

Cheapr FERN Application Development Progress

Functional Requirements Checklist

Status	Feature
Completed	User Authentication - Registration
Completed	User Authentication - Login
Completed	User Authentication - Log Out
Completed	All CRUD Functionality - Products
Completed	All CRUD Functionality - Breadcrumbs
Completed	All CRUD Functionality - Category
Completed	All CRUD Functionality - Cart & Users
Completed	API Integration - API routes are set up for managing users, products, and carts. Firebase is used for authentication and database management (real-time database).
Completed	Navigation &
Completed	Cart Functionality
Completed	Dynamic Product Pricing by Store
Completed	Price Savings Calculation
Not Complete	Responsive Design
Completed	Frontend-Backend Integration
Completed	Clear Cart Functionality
Completed	Quantity Adjustment for Products
Completed	Store Navigation and Breadcrumbs
Completed	Admin Dashboard
Completed	Product Image Upload
Completed	User Types (Admin & Normal) Changes
Completed	Sorting of Products
Completed	Flagging of Products
Completed	Admin Create Subcategory

Completed	Admin Create Product Type
Completed	Admin Creation Menu
In Progress	Preloaders & Loading Screens
In Progress	Are You Sure? Modals
In Progress	Form Validation
Not Completed	Testing & Bug Checking

System Documentation

Technical Architecture

The application is built using the FERN stack (Firebase, Express, React, Node.js), which includes the following components:

Firebase (Realtime Database & Authentication): Used for storing user data, product details, and managing real-time updates. Firebase Authentication manages user registration, login, and role-based access control.

Express.js: A Node.js framework used to build the backend API for handling requests and interacting with Firebase.

React.js: The frontend framework used to build the user interface, which interacts with the backend through API calls.

Node.js: Runs the server and handles routing, API requests, and serves the React frontend.

Data Flow:

Frontend (React): Users interact with the UI, performing actions such as viewing products, adding items to the cart, and adjusting quantities.

API Requests (Axios): The frontend sends requests to the Express backend (e.g., add product to cart, fetch product list).

Backend (Express/Node.js): The backend processes these requests and interacts with Firebase to fetch or modify data.

Firebase: The real-time database stores user data, product categories, products, and cart items. Firebase Authentication handles user management.

Frontend Rendering: Data from Firebase is sent back to the frontend for real-time updates and rendered for the user.

Database Schema:

Although Firebase is a NoSQL real-time database, the structure can be described in a hierarchical format. Below is an outline of the database structure and key relationships.

Users:

```
1354   "users": {
1355     "fDNqgeAQZsWUnZFgjS6tDD2CDMD2": {
1356       "email": "wolf@gmail.com",
1357       "firstName": "Wolf",
1358       "isAdmin": false,
1359       "lastName": "Botha",
1360       "userId": "fDNqgeAQZsWUnZFgjS6tDD2CDMD2"
1361     },
1362     > "w1VImCCfKkV3HtHBe5aYd8Sb2nr2": { ...
1368   }
1369 }
1370 }
1371
```

Categories:

```
1  {
2    "categories": {
3      "10000": {
4        "name": "food",
5        "subcategories": {
6          "10001": {
7            "name": "fruit_and_vegetables",
8            "types": {
9              "10002": {
10               "name": "fresh_fruit",
11               "productIds": [
12                 "-09_MjzDtke9i1C8juDq",
13                 "-09_0-0UmLUexg3JL6hq",
14                 "-09_0cD1SEFrs-uKpezD",
15                 "-09_RYNgRIMzU1cA5CDq",
16                 "-09_nS8tl5xE1knFbrt_",
17                 "-09_qcJuzL0SHe1x78cn",
18                 "-09_ynv3lRP3DttL_DA8",
19                 "-09i4n0vuXgsBy2HqB-9"
20               ]
            }
          }
        }
      }
    }
  }
```

Products:

```
281     },
282     "products": {
283 >       "approved": { ...
1252     },
1253 >     "flagged": { ...
1279     },
1280     "pending": {
1281       "-09i4n0vuXgsBy2HqB-9": {
1282         "adminDecisionDate": "2024-10-21 10:00:06",
1283         "amount": "1",
1284         "checkers": {
1285           "price": "656",
1286           "updated": "2024-10-21"
1287         },
1288         "created": "2024-10-21 10:00:06",
1289         "image": "https://storage.googleapis.com/cheapr-fc010.appspot.com/",
1290         "name": "Toets A",
1291         "pnp": {
1292           "price": "3434",
1293           "updated": "2024-10-21"
1294         },
1295         "spar": {
1296           "price": "345",
1297           "updated": "2024-10-21"
1298         },
1299         "unit": "kg",
1300         "woolworths": {
1301           "price": "123",
1302           "updated": "2024-10-21"
1303         }
1304       }
1305     },
1306 >     "rejected": { ...
1352   }
1353 },
```

API Endpoints:

Breadcrumb Endpoints

1. Get Breadcrumb by Product ID
 - Endpoint: `/breadcrumb/product/:productId`
 - Method: `GET`
 - Description: Retrieves the breadcrumb trail for a specific product based on the product ID.
 - Frontend Function: `getBreadcrumbByProductId(productId)`
 2. Get Breadcrumb by Group ID
 - Endpoint: `/breadcrumb/group/:groupId`
 - Method: `GET`
 - Description: Retrieves the breadcrumb trail based on a group ID, which can be a category, subcategory, or product type.
 - Frontend Function: `getBreadcrumbByGroupId(groupId)`
 3. Get Breadcrumb by Subcategory ID
 - Endpoint: `/breadcrumb/subcategory/:subcatId`
 - Method: `GET`
 - Description: Retrieves the breadcrumb trail for a specific subcategory.
 - Frontend Function: `getBreadcrumbBySubcategoryId(subcatId)`
 4. Get Breadcrumb by Product Type ID
 - Endpoint: `/breadcrumb/type/:typeId`
 - Method: `GET`
 - Description: Retrieves the breadcrumb trail for a specific product type.
 - Frontend Function: `getBreadcrumbByTypeId(typeId)`
-

User Authentication and Firebase Integration Endpoints

1. User Sign Up
 - Endpoint: Firebase authentication
 - Method: `POST`
 - Description: Signs up a new user and stores additional information in Firebase Realtime Database.
 - Frontend Function: `signUpUser(email, password, firstName, lastName)`
2. User Login
 - Endpoint: Firebase authentication
 - Method: `POST`
 - Description: Logs in a user and retrieves their details from Firebase Realtime Database.

- Frontend Function: `loginUser(email, password)`
 - 3. User Logout
 - Endpoint: Firebase authentication
 - Method: `POST`
 - Description: Logs out the currently authenticated user.
 - Frontend Function: `logoutUser()`
-

Product Management Endpoints

1. Create Approved Product (Admin Only)
 - Endpoint: `/products/approved`
 - Method: `POST`
 - Description: Creates a new approved product with form data, including image uploads.
 - Frontend Function: `createApprovedProduct(formData)`
2. Create Pending Product (User)
 - Endpoint: `/products/pending`
 - Method: `POST`
 - Description: Creates a new pending product awaiting admin approval.
 - Frontend Function: `createPendingProduct(formData)`
3. Get All Approved Products
 - Endpoint: `/products/approved`
 - Method: `GET`
 - Description: Retrieves a list of all approved products.
 - Frontend Function: `getAllApprovedProducts()`
4. Get Approved Products by Group ID
 - Endpoint: `/products/approved/group/:id`
 - Method: `GET`
 - Description: Retrieves all approved products under a specific category, subcategory, or product type.
 - Frontend Function: `getProductsByGroupId(id)`
5. Get All Pending Products
 - Endpoint: `/products/pending`
 - Method: `GET`
 - Description: Retrieves all products currently pending approval.
 - Frontend Function: `getAllPendingProducts()`
6. Update Product
 - Endpoint: `/products/:id`
 - Method: `PUT`

- Description: Updates product details, such as the name, image, and price.
 - Frontend Function: `updateProductById(productId, formData)`
 - 7. Update Product Prices
 - Endpoint: `/products/:id/prices`
 - Method: `PUT`
 - Description: Updates the price for a product in different stores.
 - Frontend Function: `updateProductPricesById(productId, priceData)`
 - 8. Approve Product
 - Endpoint: `/products/approve/:productId`
 - Method: `PUT`
 - Description: Moves a product from pending to approved status.
 - Frontend Function: `approveProductById(productId)`
 - 9. Reject Product
 - Endpoint: `/products/reject/:productId`
 - Method: `PUT`
 - Description: Moves a product from pending to rejected status.
 - Frontend Function: `rejectProductById(productId)`
 - 10. Delete Product
 - Endpoint: `/products/:id`
 - Method: `DELETE`
 - Description: Deletes a product (approved, pending, or rejected).
 - Frontend Function: `deleteProductById(productId)`
-

Cart Management Endpoints

1. Get User Cart
 - Endpoint: `/cart/:userId`
 - Method: `GET`
 - Description: Retrieves all items in the user's cart.
 - Frontend Function: `getUserCart()`
2. Add Product to Cart
 - Endpoint: `/cart/add`
 - Method: `POST`
 - Description: Adds a product to the current user's cart.
 - Frontend Function: `addToCart(productId, quantity)`
3. Update Product Quantity in Cart
 - Endpoint: `/cart/update`

- Method: **PUT**
 - Description: Updates the quantity of a product in the cart. If quantity reaches 0, the product is removed.
 - Frontend Function: `updateProductQuantity(productId, quantity)`
4. Clear User Cart
- Endpoint: `/cart/clear`
 - Method: **POST**
 - Description: Clears all products from the user's cart.
 - Frontend Function: `clearCart()`
-

Category and Subcategory Management Endpoints

1. Create Subcategory
 - Endpoint: `/categories/category/:categoryId/subcategory`
 - Method: **POST**
 - Description: Creates a new subcategory under a specific category.
 - Frontend Function: `createSubcategory(categoryId, subcategoryName)`
2. Get Subcategories by Category
 - Endpoint: `/categories/category/:categoryId/subcategories`
 - Method: **GET**
 - Description: Retrieves all subcategories for a specific category.
 - Frontend Function: `getSubcategoriesByCategory(categoryId)`
3. Create Product Type
 - Endpoint: `/categories/category/:categoryId/subcategory/:subcategoryId/type`
 - Method: **POST**
 - Description: Creates a new product type under a subcategory.
 - Frontend Function: `createProductType(categoryId, subcategoryId, productName)`
4. Get Product Types by Subcategory
 - Endpoint: `/categories/category/:categoryId/subcategory/:subcategoryId/types`
 - Method: **GET**
 - Description: Retrieves product types under a subcategory.

- Frontend Function:
`getProductTypeBySubcategory(categoryId,
subcategoryId)`

UI Design:

cheap

All Groceries

Food

Drinks

Household

Search for Products

+

Log Out

Add New Product

Cancel

Submit

Product Name

Cream Crackers

Weight / Amount

500

Unit

Grams

Image

Pick n Pay Price

☐ On Special

Checkers Price

☐ On Special

Spar Price

☐ On Special

Woolworths Price

R57.99

☒ On Special

Until 23/09/2024

Home

Groceries

Add a Product

Shopping List

cheap

Shop Smarter, Not Harder.

© 2024 Wolf Botha | 21100255 | Open Window Institute

cheapAdmin

All Groceries

New Products

Flagged

+

Log Out

Create a Product

Cancel

Create

Product Name

Cream Crackers

Weight / Amount

500

Unit

Grams

Image

Pick n Pay Price

☐ On Special

Checkers Price

☐ On Special

Spar Price

☐ On Special

Woolworths Price

R57.99

☒ On Special

Until 23/09/2024

Home

Groceries

Add a Product

Shopping List

cheap

Shop Smarter, Not Harder.

© 2024 Wolf Botha | 21100255 | Open Window Institute

cheapAdmin

All GroceriesNew ProductsFlagged

+

Log Out

All Products

Sort by Price Ascending

172 Products

Pick n PayWOOLWORTHSCheckersSPAR

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

All Groceries

New Products

Flagged Products

Add a Product

cheapShop Smarter, Not Harder.

cheapAdmin

All GroceriesNew ProductsFlagged

+

Log Out

Edit Product

CancelSave Changes

Product Name

Cream Crackers

Weight / Amount

500

Unit

Grams

Image

Pick n Pay Price

☐ On Special

Checkers Price

☐ On Special

Spar Price

☐ On Special

Woolworths Price

R57.99

☒ On Special

Until 23/09/2024

All Groceries

New Products

Flagged Products

Add a Product

cheapShop Smarter, Not Harder.

cheapAdmin

All GroceriesNew ProductsFlagged

+

Log Out

Flagged Products

Sort by Price Ascending

3 Products

Pick n PayWOOLWORTHSCheckersSPAR

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

Reported Reason

There is a spelling mistake on the word "apple".

Dismiss

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

Reported Reason

There is a spelling mistake on the word "apple".

Dismiss

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

R36.99

19/09/2024

Reported Reason

There is a spelling mistake on the word "apple".

