**Abstract**

## The "BMSTechkart" initiative introduces a comprehensive full-stack e-commerce application finely tuned to meet the distinctive requirements of developers searching for electronic resources. Skillfully amalgamating HTML and CSS on the frontend, PHP on the backend, and MySQL for database management, the project engineers an immersive and fluid shopping encounter. Beyond conventional e-commerce platforms, it integrates contemporary DevOps approaches by embracing Docker for streamlined containerization and harnessing the prowess of GitHub Actions for automated CI/CD pipeline orchestration. Operating as an electronic resource marketplace, the platform showcases a diverse array of products encompassing headphones, laptops, and smartphones. With a user-centric design, the interface facilitates browsing devoid of mandatory authentication. Authenticated users unlock advanced functionalities, seamlessly adding desired items to their carts and progressing through the ordering process. Ultimately, this e-commerce endeavor furnishes an effortless and effective resolution for online trade in electronic resources, effectively catering to the inclinations of both developers and tech-savvy consumers.

**1. Introduction**

**1.1. The Evolving Landscape of Technology**

In the contemporary milieu, the technology sector has witnessed an unparalleled surge in advancement. This evolution spans from commonplace applications that streamline everyday tasks to intricate software frameworks propelling the operations of global enterprises. Within this dynamic landscape, software development has emerged as a pivotal cornerstone. The architects behind these software achievements are developers, wielding a toolkit of tools, an array of resources, and an ongoing pursuit of knowledge, all of which profoundly shape the caliber and originality of their creations.

**1.2. Identifying the Void**

Within the expansive digital realm, a profusion of e-commerce platforms caters to diverse commodities, ranging from literature to electronic gadgets. However, a conspicuous gap surfaces when in search of a tailored marketplace exclusively catering to the developer community. Existing platforms often blend developer-centric tools, resources, and educational content with general merchandise, resulting in a dearth of finely curated offerings. Moreover, these platforms might lack attributes and functionalities meticulously calibrated to resonate with developers' specific prerequisites. These could encompass elements such as comprehensive API documentation, adept version synchronization, or seamless integration capabilities.

**1.3. Pioneering BMSTechkart's Vision**

Positioned at the vanguard of innovation, the BMSTechkart initiative is poised to ascend as the preeminent e-commerce nucleus bespoke for developers. Our vision encompasses the establishment of a digital realm where developers, regardless of their skill gradient, seamlessly access resources, utilities, and educational modules precisely calibrated to their individual requisites. This inclusive approach accommodates novice programmers embarking on their coding journey as well as seasoned experts in quest of specialized utilities. The BMSTechkart's overarching mission envisions addressing the entire spectrum of developer needs.

**1.4. The Distinctive Edge of BMSTechkart**

Beyond its role as a comprehensive resource repository, BMSTechkart distinguishes itself as a platform meticulously crafted by developers and intended solely for developers. This pivotal distinction ensures that every facet, ranging from the intricacies of search algorithms to the seamlessness of the user interface, is artfully designed with the end user's comfort and requirements as its guiding principle. The platform's ambit extends beyond mere resource provisioning, encompassing the integration of community assessments, peer-based support, and integration blueprints. These endeavours collectively empower developers to extract maximum value from their acquisitions.

In sum, the BMSTechkart initiative transcends the boundaries of a customary e-commerce arena. It emerges as a hub – an amalgamation of resources, a sanctuary for community engagement, and a reservoir of education, all seamlessly unified into a singular entity. This harmonious fusion ensures that developers encounter not merely tools, but also invaluable mentorship, unwavering assistance, and a profound sense of belonging within a shared community.

2. SOFTWARE REQUIREMENT SPECIFICATION (SRS)

2.1 Objectives

* To design a user-friendly interface tailored for consumers.
* To ensure secure storage of user data.
* To implement a robust checkout.
* To automate deployment and achieve continuous integration and continuous delivery.

2.2 Requirement

Technologies and Tools Used:

* Frontend: HTML, CSS and JavaScript
* Backend: PHP
* Database: MySQL and MySQLi
* Server: Apache
* DevOps Tools: Docker for containerization, GitHub for version control, GitHub Actions for CI/CD, and Docker Hub for deployment.

Hardware Configuration (Minimum requirements):

