COMS 3190 - Final Project Proposal

by Asray Gopa and Emre Okur

Introduction

Team Members:

- **Emre Okur**
  - ISU Email: emreokur@iastate.edu
  - Background: I am a junior studying Computer Science. I have experience in frontend development (HTML/CSS/JavaScript) and UI/UX design. My strengths lie in creating responsive layouts and the interactive elements of a page on the client-side.
- **Asray Gopa**
  - *ISU Email*: asrayg@iastate.edu
  - *Background*: I am a junior studying Computer Science. I am skilled in the logic associated with JavaScript and features, dynamic on the client-side. I am strong in algorithms and data manipulation.

Our midterm project was a camera store website. As our final project, we selected Option 1: Extend Midterm Project. We are going to convert the camera store website into a full-stack web application. The static camera store website assessed the real-world problem of a small business needing dynamic and scalable e-commerce websites with user accounts, stored data, and security when students check out.

Purpose of the Proposal

Objective:

Our goal is to design a comprehensive e-commerce platform for photography enthusiasts. Users will have the ability to explore products, leave product reviews, and make hypothetical purchases.

Importance:

People use static websites that fail to include functionalities such as user sign in, current data update, and storage of data reliably. The addition of the backend and modern frameworks will illustrate how a small business can scale their digital presence at a low cost.

Goals & Objectives

Goals:

- Develop a full-stack e-commerce platform using React, Node.js, and MongoDB.
- Implement end-to-end features with full ownership (frontend + backend) per team member.
- Align deliverables with the four mini-assignments to ensure incremental progress and adherence to deadlines.

Objectives:

Mini-Assignment 1: Getting Started

- Planning & GitLab Structure (Week 1)Complete wireframes and/ or screen sketches with Figma.
- Create GitLab directory structure in a repository with folders frontend/, backend/, Documents/.
- Submit a screenshot of the repo directory structure and a screenshot of our planning documents/wireframes, etc.

Mini-Assignment 2: Minimal UI & Routing (Week 2)

- Get a React frontend with react router working.
- Add a Home Page (with at least a hero section and navigation) and a simple Product Listing Page (with static grid).
- Submit screenshots of our UI, and explanations of your routing e.g. /products, /login, etc.

Mini-Assignment 3: Backend Integration (Week 3)

- Write an Express.js backend that connects to a MongoDB Database (with Mongoose schemas).
- Seed the database with product data; create API endpoints e.g. GET /api/products.
- Demonstrate our frontend- backend integration with Axios from the frontend, e.g. retrieving products from the API.
- Submit a screen capture of a working call to the API (request/ response.)

Mini-Assignment 4: Polishing the App (Week 4)

- Add authentication (via JWT/Bcrypt) as the advanced feature.
- Ensure the app is fully responsive across device sizes and that UI/UX is close to completion.
- Document the Software Architecture Document (SAD) and Final Report.

Project Description

1. Home page: Hero section, intuitive navigation bar, featured products.
2. Authentication pages(Login / Sign up): Secure login and sign-up features using Mongo and tokens for session validation.
3. Product listings: Searching the products the store has, and filtering options to find a product faster.
4. Product details: Details about the product: price, stock, reviews, etc.
5. Cart Management: Changing the amount of product in the cart(add/remove), proceeding to a checkout to complete the purchase.
6. Product Reviews: Review posts from the users, including a number rating and a text to express their opinions on the product.

Project Path Selection

Option 1: Expand upon our Midterm Project

New Functionality:

- User authentication, the ability to keep cart data persistent, and a user review system.
- Dynamic product filtering, now implemented via API calls.

Redevelop:

- Our midterm project's HTML/CSS/JS code would be rebuilt into React components.
- Our back-end APIs would dynamically replace static JSON data.

Justification: This approach allows us to modernize the midterm's static design while learning full-stack integration.

