1. Development Process

The project followed an Agile development process, specifically Scrum. This approach was chosen for its iterative and adaptable nature, allowing for frequent feedback, collaboration, and quick issue resolution. Daily stand-up meetings facilitated communication within the team, ensuring continuous alignment with stakeholder expectations.

2. OO Design and UML Diagrams

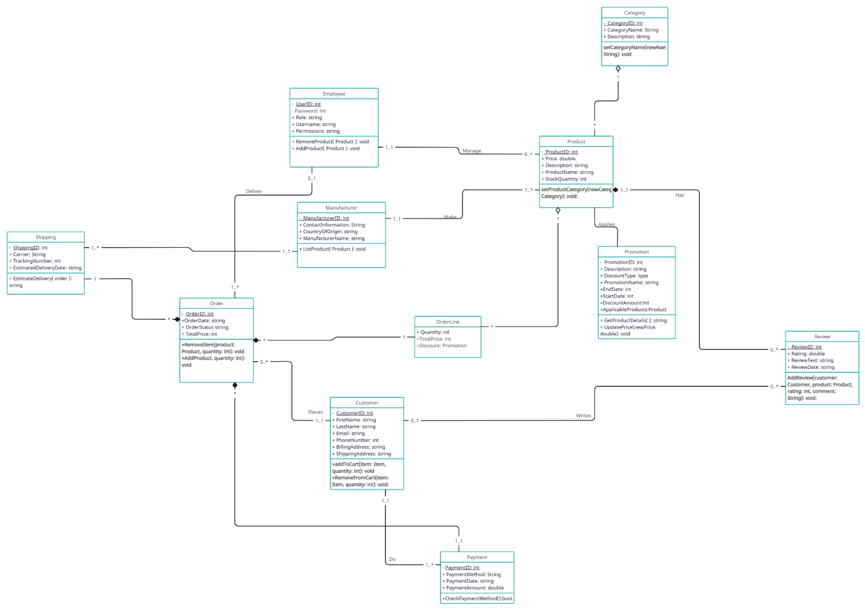
Object-Oriented Design Rationale:

The adoption of object-oriented design principles aimed to enhance modularity, reusability, and maintainability. By fostering a clear separation of concerns, the design promotes straightforward implementation and facilitates future extensions.

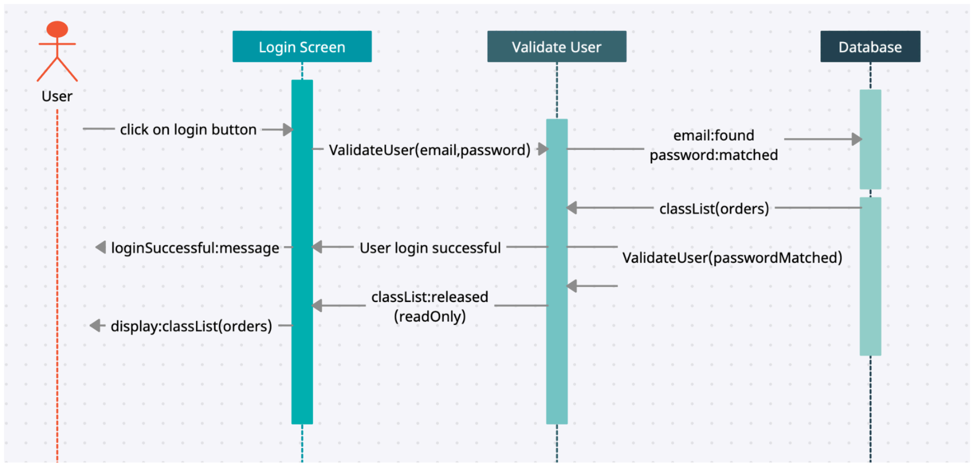
UML Diagrams:

