



**COLLEGE CODE : 9530**

**COLLEGE NAME : ST.MOTHER THERESA ENGINEERING COLLEGE**

**DEPARTMENT : COMPUTER SCIENCE AND ENGINEERING**

**STUDENT NM-ID : 017937211CB486D9CEF4D6BC13F5B7B9**

**ROLL NO : 953023104037**

**DATE : 22.09.2025**

**Completed the project named as**

**Phase_3_ TECHNOLOGY**

**PROJECT NAME : PRODUCT CATALOG WITH FILTERS**

**SUBMITTED BY,**

**NAME : JERUSHA.J**

**MOBILE NO :7397507002**

**PRODUCT SETUP :**

Laying the Foundation

- Define MVP Scope :
  Clear boundaries: product listing with basic filtering only, avoiding feature creep.
- Choose Tech Stack :
  React for responsive frontend ,Node.js backend ,MongoDB for flexible data storage.
- Repository Setup :
  GitHub initialization with proper branching strategy and collaboration workflows.
- Development Environment :
  Linting, testing frameworks, and CI/CD pipeline basics for quality assurance.

**CORE FEATURES IMPLEMENTATION :**

Building the Essentials

Product Catalogue Display :

Dynamic rendering of product cards with images, names, and prices for engaging user experience.

Filter Functionality :

Category, price range, and availability filters with efficient state management.

Responsive Design :

Seamless usability across all devices with performance-first approach.

**DATA STORAGE :**

Structuring for Flexibility and Speed

- MongoDB Schema-less Design :
  Flexible product data structure allowing easy addition of new attributes without migrations.
- Essential Product Details :
  Comprehensive storage: name, category, price, stock status, and searchable filter tags.
- Optimised Indexing :
  Strategic indexing on category and price fields to ensure lightningfast filter queries.
- Efficient API Endpoints :
  Purpose-built endpoints for filtered product retrieval with minimal data transfer.

**TESTING CORE FEATURES :**

Ensuring Reliability Early

- Unit Testing :

Jest and React Testing Library for filter logic and component rendering validation.

- Integration Testing :

API endpoint validation ensuring correct data retrieval and filtering accuracy.

- Manual Testing :

Exploratory testing across multiple devices and browsers for real-world validation.

- Automated CI Pipeline :

Continuous testing integration to catch regressions before they reach production.

**VERSION COTROL WITH GITHUB :**

Collaboration and Traceability

- Feature Branches :

Isolated development for catalogue and filters, preventing conflicts and enabling parallel work.

- Pull Request Reviews :
  Code quality maintenance through peer reviews and knowledge sharing across the team.
- Release Tagging :
  Clear MVP milestones with easy rollback capabilities and deployment tracking.
- GitHub Actions :
  Automated testing and deployment workflows reducing manual errors and deployment time.

  GitHub URL : https://github.com/jerusha-556/SMTEC-NM-IBM-PRODUCT-CATALOG-WITH-FILTERS-.git

**VISUAL STORYTELLING :**

Before & After MVP

Before Implementation :

- Static product list presentation
- No filtering capabilities
- Slow, cumbersome user experience

- Limited product discoverability

After Implementation :

- Interactive catalogue with real-time filters
- Faster load times and smooth interactions
- Enhanced user experience and engagement
- Improved product discoverability

The transformation showcases how focused MVP development delivers immediate value whilst establishing a foundation for future enhancements.

**CONCLUSION :**

MVP as a Launchpad for Success

- Risk Minimisation :

MVP approach accelerates learning whilst reducing development risk through focused feature delivery.

- Immediate Value :

Core features deliver tangible user value from day one, establishing product-market fit foundation.

- Quality Assurance :

Robust testing and version control ensure team efficiency and maintainable, scalable codebase.

- Iterative Growth :

Data-driven iteration based on real user insights creates a pathway to sustainable product evolution.