:data[1],products:data[0]});
  })
})
```

PHẦN II

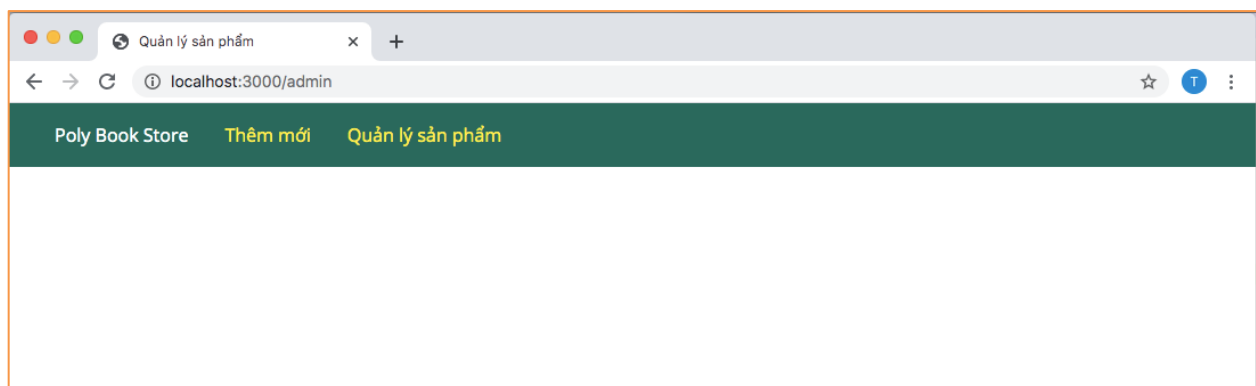
Bài 4 (2 điểm)

Xây dựng trang thêm sản phẩm

Bước 1: Cài đặt module hỗ trợ upload ảnh

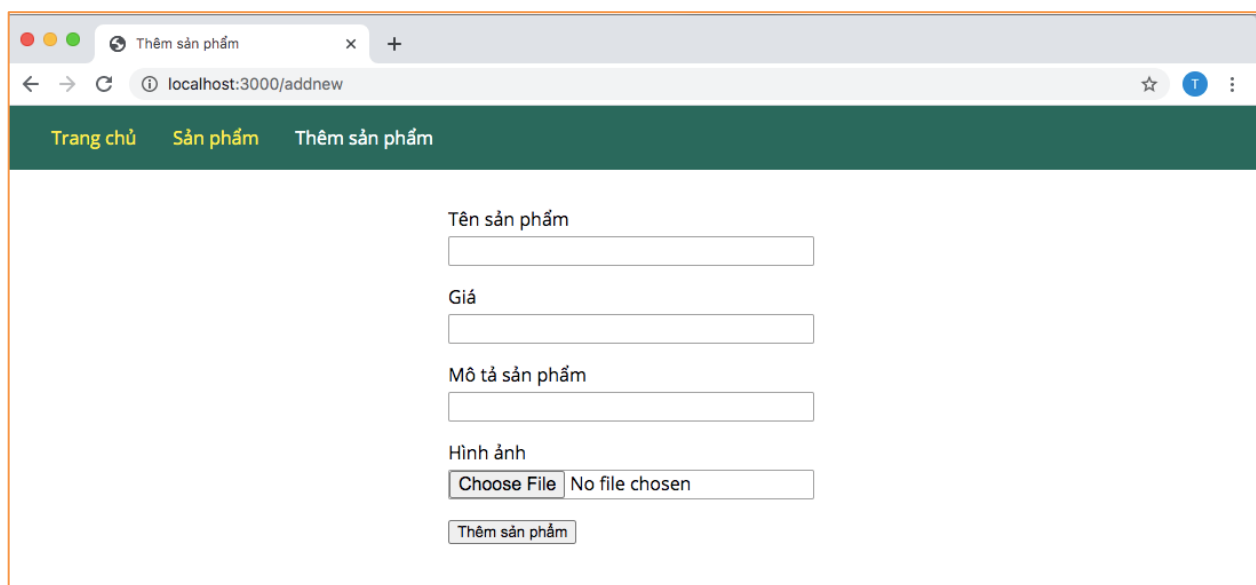
`Npm install multer`

Bước 2: Tạo route để chuyển sang phân quản trị sản phẩm (localhost:3000/admin)



Hướng dẫn: sinh viên tạo thư mục admin trong view, chứa các view home, addnew, list,...)