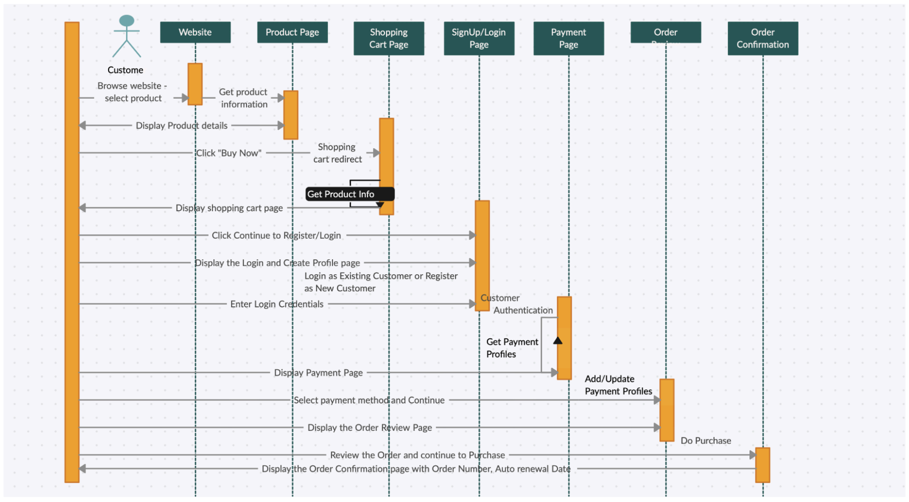
The UML diagrams include:

- Class Diagrams:

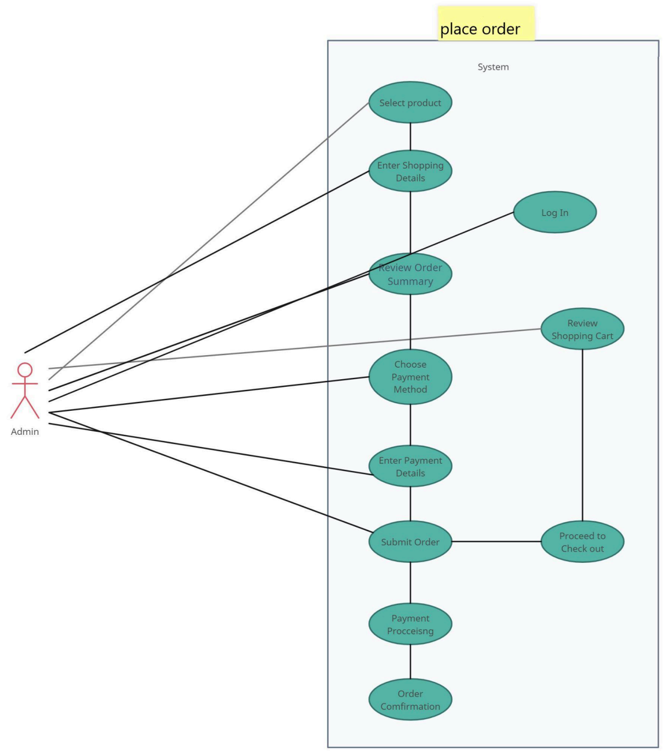


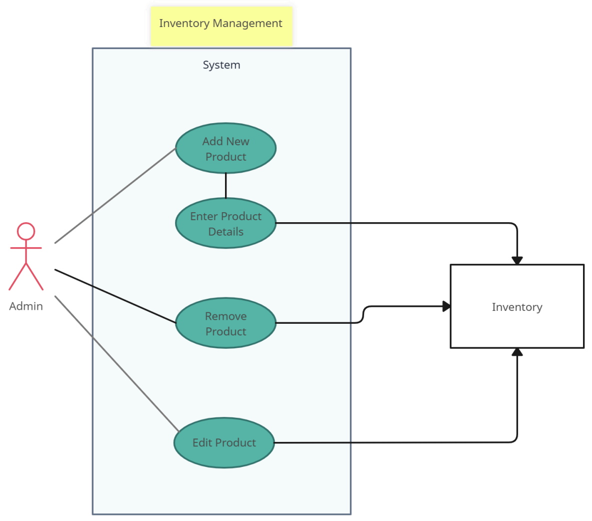
- Sequence Diagrams:




Custome
Website
Product Page
Shopping Cart Page
SignUp/Login Page
Payment Page
Order
Order Confirmation
Browse website - select product
Get product information
Display Product details
Click "Buy Now"
Shopping cart redirect
Get Product Info
Display shopping cart page
Click Continue to Register/Login
Display the Login and Create Profile page
Login as Existing Customer or Register as New Customer
Enter Login Credentials
Customer Authentication
Get Payment Profiles
Display Payment Page
Add/Update Payment Profiles
Select payment method and Continue
Display the Order Review Page
Do Purchase
Review the Order and continue to Purchase
Display the Order Confirmation page with Order Number, Auto renewal Date

- Use Case Diagrams:

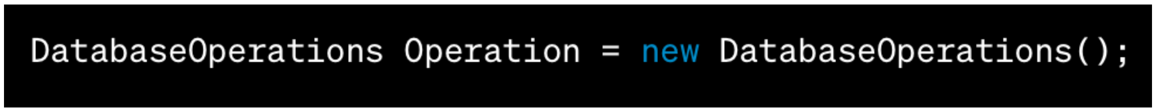




3. Design Patterns

Two design patterns were applied in the system:

- Singleton: **ProductModel** class, creating a single instance of the DatabaseOperations class:

  ```
  DatabaseOperations Operation = new DatabaseOperations();
  ```

  Modularity Benefit: This ensures that there is a single point of access to the DatabaseOperations instance throughout the ProductModel class. It centralizes the creation and management of the database operations, making it easier to control and maintain.

- Observer: In the **OnPostSubmitReview** method, you are updating various properties based on the submitted review.
  The observers, in this case, are the properties like ReviewsCount, avg_rating, star1Count, star2Count, etc., which gets updated when a new review is submitted.
  Modularity Benefit: By observing changes in the review data, you can easily update various properties without tightly coupling the logic. This promotes a modular approach where the update logic is separated from the actual data.

4. Testing Plans and Scripts

Comprehensive testing plans involved the use of xUnit for C# unit testing and Selenium for automated UI testing. The scripts covered functional and non-functional requirements, ensuring correctness, reliability, and performance.

5. CI/CD Workflow

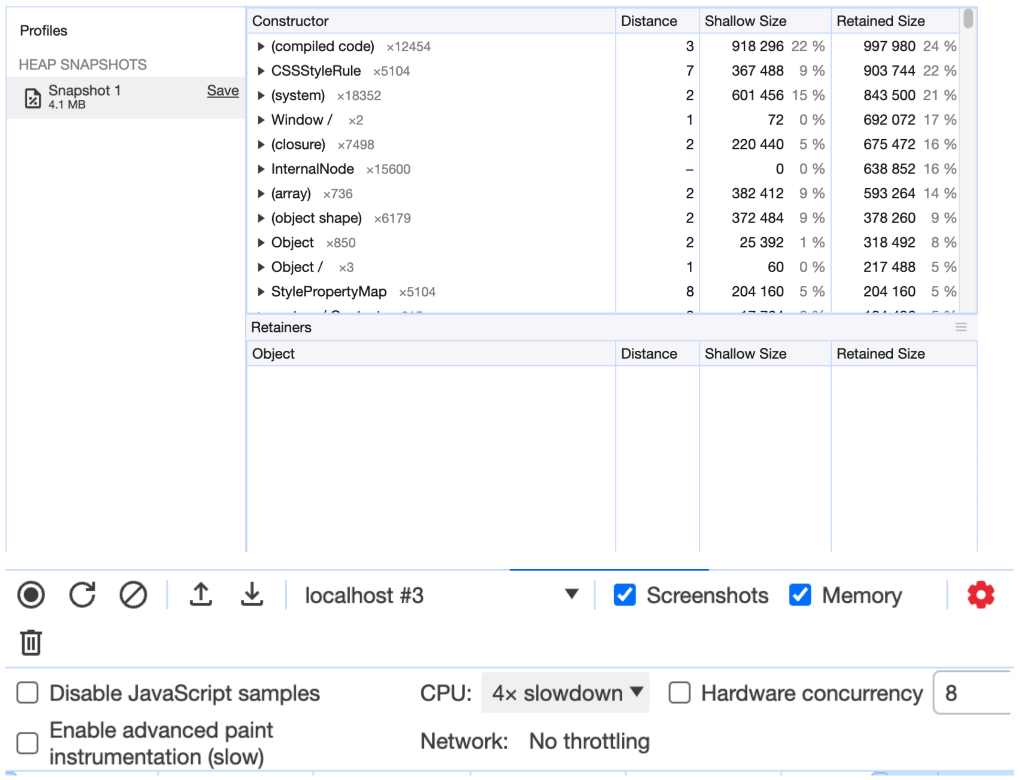
Continuous Integration (CI) was achieved through Azure Pipelines. Code repositories on GitHub triggered automated builds upon each commit. Continuous Deployment (CD) was implemented using Azure Pipelines, ensuring that each successful build was automatically deployed to the staging environment for further testing and validation.