Feature Ownership & Responsibility

Feature	Assignee	Tech Stack
User Authentication	Emre Okur	React (Frontend) + Node.js (Backend)
Confirmation Page	Asray Gopa	React + Node.js/MongoDB
Home/Landing Page	Emre Okur	React + Node.js/MongoDB
About Page	Asray Gopa	React + Node.js/MongoDB
Product Listings & Details	Emre Okur	React + Node.js/MongoDB
Purchase Confirmation Page	Asray Gopa	React + Node.js/MongoDB
Cart Page	Emre Okur	React + Node.js/MongoDB
Order Management Page(s)	Asray Gopa	React + Node.js/MongoDB
Not Found Page	Emre Okur	React + Node.js/MongoDB

Resources and Tools

Frontend: React, Axios, React Router.
Backend: Node.js, MongoDB, Mongoose, JWT.
Tools: GitLab (version control), Figma (wireframing), Postman (API testing).
APIs: Custom RESTful APIs for product and user data.
Time Commitment: 8–10 hours/week per member.

File Structure and Project Organization

frontend/
  src/
    assets/ – Product images, logos
    components/ – Navbar, ProductCard, ReviewForm, ContactForm
    pages/ – Home, ProductListing, ProductDetail, Cart, Success, MyAccount, MyOrders
backend/
  models/ – Product, User, Review
  routes/ – productRoutes, authRoutes, reviewRoutes
  server.js
Documents/ – Wireframes, SAD, Final Report

Data Sources and Management

Data Source: The product listings will be provided by the admin/owner. These data will be stored using MongoDB. User data will be collected via signup. Product review will be provided by the users.

Sample Product Schema:

```
{
  "productId": 1,
  "name": "Canon EOS R5",
  "price": 3499,
  "category": "Mirrorless",
  "specs": { "sensor": "45MP", "iso": "100-51200" }
}
```

Sample User Schema:

```
{
  “userId”: 1,
  "name": "Emre Okur",
  "email": “emreokur@iastate.edu”,
  "password": "hashed-pwd-here",
  "DateOfBirth": 04-09-2004
}
```

Sample UserReview Schema:

```
{
  "userId": "1",
  "title": “Title Here”,
  "text": "Text body here.",
  "numberRating": 3.0
}
```

Sample Order Schema:

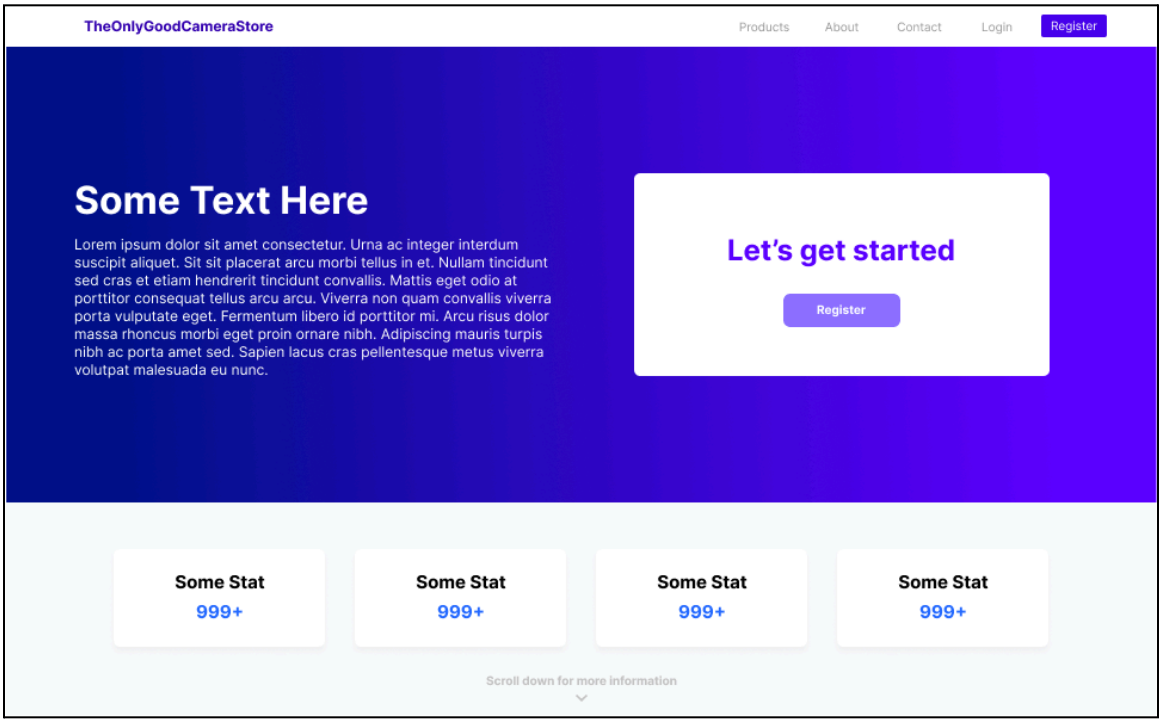
```
{
  "orderId": 1,
  "userId": 1,
  "productId": 1,
  "Status": “Shipped”
}
```

