

PHASE-WISE REPORT: SHOP EZ – E-COMMERCE WEB APPLICATION

Phase 1: Planning

Project Name:

SHOP EZ – A Single Page E-Commerce Web Application

Team ID: SWTID1743839665

Team Members:

- Triven Khoushi (Team Leader)
- Rohan Singha
- Azeem Ahamad
- Gurleen Kaur Makhija

Objective:

- Design and develop a responsive e-commerce platform focused on men's jeans.
- Create a modern single-page application using ReactJS, Redux, React Router, and Material UI.
- Simulate backend interaction with a static JSON file for dynamic rendering.
- Gain hands-on experience in frontend development and deployment.

Motivation:

- Address the growing demand for minimal, niche-specific shopping platforms.
 - Develop practical skills in full-stack development under SmartBridge mentorship.
 - Apply modern tools and frameworks in a real-world project environment.
-

Phase 2: System Design

Architecture:

- Frontend: SPA using ReactJS
- State Management: Redux & Redux Thunk
- Styling: Material UI, Bootstrap, Styled Components
- Routing: React Router DOM
- Data Source: db.json (Mock backend)
- Notification System: React Toastify
- Image Handling: React Simple Image Slider
- Testing: Jest, React Testing Library

Component Breakdown:

- Product listing, filter/sort modules, image sliders, notification pop-ups
- Redux store for global state management
- Axios for data fetching

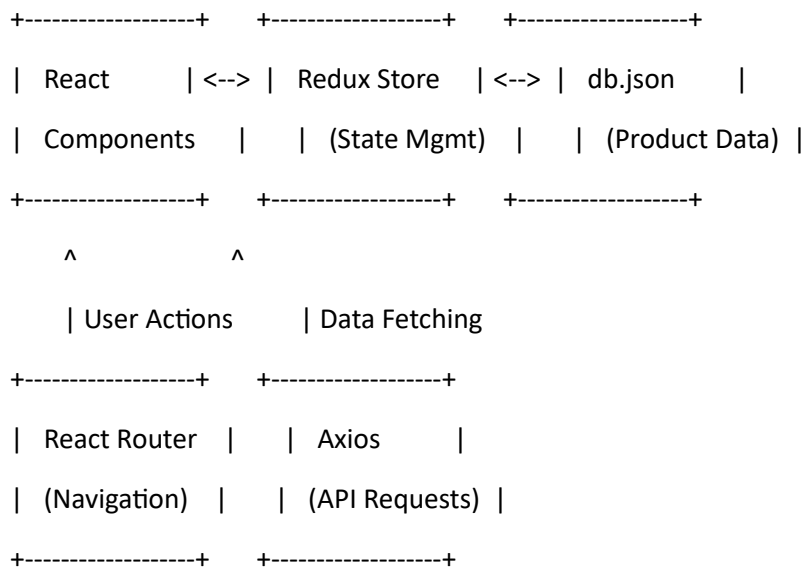
Data Flow:

1. Load product data from db.json via Axios
2. Store in Redux and render through components
3. User interactions (filter, sort, add to cart) update store and UI

Diagram Overview:

sql

CopyEdit



Phase 3: Implementation

Technology Stack:

- React v18.1.0
- Redux v4.2.0, Redux Thunk
- Material UI v5.6.4
- Bootstrap v5.1.3
- Styled Components v5.3.5
- Axios v0.27.2
- React Toastify v9.0.1

Scripts Used:

- start: Run development server
- build: Create production build
- test: Run test suite

Product Features:

- Dynamic listing of men's jeans
- Real-time price, size, rating display
- Image sliders for product images
- Sorting/filtering functionality
- Toast notifications for user actions

Sample Data Structure:

json

CopyEdit

```
{
  "id": 1,
  "name": "Men Navy Blue Slim Fit Jeans",
  "price": { "mrp": 1499, "discount": 60, "sp": 599 },
  "brand_name": "Roadster",
  "sizes": [28, 32, 34, 36, 38],
  "customer_rating": 3.9,
  "product_details": ["Clean look, stretchable fabric"],
  "images": ["<image-URL-1>", "<image-URL-2>"],
  "tag": "men-jeans"
}
```

Phase 4: Testing

Testing Tools:

- Jest
- React Testing Library
- User Event Library

Testing Achievements:

- Unit tests for component behavior

- Integration tests for Redux store updates
- UI responsiveness validation

Performance Tools:

- Web Vitals library to track metrics like load time and interactivity
-

Phase 5: Deployment

Platform: local server

Deployment Configuration: (Note: vercel.json was empty – needs completion for full deployment setup)

Deployment Type: SPA with route rewrites for seamless navigation

Build Script: npm run build

Phase 6: Evaluation and Review

Results:

- Functional UI with responsive product listing and real-time feedback
- Complete SPA structure using React ecosystem
- Mock data integration emulates backend functionality
- Scalable architecture with potential for backend extension

Limitations:

- No real backend (uses mock db.json)
- Missing deployment configuration file (vercel.json)

Suggestions for Future Improvements:

- Integrate backend (Node.js/Express + MongoDB)
 - Implement login/authentication
 - Add cart/checkout functionality
 - Enable user reviews/ratings
-

Phase 7: Individual Contributions

Triven Khoushi (Team Leader):

- Architecture design, task allocation, Redux setup, integration, deployment

Rohan Singha:

- Product display UI, filtering/sorting, image slider integration

Azeem Ahamad:

- Axios integration, loading indicators, toast notifications
Gurleen Kaur Makhija:
- UI design, styling with MUI/Bootstrap, accessibility, usability testing