Dismiss

All Groceries

New Products

Flagged Products

Add a Product

cheapShop Smarter, Not Harder.

cheapAdmin

All GroceriesNew ProductsFlagged


+Log Out

New Products

Sort by Price Ascending

22 Products

Click n PayWOOLWORTHSCheckersSPAR




Top Red Apples
1.5kg

R36.99
19/09/2024

R36.99
19/09/2024

R36.99
19/09/2024

R36.99
19/09/2024



Top Red Apples
1.5kg

R36.99
19/09/2024

R36.99
19/09/2024

R36.99
19/09/2024

R36.99
19/09/2024

✕

✓



Top Red Apples
1.5kg

R36.99
19/09/2024R36.99
19/09/2024R36.99
19/09/2024R36.99
19/09/2024



Top Red Apples
1.5kg

R36.99
19/09/2024R36.99
19/09/2024R36.99
19/09/2024R36.99
19/09/2024



Top Red Apples
1.5kg

R36.99
19/09/2024R36.99
19/09/2024R36.99
19/09/2024R36.99
19/09/2024

All Groceries

New Products

Flagged Products

Add a Product

cheap

Shop Smarter, Not Harder.

cheap

All GroceriesFoodDrinksHousehold

Search for Products


+Shopping CartLog Out

Cart Planner

ResetClear All

15 Products


Click n PayWOOLWORTHSCheckersSPAR




Top Red Apples (1.5kg)
R36.99

1


− + ✓ ✕




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99




Top Red Apples (1.5kg)
R36.99



Top Red Apples (1.5kg)
R36.99



Top Red Apples (1.5kg)
R36.99



Top Red Apples (1.5kg)
R36.99

Home







Groceries

Add a Product

Shopping List

cheap

Shop Smarter, Not Harder.



© 2024 Wolf Botha | 21100255 | Open Window Institute

cheapo

All GroceriesFoodDrinksHousehold

Search for Products






+🛒Log Out

Food > Fruits & Vegetables > Fresh Fruit

Fresh Fruit

Sort by Price Ascending






5 ProductsPick n PayWOOLWORTHSCheckersSPAR

	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024
	R36.99 Updated 19/09/2024	R36.99 Updated 19/09/2024	R36.99 Updated 19/09/2024	R36.99 Updated 19/09/2024
	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024
	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024
	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024	R36.99 19/09/2024

HomeGroceriesAdd a ProductShopping List

cheapo


Shop Smarter, Not Harder.



© 2024 Wolf Botha | 21100255 | Open Window Institute

cheapo

LoginSign Up




Shop like a pro

The best grocery deals in SA in one place.


[Find Deals](#)

Why cheapo




Crowd-driven Deals

Our community of users helps you find the best deals by updating product prices, so you never miss out.




Save Time & Money

Create a shopping plan, compare prices, and see the best route to maximise your savings every time you shop.



Historical Prices

Track prices over time, so you know exactly when to buy and when to hold off for better savings.




Local is Lekker

Our platform is proudly South African, and we love supporting local businesses since 2004.

Our Store Partners

Pick n PayWOOLWORTHSCheckersSPAR








Join Us & Start Saving Today

[Create An Account](#)