CRUD Operations:

The users will be able to create, read, update, and delete their account information.
The users will be able to create, read, update, and delete their product reviews.
The admins will be able to create, read, update, and delete product listings.
The admins will be able to create, read, update, and delete user accounts as they wish.
The admins will be able to create, read, update, and delete orders.

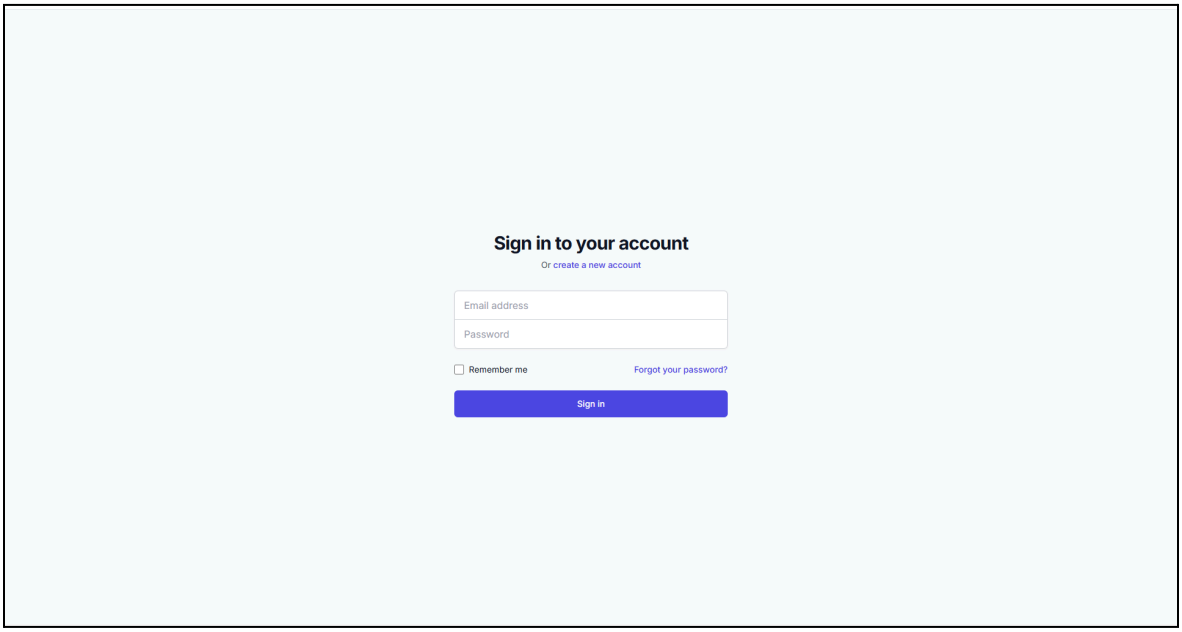
User Experience Views

Home Page:



