CMPS-350 Phase-2

Mohammed Faheem Ali Zaidi – 202109963  
Sheikh Hasin Ishrak – 202108209  
Nafin Ahsan Kabir Mahmoud – 201913152

Report

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Index** | **Page No.** |
|  |
| 1 | Data Model | 2 |  |
| 2 | Database Queries | 4 |  |
| 3 | User Interface | 16 |  |
| 4 | Contributions | 17 |  |

**Data Model**

A screenshot of a computer

Description automatically generated

1. **User**:
   * Represents a user.
   * Each user possesses a unique **id**, **username**, and **password**.
   * The “**type”** field indicates the type of user, i.e., a buyer, seller, or an admin.
   * Users can have some optional personal information such as **name**, **surname**, **money\_balance**, **company\_name**, or **bank\_account**.
   * Fields responsible for storing shipping address information are **contact\_person\_name**, **street**, **apartment\_suite\_number**, **city**, **state**, **zip\_code**, and **mobile\_number**, mainly for buyers.
   * The user has well defined relationships with purchases and products.
2. **Product**:
   * Represents an item listed for sal.
   * Each product has a unique **id**.
   * There are information fields such as **year**, **make**, **model**, **type**, **price**, **distance**, **image\_url**, and **isSold**, each carrying some information associated with the product.
   * There is a foreign key **seller\_id** referencing the **id** of the user [seller] who lists the product for sale.
   * A product has well defined relationships with users [sellers] and purchases.
3. **Purchase**:
   * Represents a transaction.
   * Each purchase has a unique purchase **id**.
   * There are foreign keys **buyer\_id** and **product\_id** referencing the **id** of the user [buyer] who made the purchase and the **id** of the product being purchased respectively.
   * It includes a **purchase\_date** to denote the date of purchase.
   * A purchase has well defined relationships with users [buyers] and products.

This model allows users to register, list products for sale, purchase products, as well as manage their information. It also maintains clear and concise relationships between users, products, and purchases to facilitate efficient e-commerce transactions.

**Database Queries**

1. GET /api/products Returns all Products

A screenshot of a computer

Description automatically generated

2. POST /api/products Creates a Product

A screenshot of a computer

Description automatically generated

3. GET /api/products/[type] Returns Specific Products

A screenshot of a computer

Description automatically generated

5. PUT /api/products/[id] Updates a Specific Product

A screenshot of a computer

Description automatically generated

6. DEL /api/products/[id] Deletes a Specific Product

A screenshot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated

7. GET /api/purchases Return all Purchases

A screenshot of a computer

Description automatically generated

8. POST /api/purchases Creates a Purchase

A screenshot of a computer

Description automatically generated

9. GET /api/purchases/[purchase\_id] Returns a Specific Purchase

A screenshot of a computer

Description automatically generated

10. PUT /api/purchases/[purchase\_id] Updates a Specific Purchase

A screenshot of a computer

Description automatically generated

11. DEL /api/purchases/[purchase\_id] Deletes a Specific Purchase

A screenshot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated

12. GET /api/users Returns all Users

A screenshot of a computer

Description automatically generated

13. POST /api/users Creates a User

A screenshot of a computer

Description automatically generated

14. GET /api/users/[user\_id] Returns a Specific User

A screenshot of a computer

Description automatically generated

15. PUT /api/users/[user\_id] Updates a Specific User

A screenshot of a computer program

Description automatically generated

16. DEL /api/users/[user\_id] Deletes a Specific User

A screenshot of a computer

Description automatically generated  
A screenshot of a computer

Description automatically generated

17. GET /api/stats/1-total-purchases Total Purchase Statistics\

A screenshot of a computer program

Description automatically generated

18. GET /api/stats/2-buyers-per-state Buyers Per State Statistics

A screenshot of a computer

Description automatically generated

19. GET /api/stats/3-top-products-on-sale Top Products Statistics

A screenshot of a computer

Description automatically generated

20. GET /api/stats/4-average-pricing Average Pricing Statistics

A screenshot of a computer

Description automatically generated

21. GET /api/stats/5-sold-to-unsold Sold/Unsold Statistics

A screenshot of a computer

Description automatically generated

22. GET /api/stats/6-top-sellers Top Seller Statistics

A screenshot of a computer

Description automatically generated

**User Interface**

The remaining is the same as phase-1, with all storage references now using the APIs.

**Contributions**

List of Accomplishments:

* Migration to Nextjs environment.
* Repo File Creation.
* 16 Data API Route Creations.
* 6 Statistics API Route Creations.
* Prisma Installation & Schema Creation.
* Seed File Creation.
* Nextjs Use Case UI Creation.
* Report Creation.
* Deployment [Currently Incomplete].

Contribution Table:

|  |  |
| --- | --- |
| **Name** | **Contributions** |
|  |
| Mohammed Faheem Ali Zaidi | Migration to Nextjs environment.  Repo File Creation.  16 Data API Route Creations.  6 Statistics API Route Creations.  Prisma Installation & Schema Creation.  Seed File Creation.  Nextjs Use Case UI Creation.  Report Creation. |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| Sheikh Hasin Ishrak | Populating seed.js with JSON data  Api/products/route.js  Api/stats /2-buyers-per-state /route.js  Api/stats /4-average-pricing/route.js  Api/stats /6-top-sellers/route.js  Global.css |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |  |  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |