

Center of Excellence

Overview – Technology E-Commerce Project Requirements

Preliminary Knowledge

Assignment of this full stack curated technology project implies that the cohort has a proficient understanding of fundamental, full stack technologies (Java, JDBC, C#, ASP.NET, HTML, etc.) and has completed the foundations project.

Functional Requirements

Here are the requirements for Project 2 from the Center of Excellence team:

- Login
 - As a User or Admin, you can log into the application.
- Register
 - As a User, you can register a new account.
- Display Products
 - As a User Guest or Admin, you can see a list of available products for you to add to your cart.
- Cart
 - As a User or Guest, you can add items to your cart that you will later purchase or remove from your cart.
- Checkout
 - As a User or Guest, I should be able to checkout with the items in my cart, purchase them and remove them from the inventory.
- View Previous Orders
 - As a User, I should be able to view a list of all my previous orders and access the details of each order.
- User Profile
 - As a User, I should be able to create and maintain a profile page.

Development Requirements

Beyond the functional requirements chosen by the associates, the capstone team will need to complete other objectives that showcase their understanding of core development processes such as devops, testing, and documentation. These extra objectives include:

- **Deploy the Application**
 - Ensure the application is accessible from the cloud, using the platform that was taught during the training program.
- **Documentation**
 - Startup process to ensure another team can get the application running in a local environment as quickly as possible.
 - General application information that provides key details of the application.

Presentation Requirements

The teams will be required to articulate the way they completed the requirements above during a project presentation. These are the core objectives and guidelines for the project presentation:

- **Code Freeze**
 - 24-48 hours before the presentation, the team should halt all development and focus on the presentation.
 - This is a great time to run mock presentations.
- **Preparations**
 - Matching, appropriate, and relevant zoom backgrounds.
 - Business professional attire.
 - Everyone must speak during the presentation.
- **Slide Deck**
 - Technologies utilized in the project
 - Gitflow and branching strategy
 - Application Architecture and ERD Diagram
 - Outstanding Defects or Issues
 - Complications or Challenges during the development experience
- **Live Demonstration**
 - *Do not show any code!*
 - Ensure you demonstrate the functionality was properly implemented.

- Do not show incomplete features in the demonstration.
- Practice the demonstration during the code freeze, too! (Not just the slides)
- Make the UI styling presentable and professional.
- Walk through all the MVP features that were completed during the capstone.

Optional Features

This section describes all the potential extensions to the project functionality.

- **Reset Password**
 - As a User or Admin, I should be able to reset my password.
- **Remove Items from Cart**
 - As a [User|Admin], I should be able to remove items from my cart within the Cart View.
- **Create/Update Products**
 - As an Admin, I should be able to create a new product to be displayed or update an existing product.
- **Product Details View**
 - As a [User|Admin], I should be able to click on a product to get more details on the product itself.

GitFlow Strategy Suggestion

This section outlines a strategy for Git version control flow. Leveraging proper GitFlow tactics is vital to the team's success during the capstone project. We offer this suggestion as guidance for proper gitflow tactics that will be best to practice before moving on to client projects. Please reference the outline below.



Here are the important details of the recommended GitFlow strategy:

- **Main/Master Branch**
 - This is the deployment branch of the project and only working code should be pushed to this branch.
 - Utilize this branch to set up webhooks for CI/CD pipelines.
- **Development Branch**
 - This branch should be a direct extension of the main/master branch in your repository.
 - This branch serves as the first gateway for code to cross the bridge and move into the main branch.
 - Utilize the development branch to settle merge conflicts when features are completed.
- **Feature Branches**
 - Feature branches are utilized to create version control of the ongoing feature development within the team structure.
 - Example: Login feature branch, Register feature branch, etc.
 - Multiple teams can work on a single feature branch and should have their own branches off the feature branch.
 - If the team structure is designed around feature-based teams, this gitflow strategy will work to its maximum capacity.