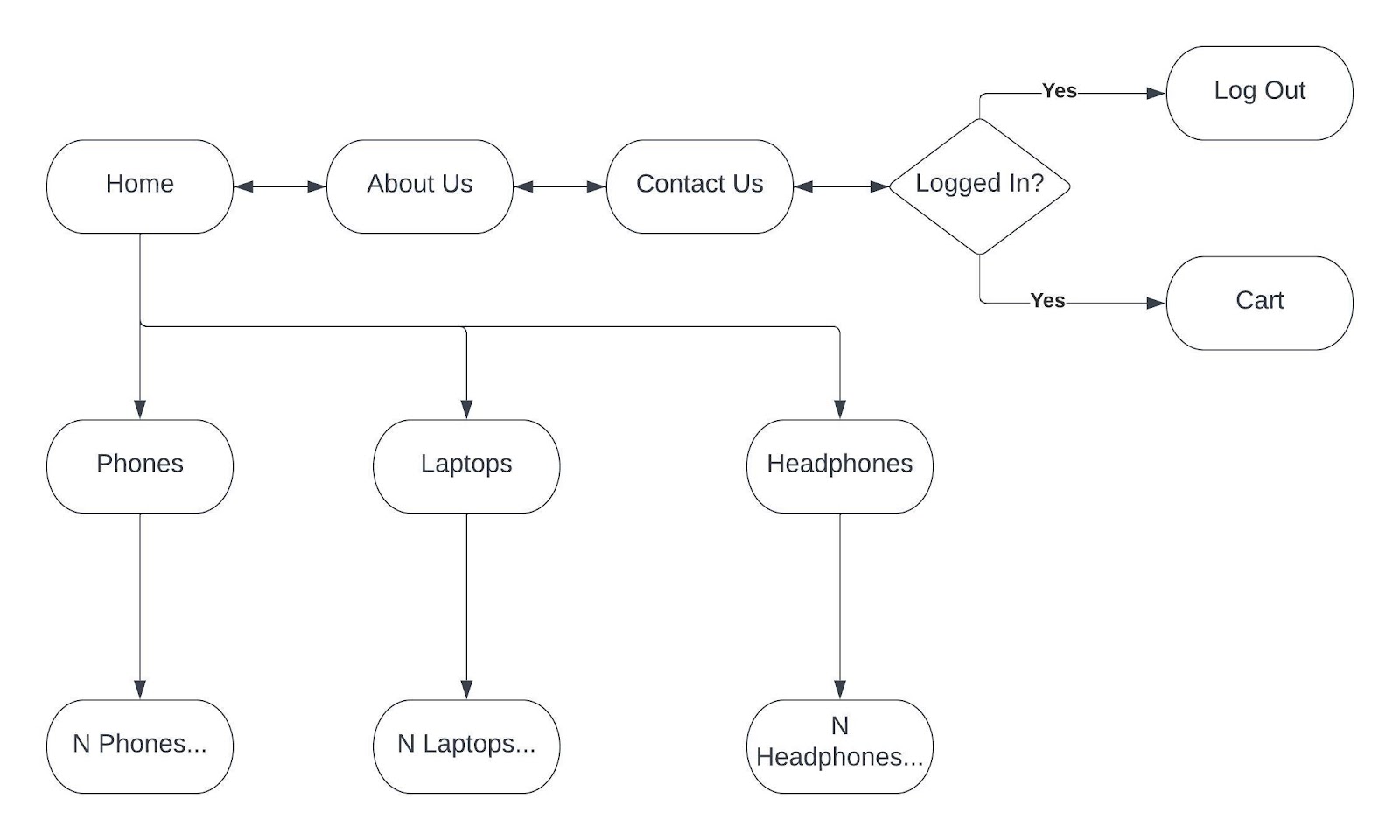
Processor: Core i3, 1.5MHz

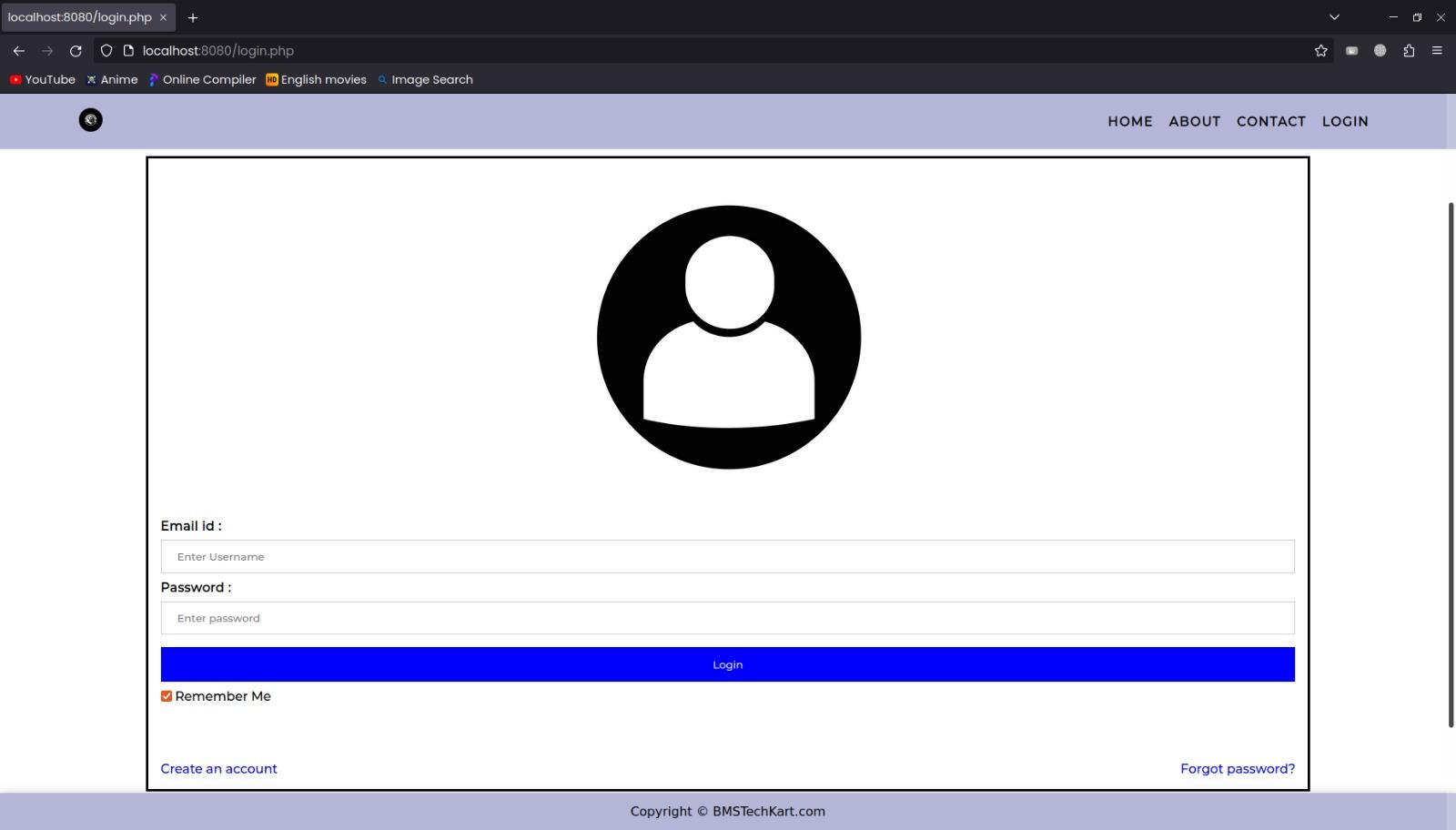
Hard Disk: 150 GB RAM: 2GB

Resolution: 480 X 800

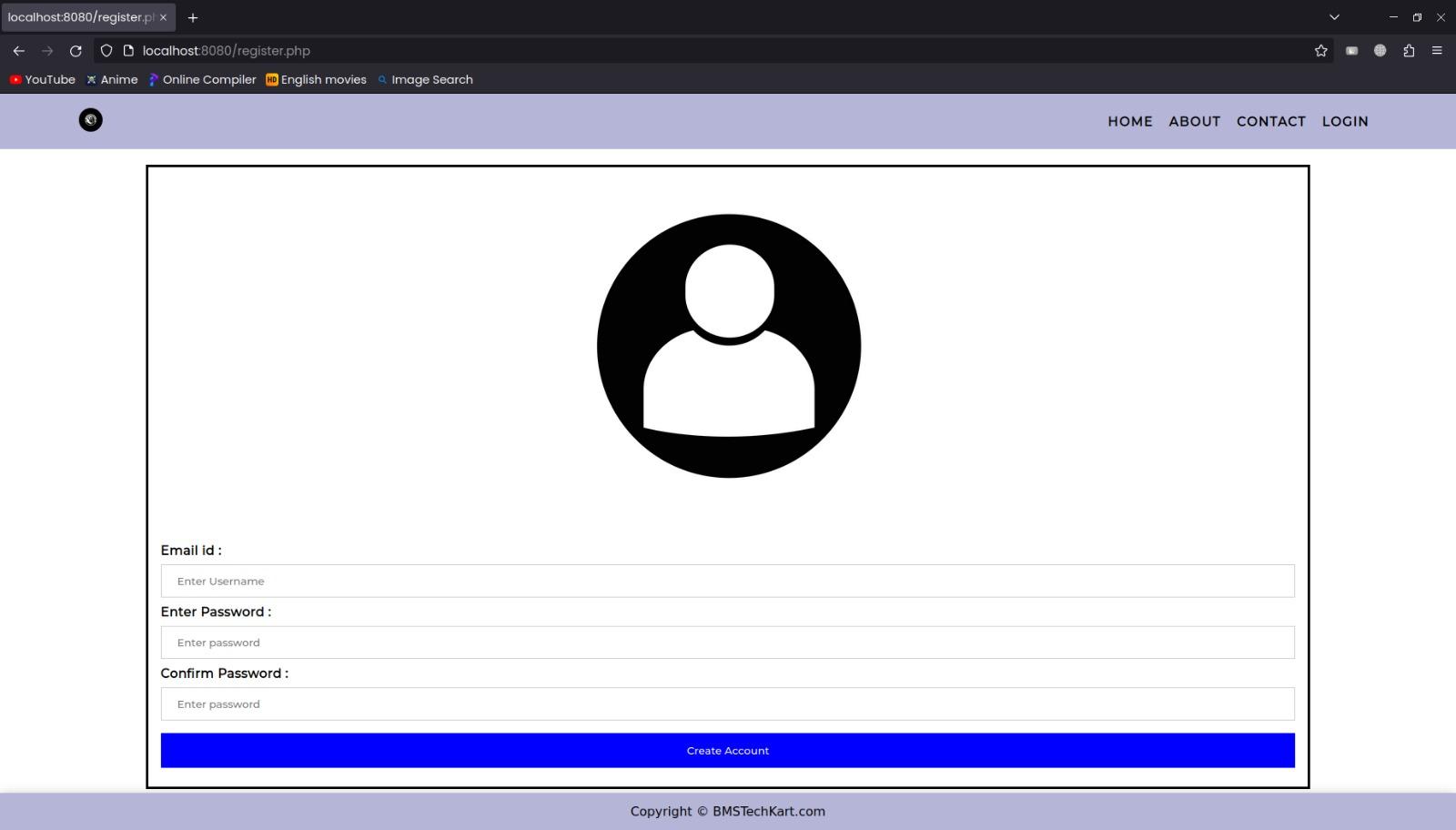
3. System Design

**3.1. Frontend Design and Flow**



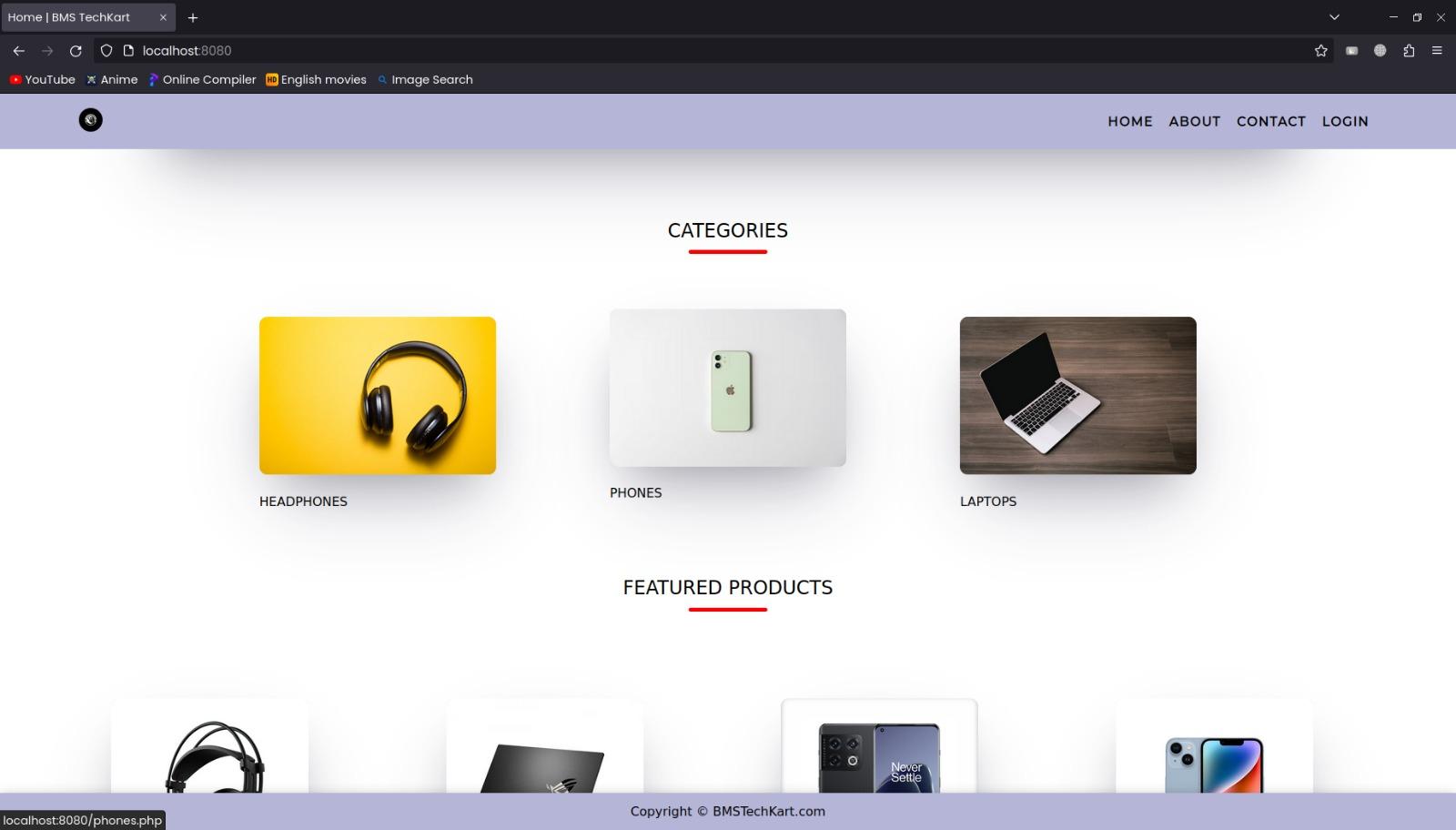
**3.2: Register/Login Webpage**

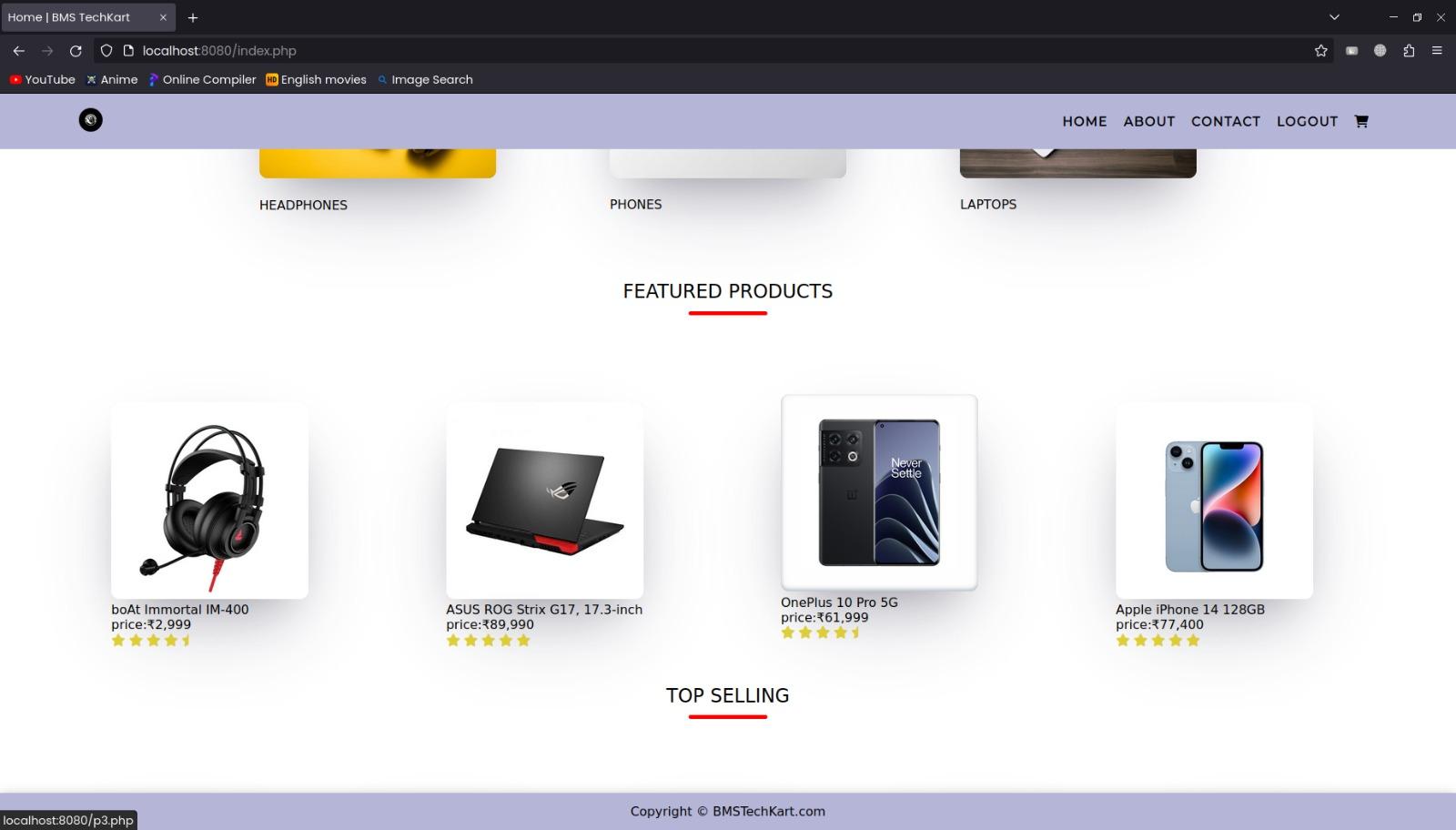
**Fig.3.2.a Login Screen**



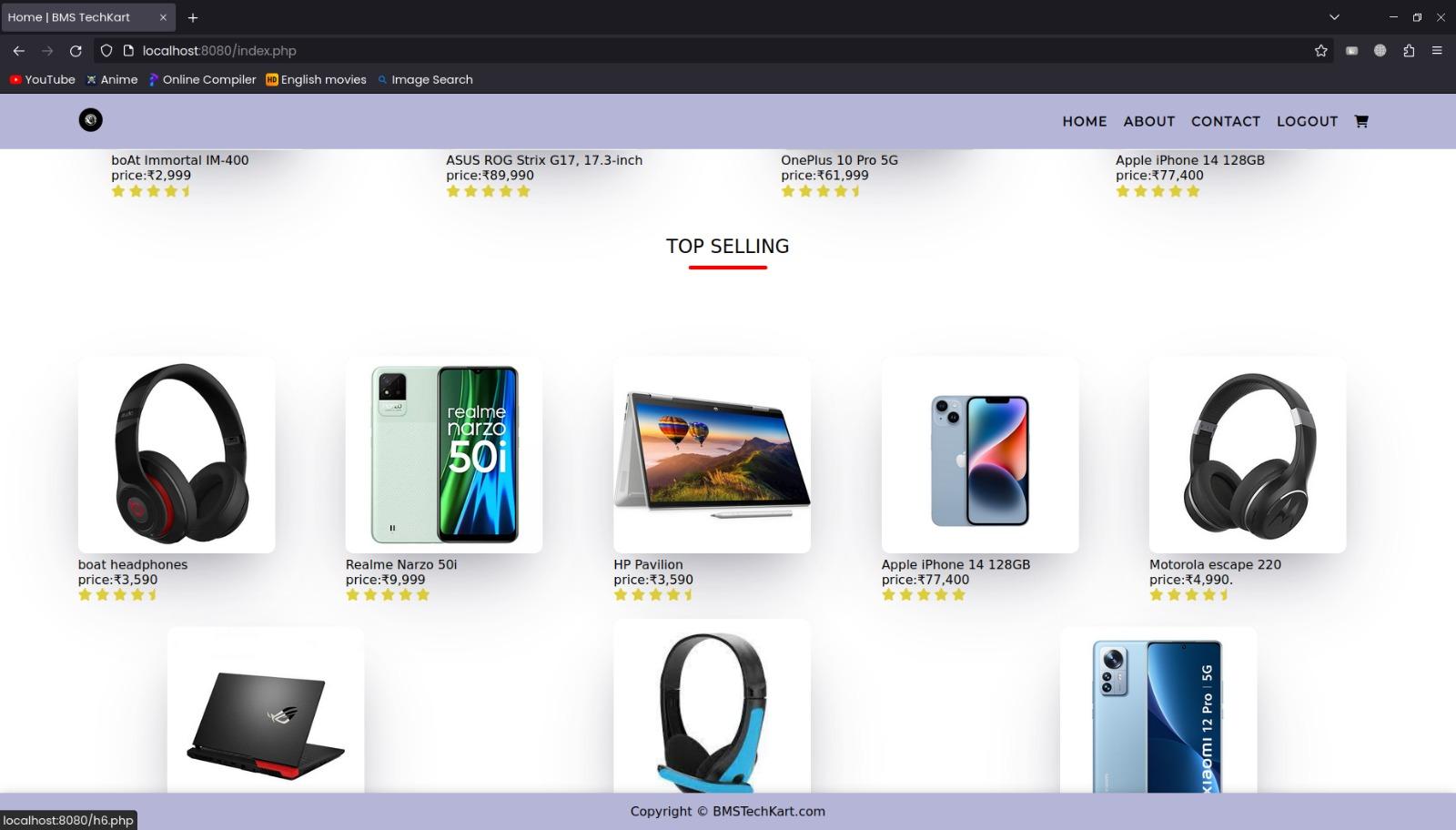
**Fig.3.2.b Register Screen**

**3.3: Home Webpage**



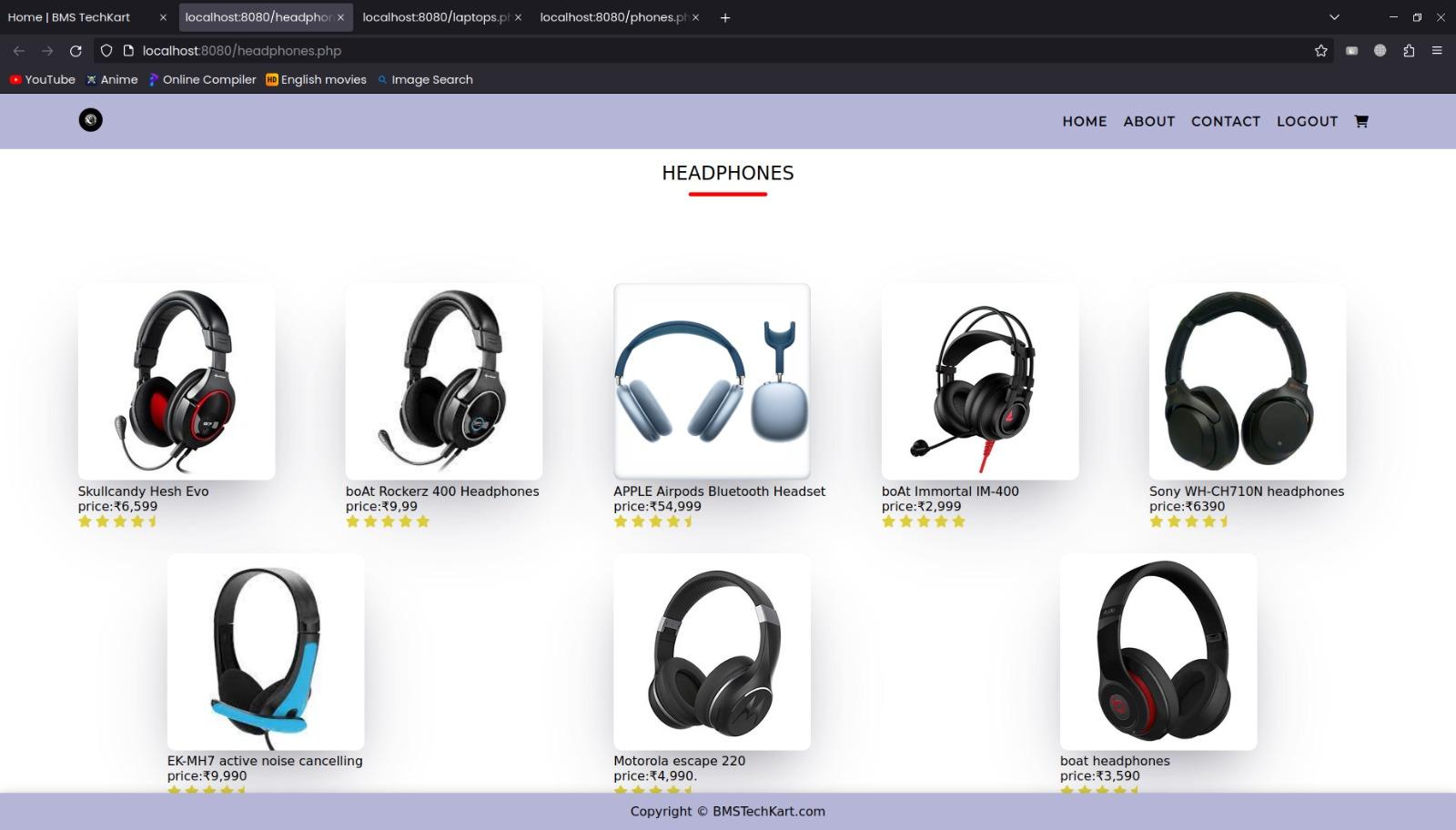


**Fig.3.3.b Featured Products**

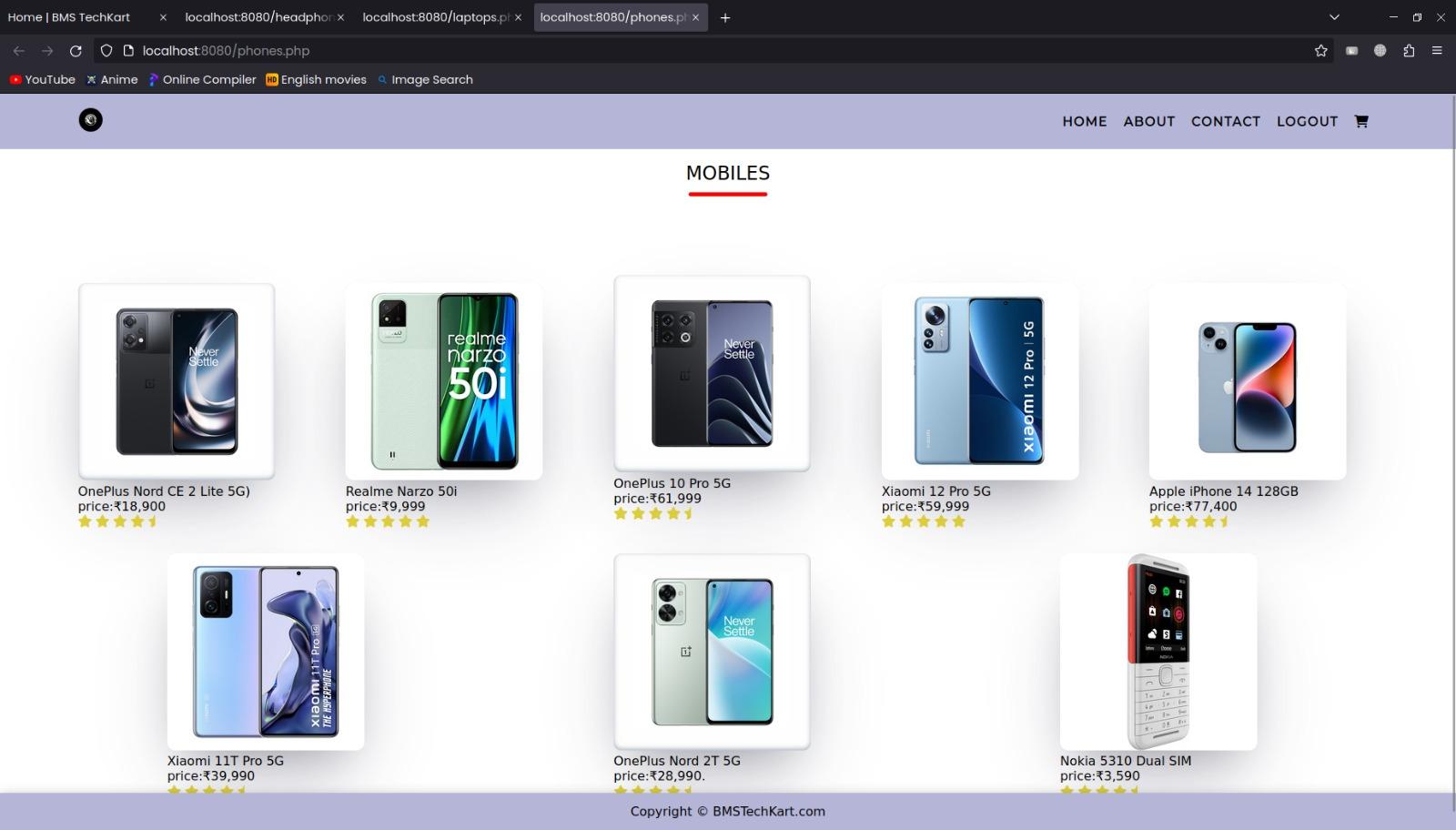


**Fig.3.3.c Top Selling**

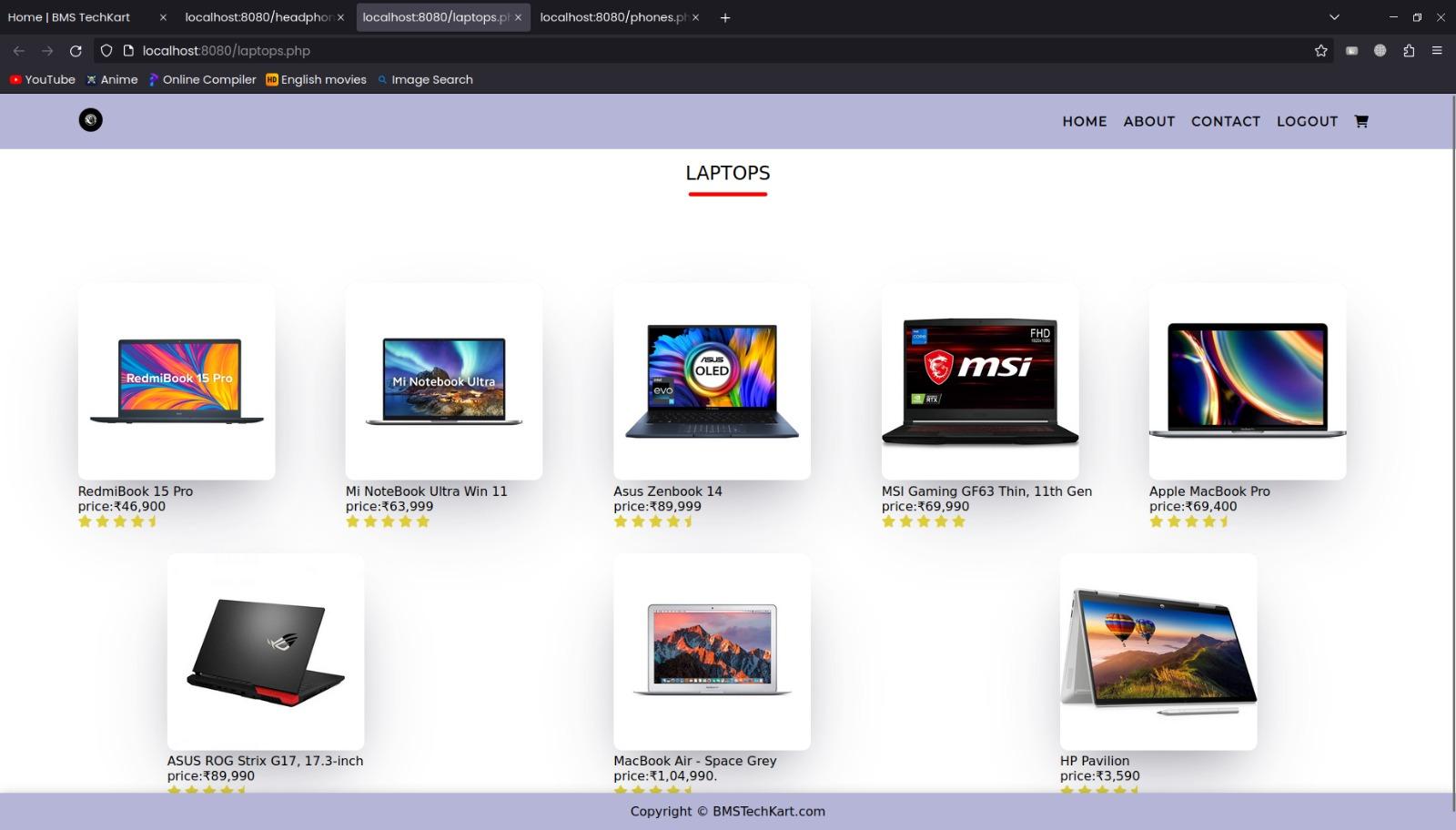
**3.4: Category - HEADPHONES**



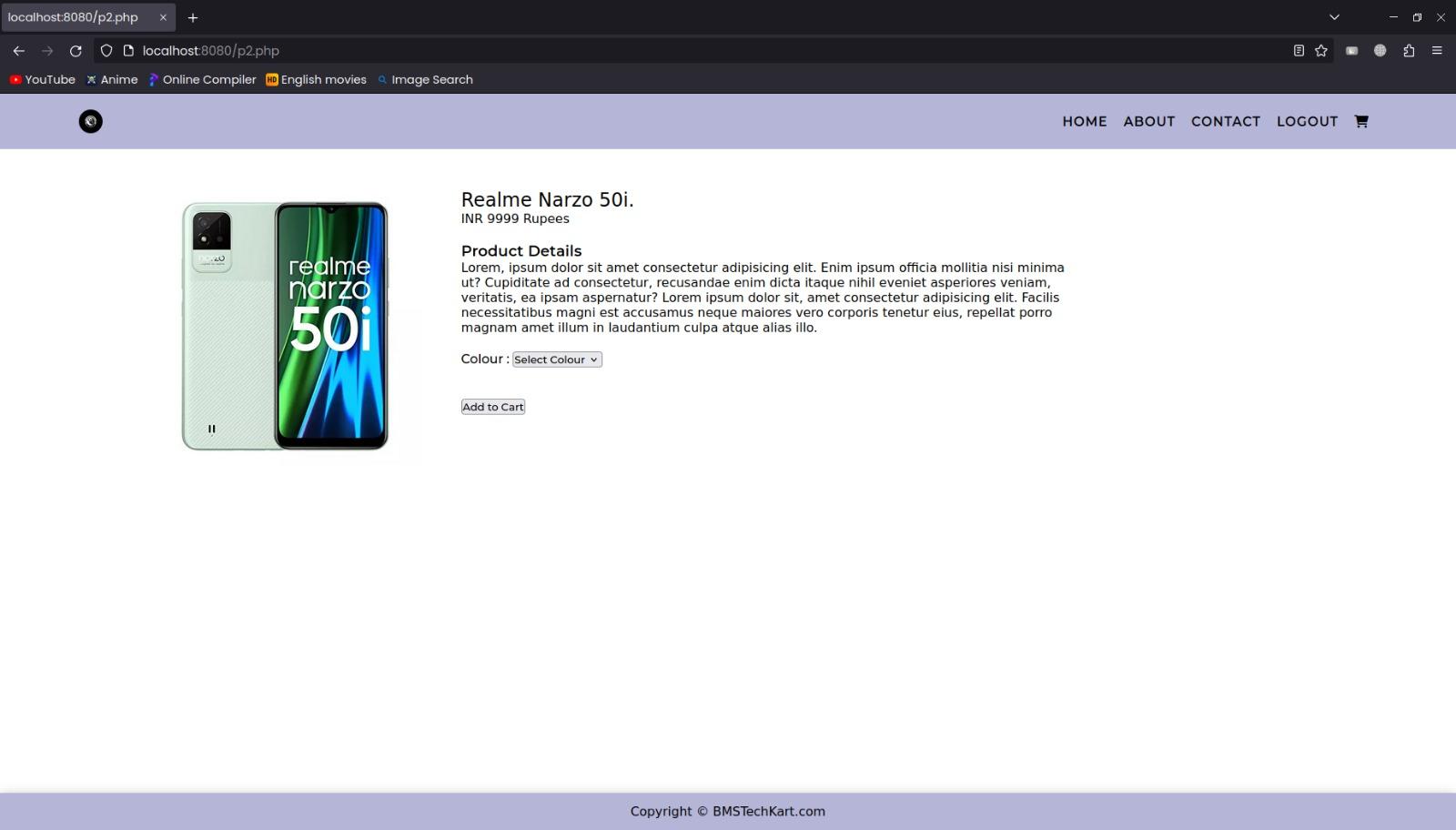
**3.5: Category - PHONES**



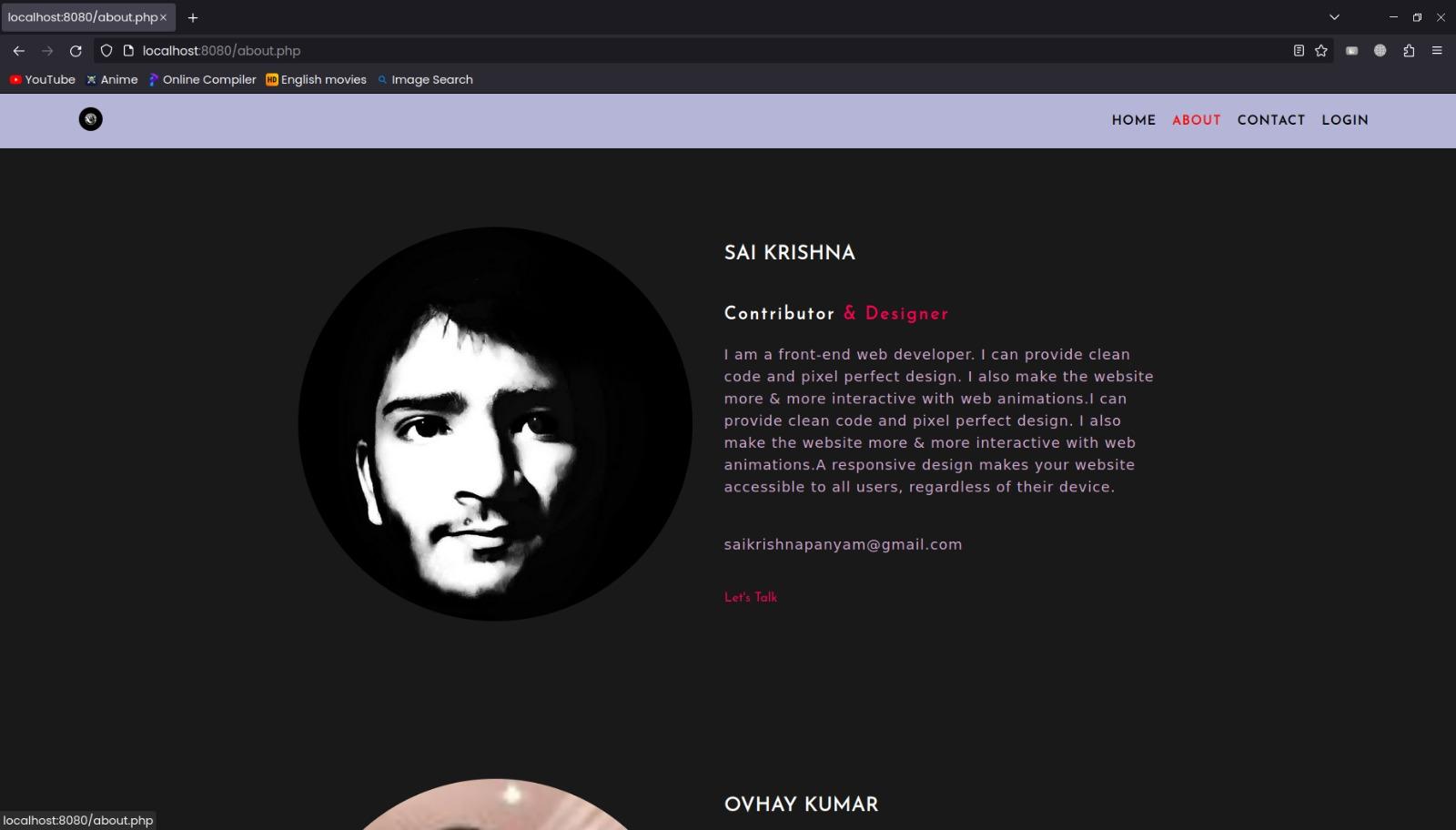
