Je-Hyun Kim
TCS / Simplilearn
18 November 2020

MEAN Stack Fashionista Capstone: Final Specification Document

Github Repository: https://github.com/je-hyun/fashionista

Instructors: Bhawna Gunwani & Akash Kale

Table of Contents

Project Details

Create a website that deal with online shopping for fashion goods and accessories.

Background of the problem statement:

You work at XYZ solutions as a MEAN stack developer. Due to the Covid-19 situation every business is becoming online. You are asked by your management to lead and develop a dynamic website for online shopping of fashion items and accessories.

Implementation requirements:

1. Create a rich frontend
2. Create admin and user portals
3. Create functionalities for admin portal such as adding, updating, and deleting an item and user.
4. Create functionalities for user portal such as, registration, sign in, purchasing item, adding to the cart.
5. Create database for storing all application related information.

The following tools must be used:

1. VS Code
2. Angular
3. NodeJS
4. MongoDB

The following things to be kept in check:

1. You need to document the steps and write the algorithms in them.
2. The submission of your GitHub repository link is mandatory. In order to track your tasks, you need to share the link of the repository.
3. Document the step-by-step process starting from creating test cases, then executing them, and recording the results.
4. You need to submit the final specification document, which includes:
    a. Project details
    b. Concepts used in the project
    c. Links to the GitHub repository to verify the project completion
    d. Your conclusion on enhancing the application and defining the USPs (Unique Selling Points)

BACKLOG

Backend:

- ✓ Initialize project backend api
- ✓ User Create
- ✓ User Read
- ✓ User Update
- ✓ User Delete
- ✓ Product Create
- ✓ Product Read
- ✓ Product Update
- ✓ Product Delete

Frontend:

- ✓ Register
- ✓ Sign in
- ✓ Admin CRUD on products
- ✓ Admin CRUD on companies
- ✓ View all products
- ✓ Add to cart
- ✓ Checkout page

Concepts Used

Several technical concepts were used in this project.

1. MongoDB
   a. Collections
   b. Embedded documents (to model 1:m relationships)
   c. Mongoose / schemas
2. REST API
   a. Backend managed through a separate api
   b. GET, POST, PUT, DELETE
   c. MVC
3. Angular
   a. Dependency Injection (to make our application memory performant and easy to manage)
   b. Routing (using angular router)
   c. CORS (Cross Origin Resource Sharing)
   d. Components
   e. Services
   f. Modules
   g. MVC

Steps documented

1. Created a project initialized with a backend and frontend component.
    a. Backend will use mongodb and express to serve the frontend, which is done in angular.
2. Created a project backlog to guide and track progress and a README.md
3. Create login/register/authentication
4. Created companies-list route to list out companies for admin portal
5. CRUD implemented for companies
6. Created products-list route to list out products for admin portal
7. CRUD implemented for products
8. Add check for admin/user account (routes/header)
9. Add routes for shop, checkout, and cart
10. Implement shop view
11. Implement shop add to cart functionality
12. Implement cart view and deletion
    a. Carts are stored in localstorage
13. Implement checkout page
14. Update screenshots and documentation

Enhancements

It’s crucial to note future enhancements that could be possible for the growth of a project. Among notable enhancements include:

- Display related products suggested alongside a product’s page.
- Allow for creation of more than one admin account, possibly with different privileges (for example, an admin account with access to a specific company only)
- Filtering and searching products by specific criteria
- Saving payment info / securing that saved information

USPs

There are several unique selling points related to the fashionista website. The website is general-purpose in that it could easily be altered to accommodate different products by simply editing the mongodb collections to our needs. The shopping cart allows users to have a seamless experience in shopping for multiple items before checking out. The admin portal allows for separate login with different functionalities. An admin can do CRUD operations on companies and products.