Bước 3: Trang thêm mới – khi chọn menu thêm mới:



Hướng dẫn:

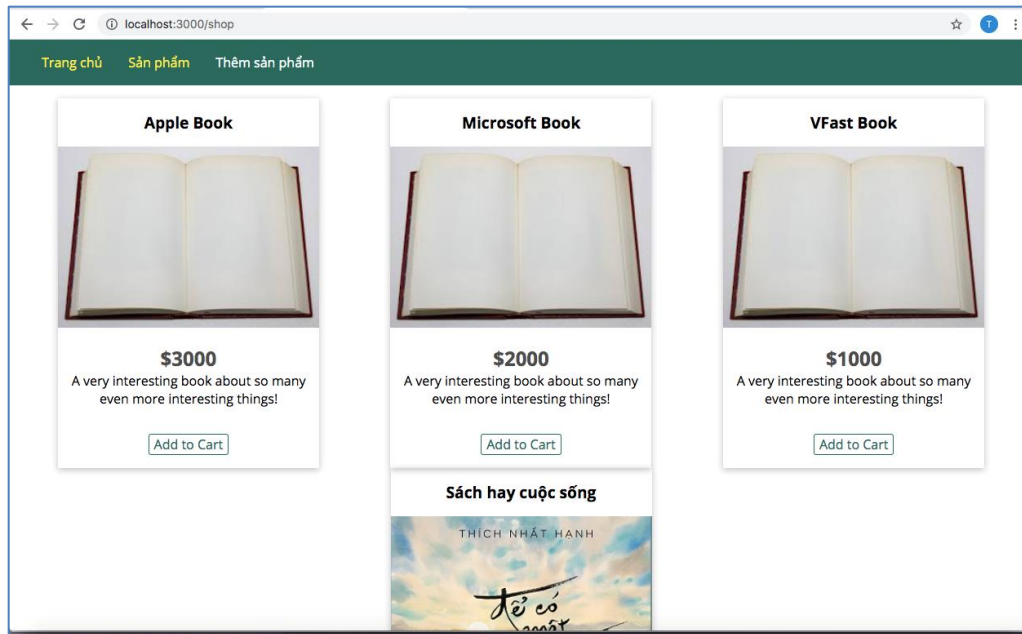
```
//quản lý sản phẩm
```

```
app.get("/admin", (req, res) => {
  res.render("admin/home");
})
app.get("/addnew", (req, res) => {
  res.render("add-product");
})

app.post('/addnew', upload.single('productImage'), (req, res) => {
  const file = req.file
  let title = req.body.productName;
  let price = req.body.price;
  let description = req.body.description;
  let nameImage = file.filename;
  product = {
    nameProduct: title,
    priceProduct: price,
    sortDescription: description,
    images: nameImage,
  }
  db.query('insert into products SET ?', product, function(err, data) {
    if (err) throw err;
    res.redirect('/shop');
  })
});
```

- Kết quả khi thêm một sản phẩm mới

The screenshot shows a web browser window with the address bar displaying 'localhost:3000/addnew'. The page has a dark green header with three navigation links: 'Trang chủ', 'Sản phẩm', and 'Thêm sản phẩm'. The main content area contains a form for adding a new product. The form has four input fields: 'Tên sản phẩm' (Product Name) with the value 'Sách hay cuộc sống', 'Giá' (Price) with the value '23000', 'Mô tả sản phẩm' (Product Description) with the value 'Là một cuốn sách hay cho mọi nhà', and 'Hình ảnh' (Image) with a file upload button labeled 'Choose File' and the filename 'sach.png'. At the bottom of the form is a button labeled 'Thêm sản phẩm'.



Bài 5 (3 điểm)

Sinh viên thực hiện xây dựng trang quản lý sản phẩm: Hiển thị danh sách sản phẩm dạng grid với các chức năng xoá, sửa.

*** Yêu cầu nộp bài:

SV nén file (hoặc share thư mục google drive) bao gồm các yêu cầu đã thực hiện trên, nộp LMS đúng thời gian quy định của giảng viên. KHÔNG NỘP BÀI COI NHƯ KHÔNG CÓ ĐIỂM.

--- Hết ---