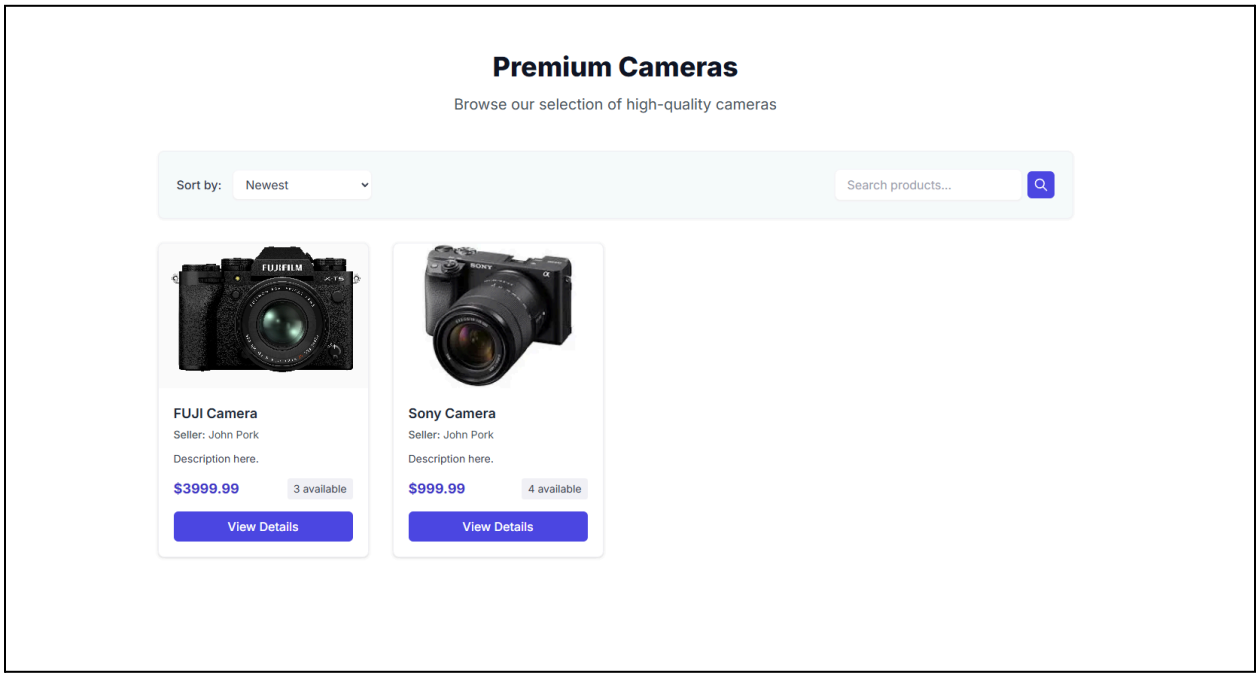
This will be the landing page. Users can see what our project is about. From this page, they can go to the login and register pages. They can also go to our About and Contact pages.

Authentication Page:



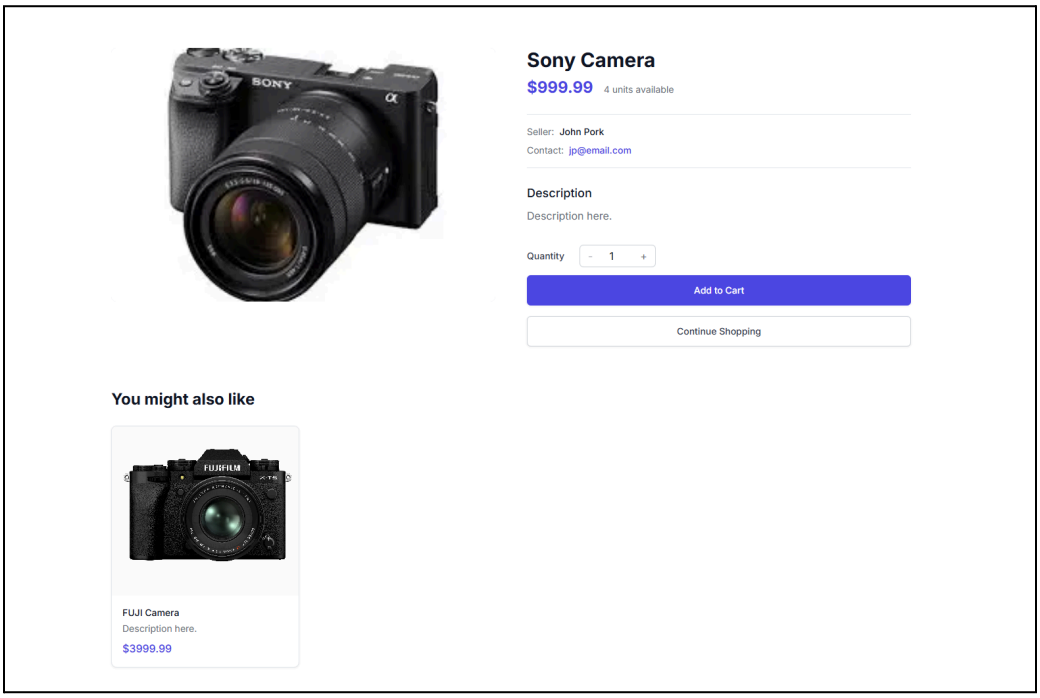
This will be the page where users will sign in / sign up to the website. We will have the same navigation bar on this page as well.