6. Model using Docker Image

The deployment model relied on Docker containers for encapsulating the application and its dependencies. Docker images were used to ensure consistency across different environments, facilitating seamless deployment. Docker Compose was employed for orchestrating multi-container deployments.

7. Performance Measures

Performance measures include:



Profiles

HEAP SNAPSHOTS

Snapshot 1 4.1 MB Save

Constructor	Distance	Shallow Size		Retained Size	
▸ (compiled code) ×12454	3	918 296	22 %	997 980	24 %
▸ CSSStyleRule ×5104	7	367 488	9 %	903 744	22 %
▸ (system) ×18352	2	601 456	15 %	843 500	21 %
▸ Window / ×2	1	72	0 %	692 072	17 %
▸ (closure) ×7498	2	220 440	5 %	675 472	16 %
▸ InternalNode ×15600	–	0	0 %	638 852	16 %
▸ (array) ×736	2	382 412	9 %	593 264	14 %
▸ (object shape) ×6179	2	372 484	9 %	378 260	9 %
▸ Object ×850	2	25 392	1 %	318 492	8 %
▸ Object / ×3	1	60	0 %	217 488	5 %
▸ StylePropertyMap ×5104	8	204 160	5 %	204 160	5 %

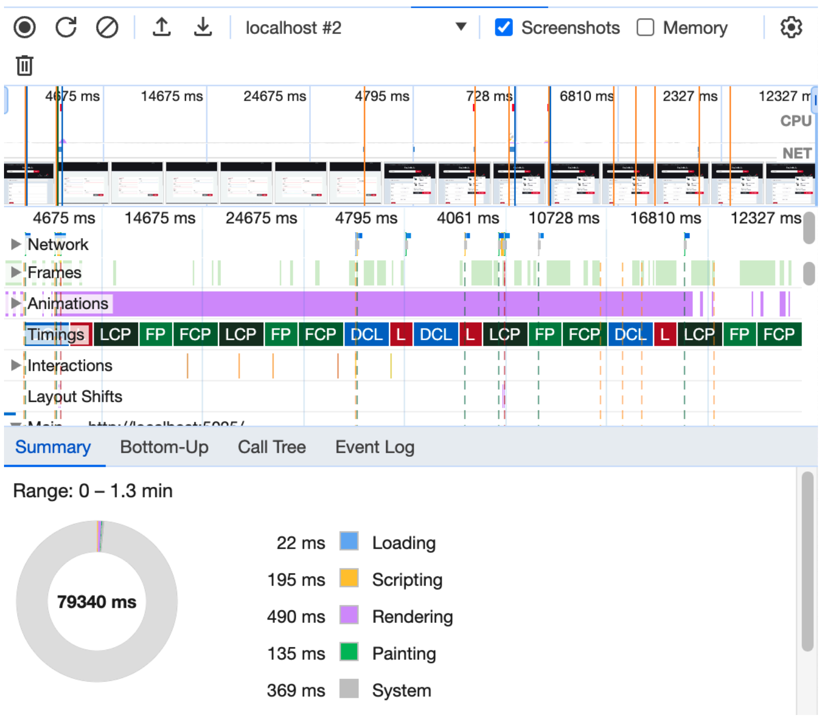
Retainers

Object	Distance	Shallow Size	Retained Size

localhost #3 ▾ ☑ Screenshots ☑ Memory

☐ Disable JavaScript samples CPU: 4× slowdown ▾ ☐ Hardware concurrency 8

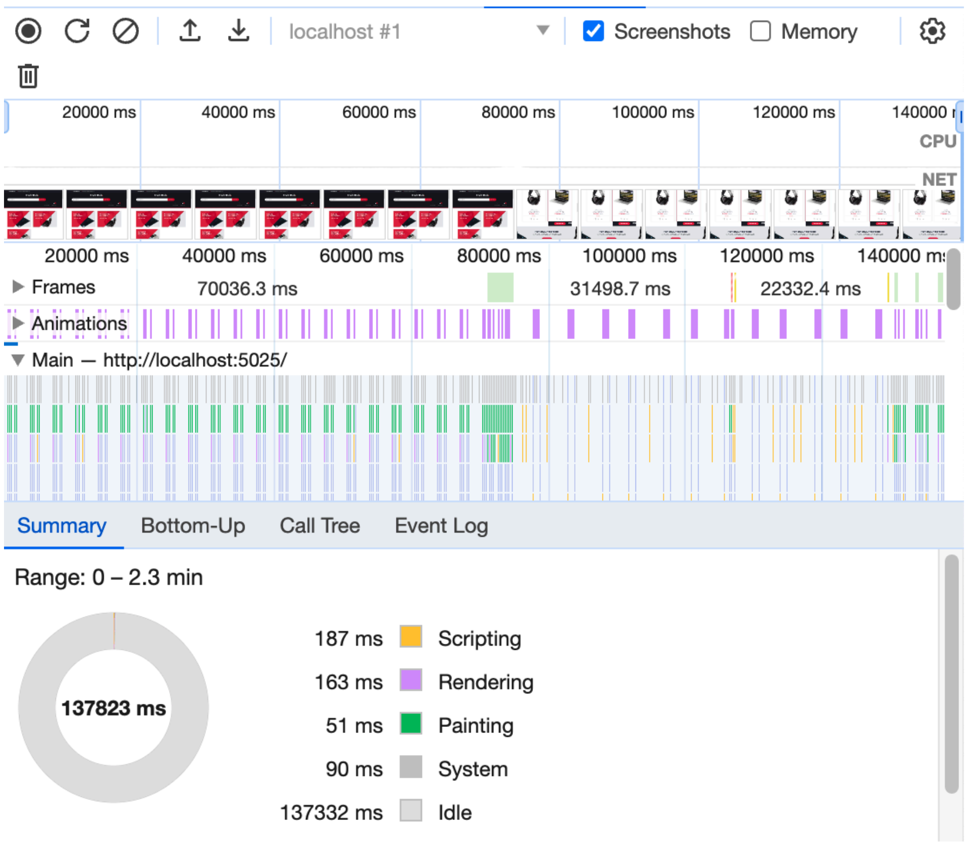
☐ Enable advanced paint instrumentation (slow) Network: No throttling



Range: 0 – 1.3 min

79340 ms

22 ms	🟦	Loading
195 ms	🟨	Scripting
490 ms	🟪	Rendering
135 ms	🟩	Painting
369 ms	⬜	System



Range: 0 – 2.3 min

137823 ms

187 ms	🟨	Scripting
163 ms	🟪	Rendering
51 ms	🟩	Painting
90 ms	⬜	System
137332 ms	⬜	Idle

Repo link:
https://github.com/nour-awad/tech-hub.git