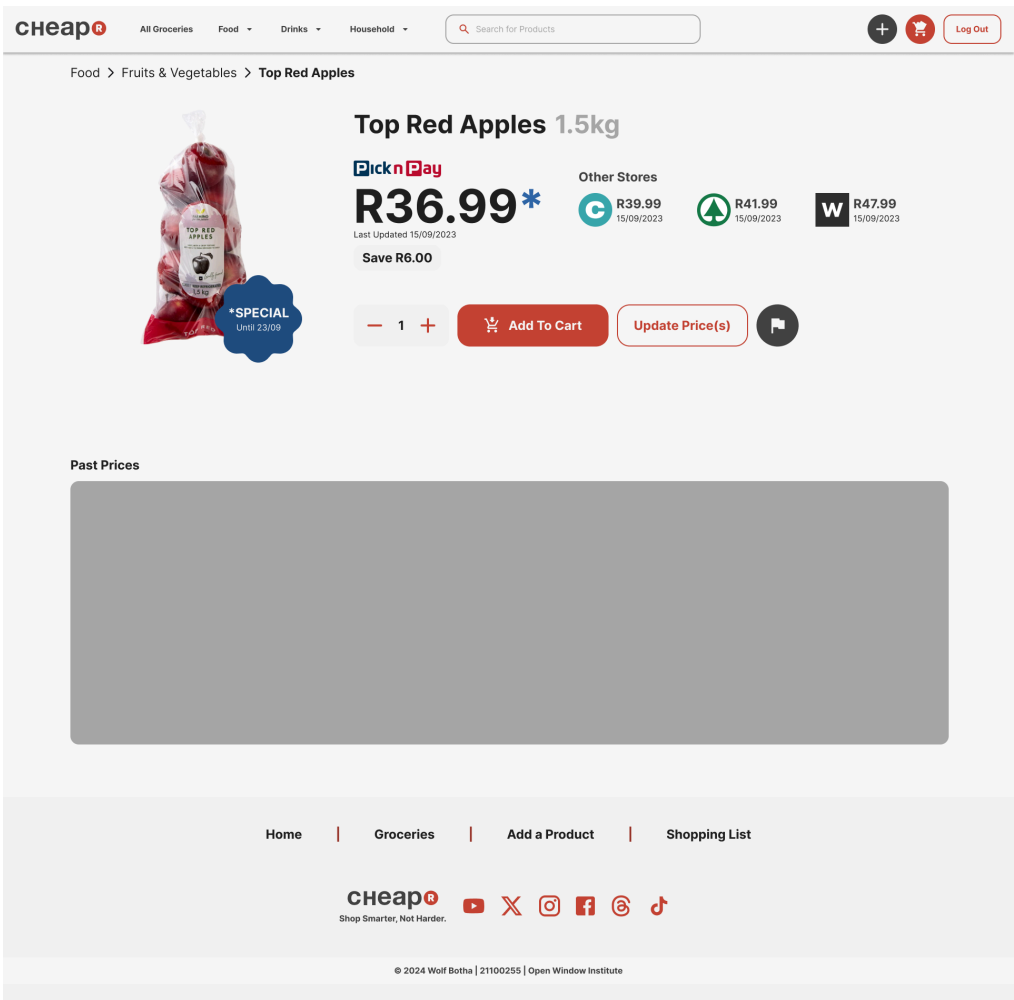
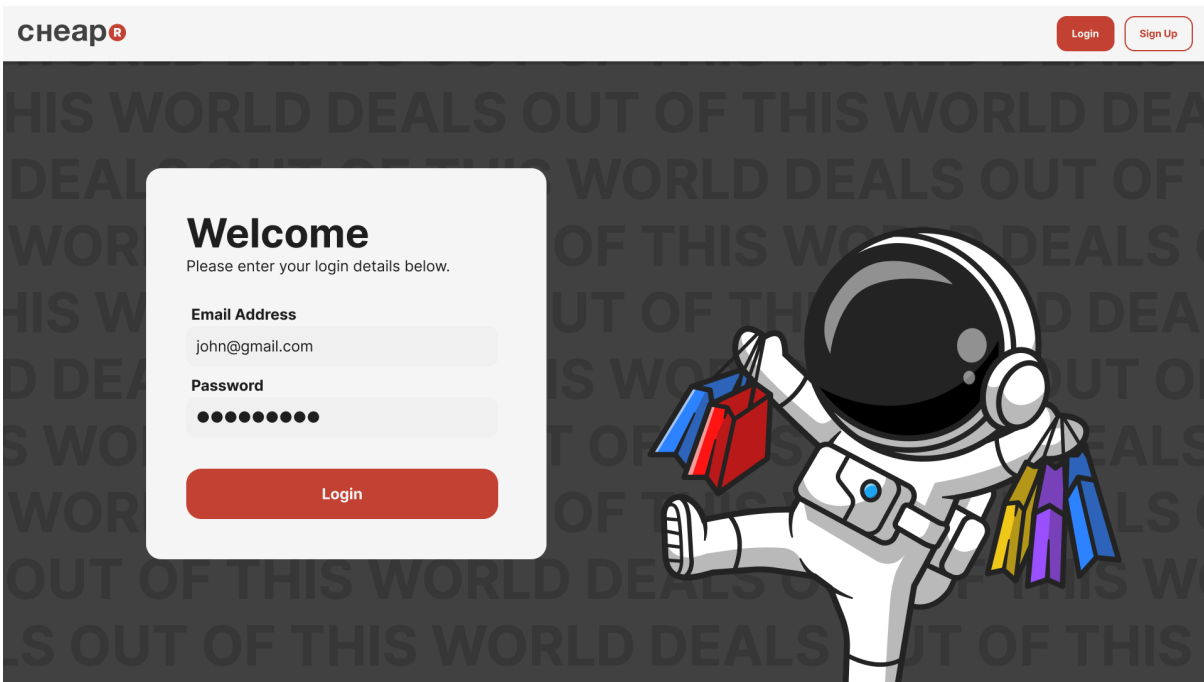
HomeGroceriesAdd a ProductShopping List