Product Listing Page:



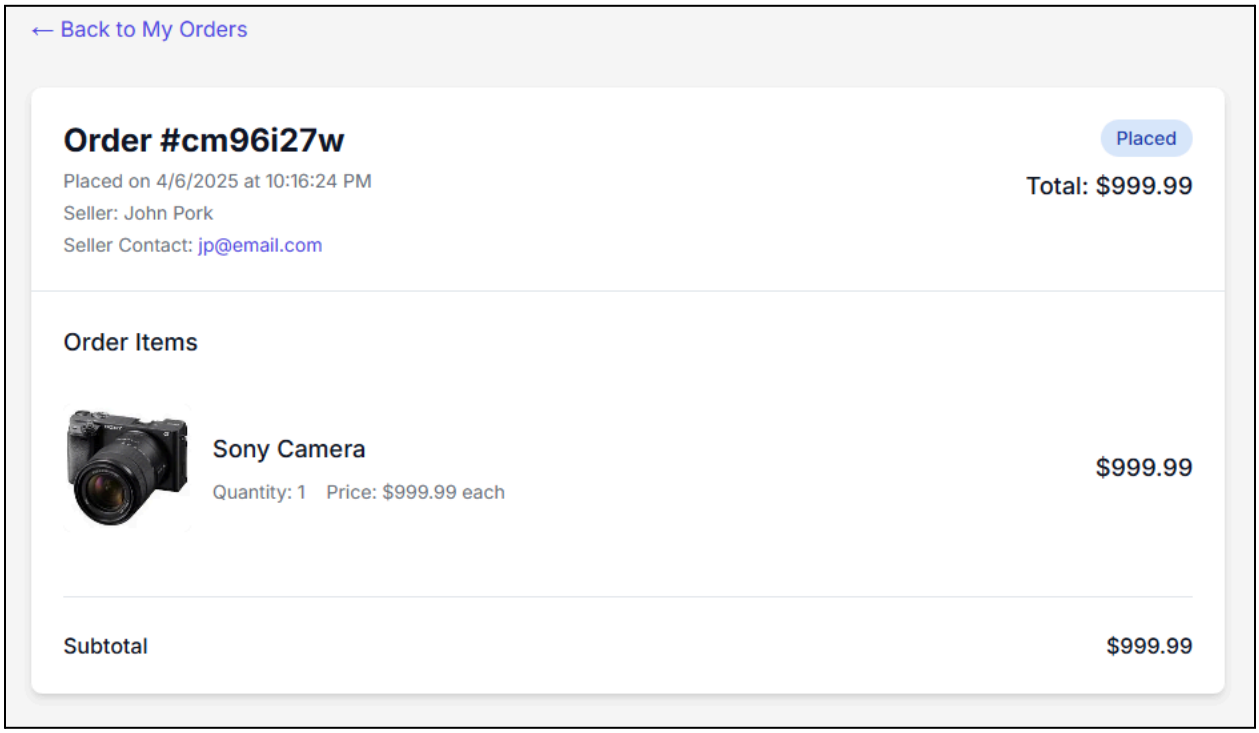
Here on this page users will be able to see the listed products and click to see the product details.

Details Page:



Here users can see the details about a product and add it to their cart.

Order Success Page:



After completing the order, users will see something like this. They can go see all their orders.

Order Management Page:

This is what the website admin will see. They will be able to update the order status.

Final Comments

We aim to master full-stack development by owning features end-to-end. This project will solidify our skills in React, Node.js, and database management.