**3.6: Category - LAPTOPS**



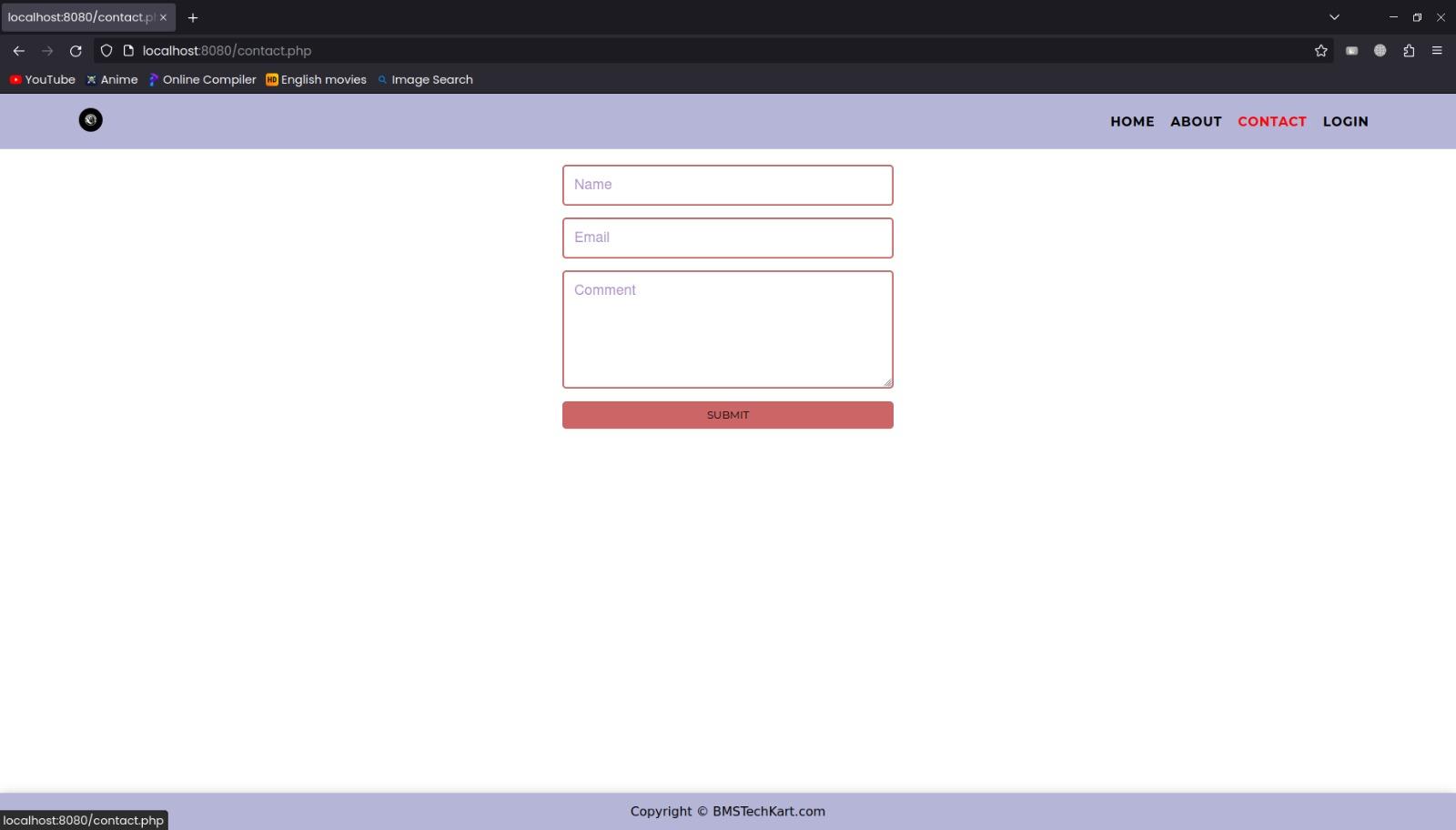
**3.6: PRODUCT DESCRIPTION PAGE**



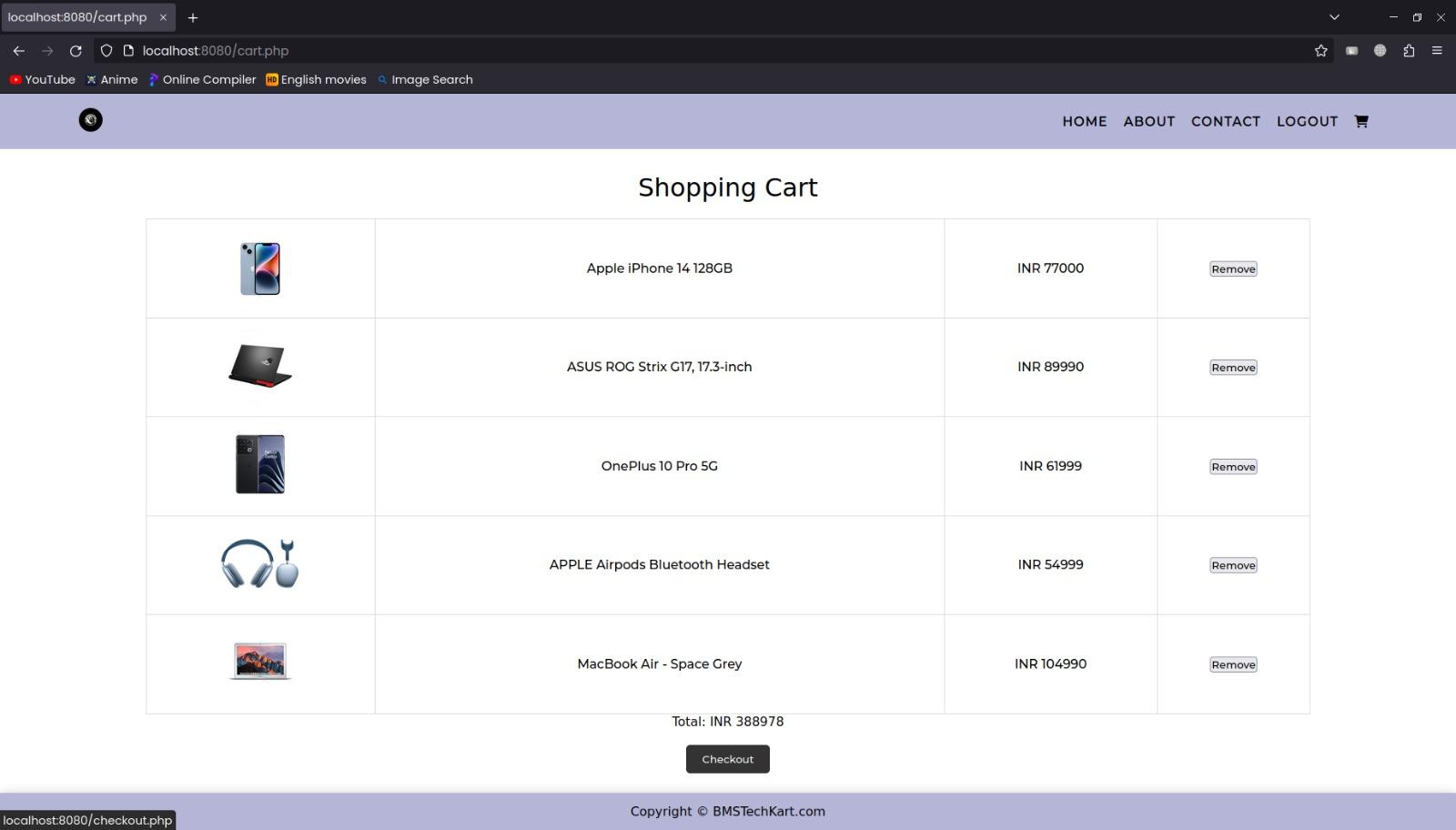
**3.6: ABOUT US PAGE**



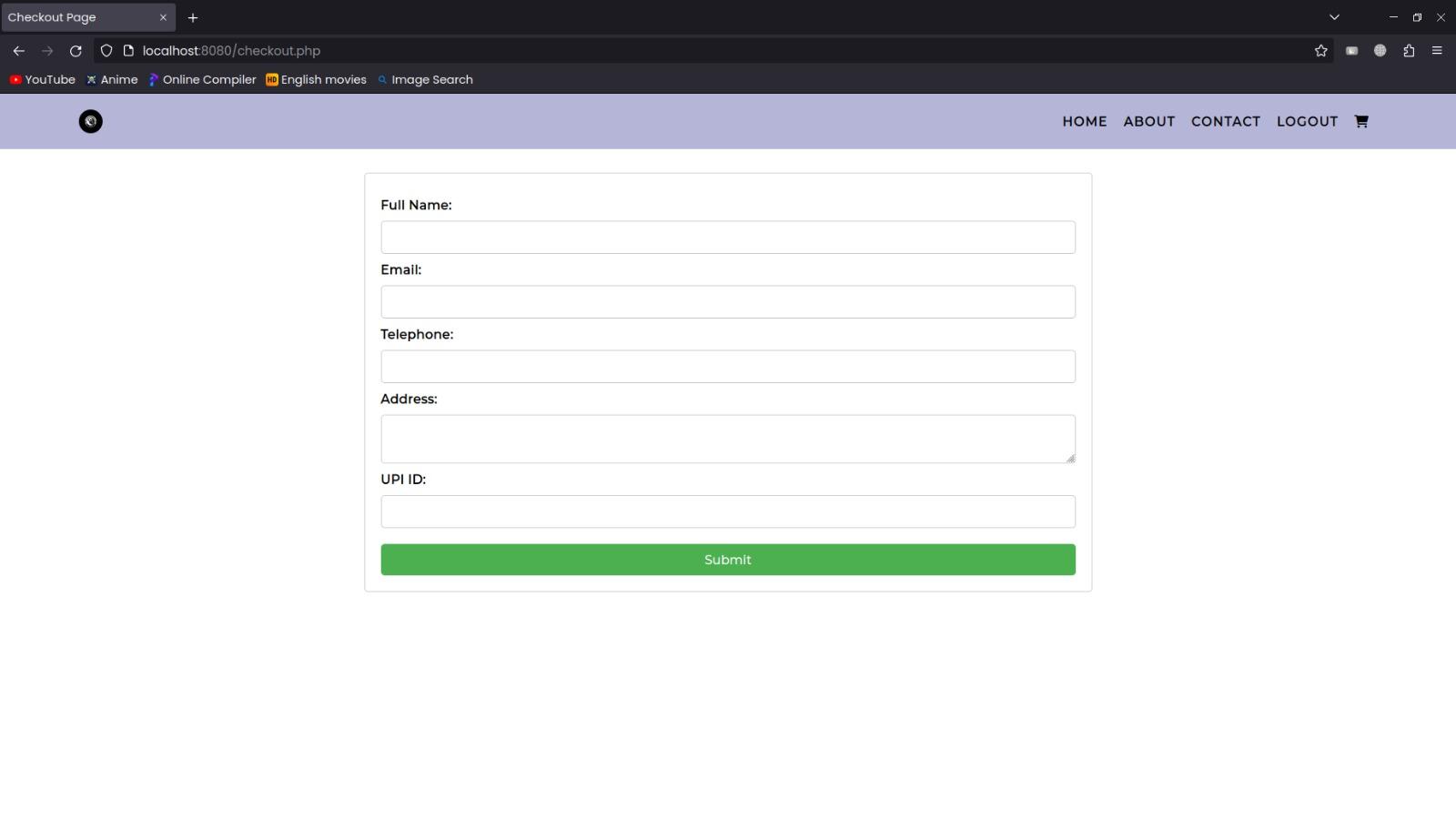
**3.7: CONTACT US PAGE**



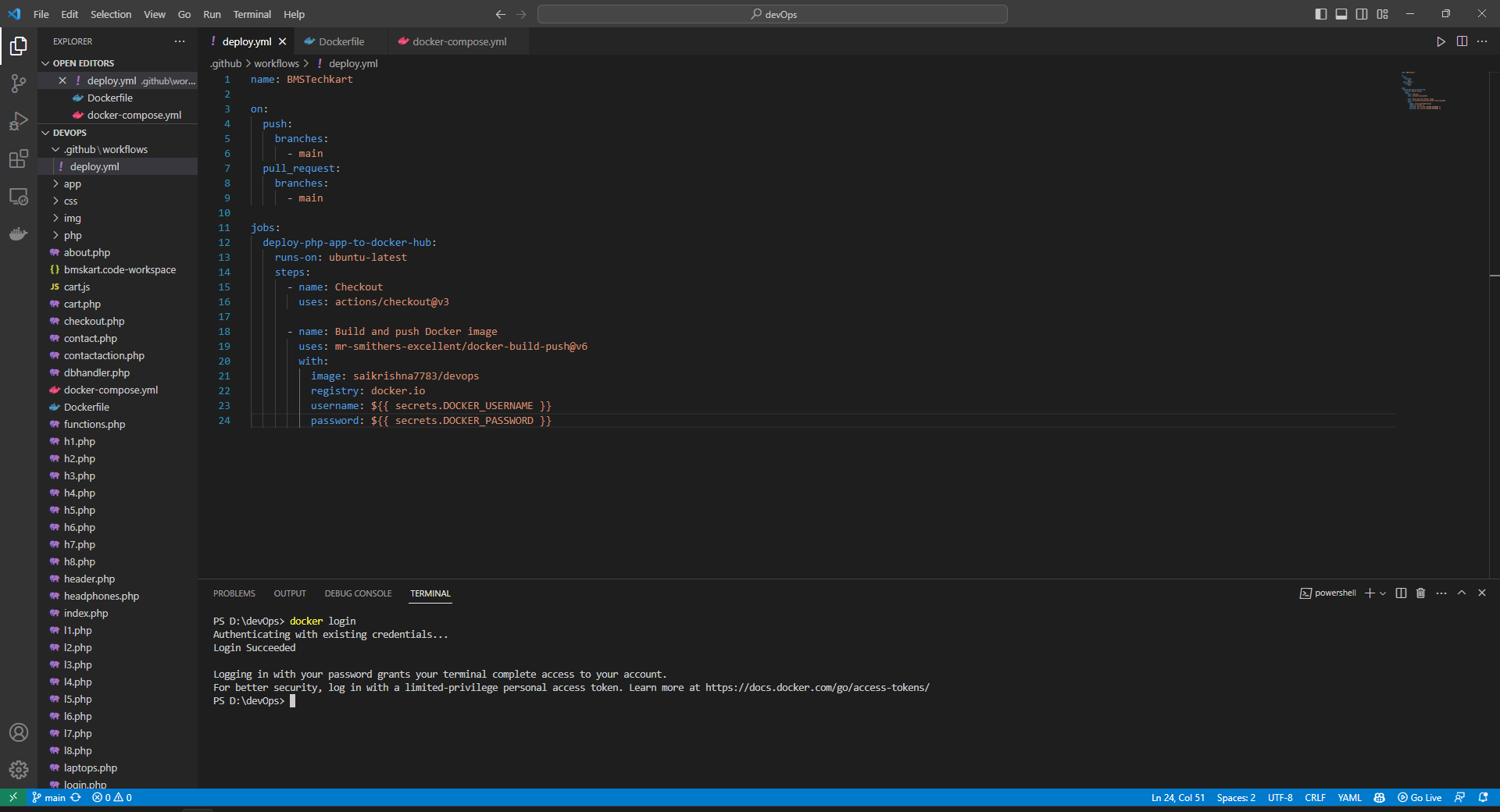
**3.8: CART PAGE**



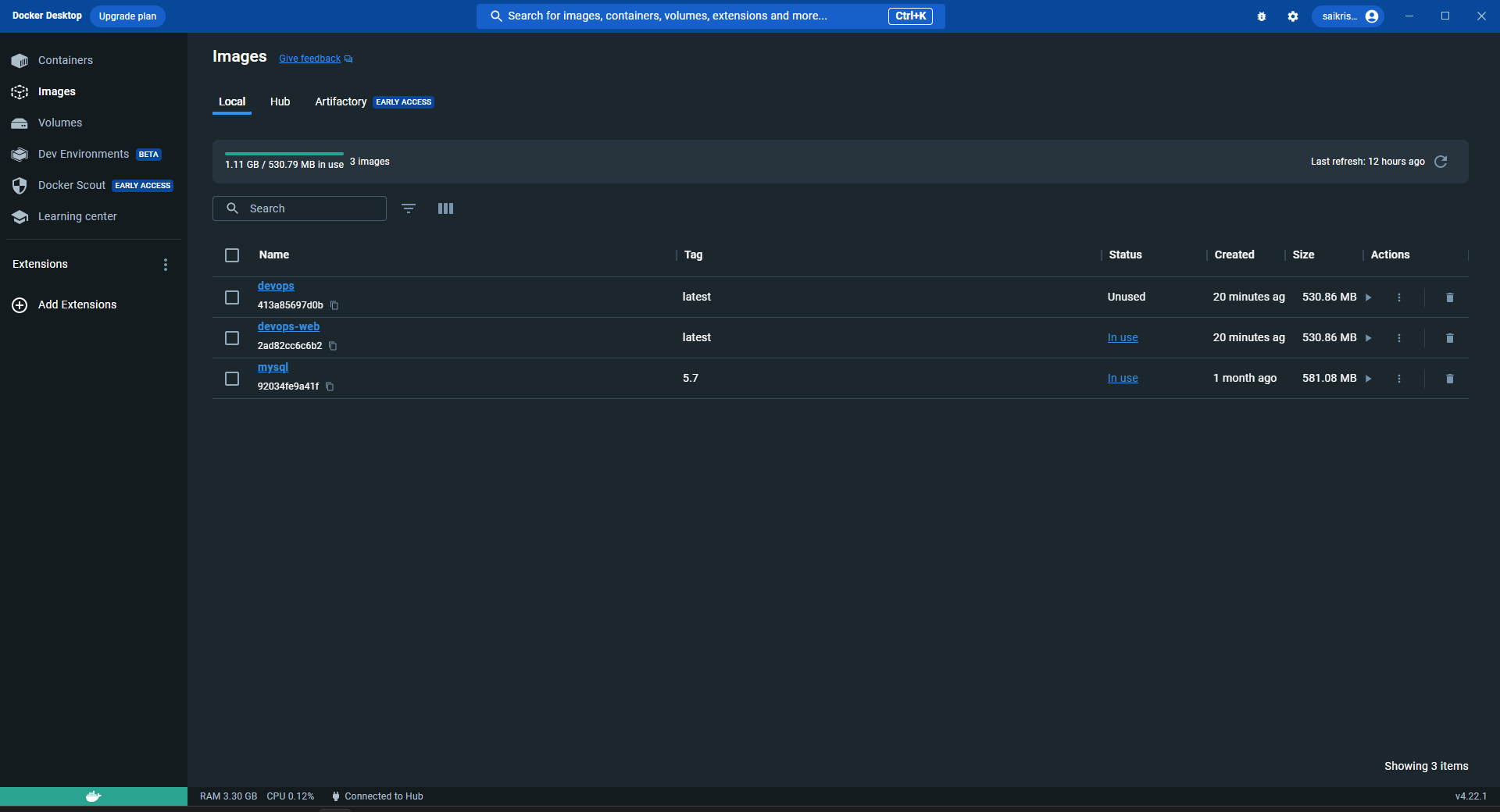
**3.9: CHECKOUT PAGE**



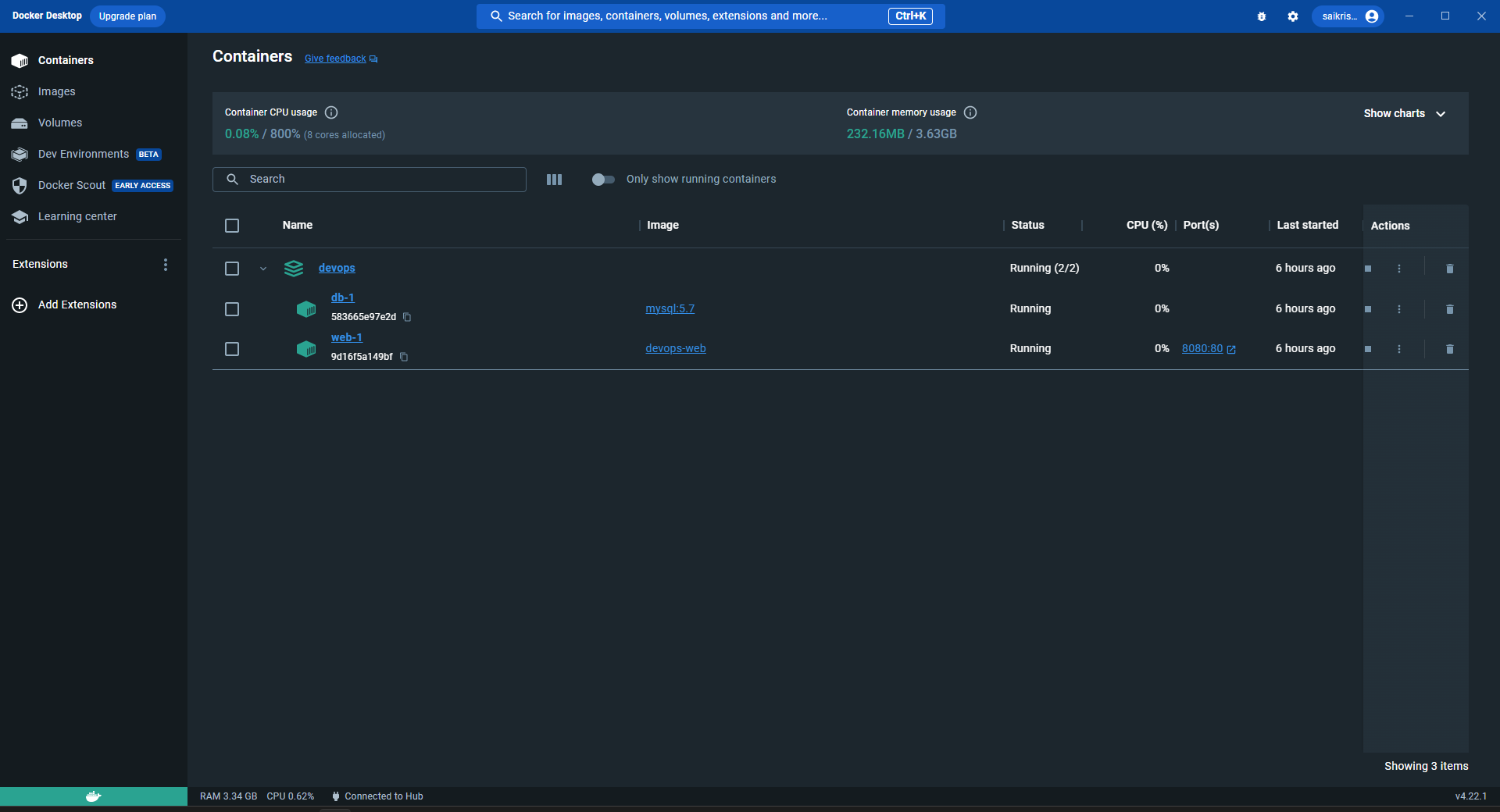
**3.10 Docker files**

****

**3.11 Docker Images**

****

3.12 Containers



Home Page:

* Layout: Top navigation bar containing logo, search bar, user profile/login icon, and shopping cart icon. The main area displays featured products, categories, and trending items.
* Features:
  + Go to the required section without much of hazzle.
  + Quick view option for products without navigating away from the homepage.