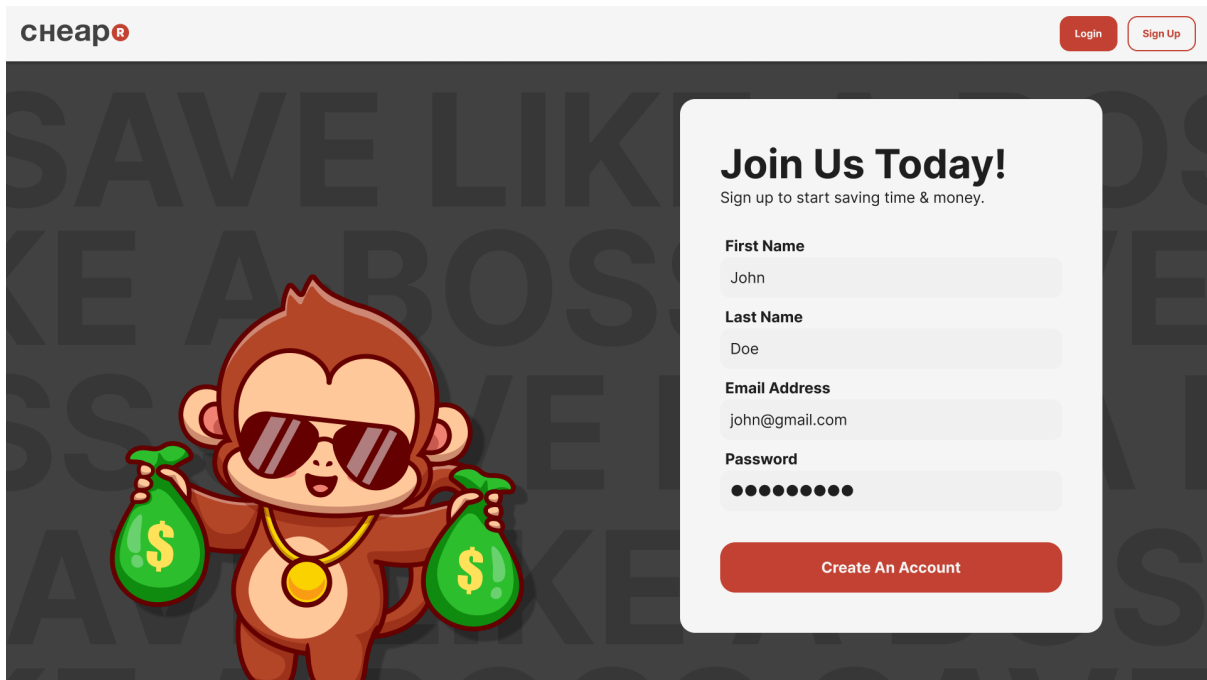
cheapo

Shop Smarter, Not Harder.



© 2024 Wolf Botha | 21100255 | Open Window Institute





Instructions for Running the Application

1. **Clone the Repository:**
 - Run `git clone <repository-url>` to clone the project to your local machine.
2. **Install Dependencies:**
 - Navigate to both the frontend and backend directories, and run `npm install` to install all dependencies.
3. **Configure Firebase:**
 - Create a Firebase project and configure Realtime Database and Authentication.

Add Firebase configuration in a `.env` file for both frontend and backend:
Bash.

Problem Statement:

In South Africa, the ongoing rise in food prices has put a significant strain on families, making it harder to afford daily essentials. Consumers face difficulties in comparing product prices across stores, often resulting in higher expenditures. This lack of transparency and price comparison options leads to unnecessary spending, disproportionately affecting lower-income groups. My application offers a solution by creating a platform that allows users to collaboratively share and compare the prices of groceries from various retailers. With features like product price tracking, personalised shopping lists, and the ability to find the most cost-effective shopping route, the platform simplifies the grocery shopping experience. By making price comparisons more accessible and user-friendly, the application helps individuals and families make better financial decisions, ultimately reducing the burden of food costs on households.