Product Detail Page:

* Layout: Product image(s) on the left, detailed description and purchase options on the right.
* Features:
  + Crisp description
  + Add to cart

User Authentication & Profile Management:

* Sign-up using email id.
* Sign in to buy products.

Shopping Cart & Checkout Process:

* Allows users to update quantities or remove items.
* Uses a form to take user’s payment.

3.2. Backend Architecture:

****Backend Routes:****

* Users:
  + POST /users/login: Authenticate and log in a user.
  + POST /users/register: Register a new user.
  + GET /users/profile: Fetch details of the logged-in user.

Middleware:

* Authentication Middleware: To verify the authenticity of a user making requests.
* Error Handling Middleware: Catches and handles errors, returning appropriate status codes and messages.

3.3. Database Schema:

* Users Collection:
  + Fields: Email, Password (hashed), Name, Address, Phone, PurchaseHistory, etc.

3.4. DevOps Flow

To ensure a smooth development-to-deployment flow, a DevOps pipeline is set up:

Version Control: GitHub repositories for frontend and backend codebases.

Continuous Integration (CI):

* Automated tests using GitHub Actions.
* Linting checks to maintain code quality.

Continuous Deployment (CD):

* Upon successful CI, the code is automatically containerized using Docker.
* The containers are then pushed to Docker Hub.
* Automated deployment scripts then pull these containers and deploy them on cloud infrastructure.

4. Implementation Details

Within the implementation details of this project, the frontend is crafted using a combination of HTML and CSS to create an engaging webpage design, while JavaScript is employed for robust form validation. On the backend, PHP is utilized to establish a secure login system, and MySQL plays a pivotal role in database creation and management, facilitated through MySQLi for enhanced database functionality and interactions. This comprehensive stack ensures a seamless user experience and efficient data handling for the application.

4.1. Frontend:

* + Used HTML ans CSS for the design of the webpage
  + Javascript for form validations.

4.2. Backend:

* PHP for login system
* MySQL for database creation
* MySQLi for database management

5. DevOps Implementation

In the realm of DevOps implementation, leveraging modern technologies and methodologies is paramount for streamlining software development and deployment processes. This section explores two key facets: containerization with Docker and CI/CD automation through GitHub Actions. These practices enable efficient packaging and distribution of applications while ensuring code quality and seamless deployment, culminating in the successful deployment of applications on cloud infrastructure like Amazon Elastic Container Service (ECS).

* 1. Containerization with Docker:
  + Created a Dockerfile to containerize the frontend, backend, and database services.
  + Used Docker Compose to manage multi-container applications.

5.2. CI/CD with GitHub Actions:

* + Set up workflows to automate testing and deployment.
  + Ensured code quality with linting and unit tests.

5.3. Deployment:

* + Used Docker Hub to store container images.
  + Deployed the application on AWS using Amazon Elastic Container Service.

References:

* www.w3school.com
* **"JavaScript: The Definitive Guide"** by David Flanagan - This book provides a deep dive into JavaScript, a fundamental language for full-stack development
* **"DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations"** by Gene Kim, Patrick Debois, John Willis, and Jez Humble - This handbook provides practical insights into DevOps implementation for improving software delivery processes
* [www.stackoverflow.com](http://www.stackoverflow.com/)
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.javapoint.com](http://www.javapoint.com/)
* [www.youtube.com](http://www.youtube.com/)
* <http://php.net/manual/en/function.ucfirst.php>
* https://www.w3schools.com/howto/howto\_google\_fonts.asp
* Docker Documentation: <https://docs.docker.com/>