Dr. Mucahid Kutlu

CMPS 350, L02

Project Phase 2

E-Commerce Platform: Phonie

Haris Khan, 202104099

AhmedFaseeh Akram, 202005685

Mohammed Al-Hasna, 202009492

Data Model and Seed.js Implementation

We now have 4 Json files: phones, admins, sellers and customers. We split the users.json into 3 parts. Here is the diagram for the Database used for the Phonie Website:
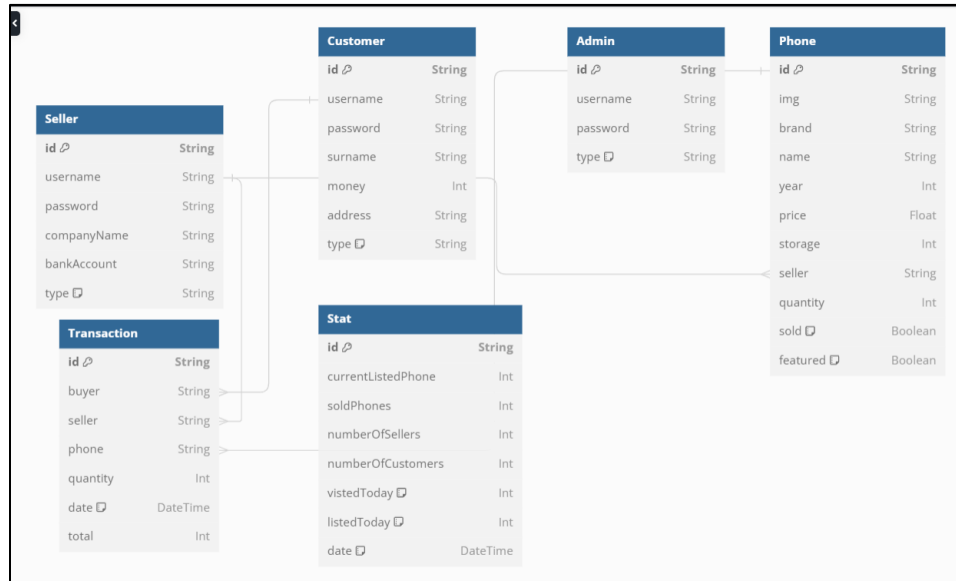

Figure 1. Data Model for Phonie

For seeding the database, we created a generic seed.js file that populates the database. We also have a getter.js file which runs when we run: "pnpm run dev" and automatically populates the database so we do not need to run the seed.js at all. We have kept both files in our project.
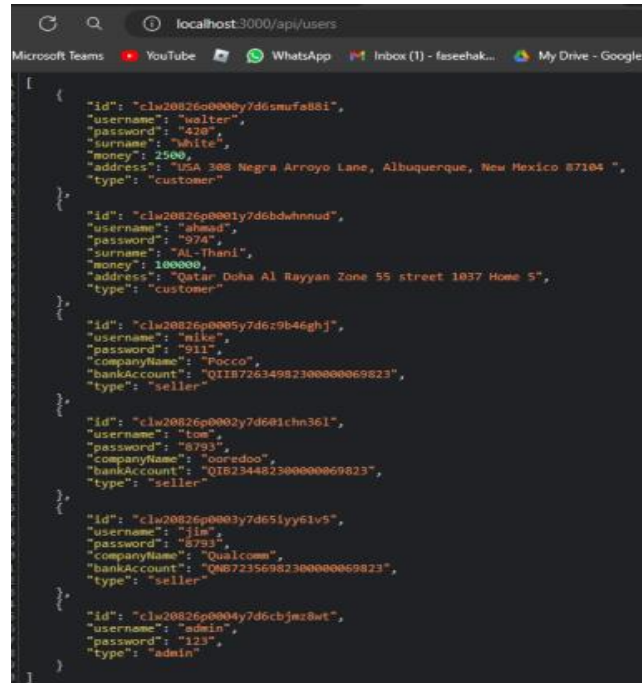
API and Repository

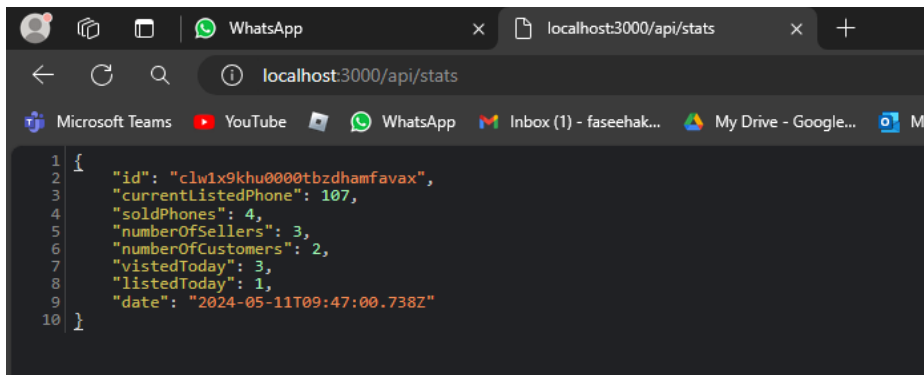So, we have 3 web API: stats, phones, users. Here the SS:

Figure 2. Phones API

Figure 3. Users API

Figure 4. Stats API

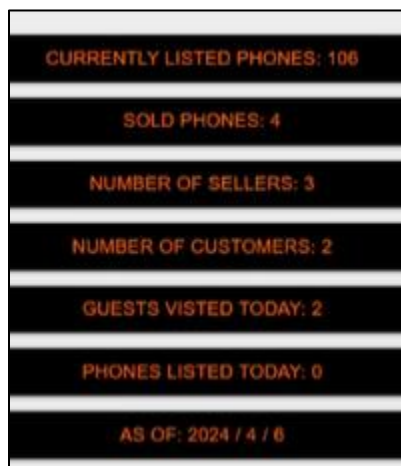We also have 3 repos: PhonesRepo and UsersRepo and a stats repo.

UsersRepo handles the 3 tables: Customers, Sellers, Admins, these are all users so that's why they are under 1 repository. They all use prisma functions which are using SQL queries.

Use Case: Statistics Next.js Page

For this new use case we had to display 6 statistics about our website. We accomplished this by creating:

- A route: /api/stats/route.js (has GET and PUT methods)
- A react page: /stats/page.jsx to display the page.
- A javascript file using prisma function code: /repos/stats.js

With these files we are able to fulfill the criteria to create a statistics page. Here is snapshot of the page:

Figure 5. The 6 stats

Our Statistic page is a table. Therefore we are able to use sql queries but it is only 1 entry. Therefore ID is fixed.

```
app > repos > JS stats.js > ⬡ getStats
 1   import prisma from "./prisma";
 2   const stat="clw1x9khu0000tbzdhamfavax"
 3   export async function addPhone(quan){
 4       await prisma.stat.update({
 5           where:{
 6               id:stat,
 7           },
 8           data:{
 9               currentListedPhone: {increment : quan},
10           }
11       })
12   }
```

Figure 6. Prisma Functions in stats.js

We have successfully created a Statistics page using react pages and prisma functions

UI Screenshots and Testing

The UI is mostly like Phase 1, but we added a new react page for stats as it was a requirement. This stats page is directly linked to the stats repo which uses Prisma functions. This stats will change according to the activity on the website:

Figure 7. Before and After buying Phones

As you can see, the stats get incremented and decremented accordingly after some activity has occurred on the Phonie website.

The rest of the pages are using html with repositories functions which are related to the database directly. We do not use localStorage for this phase and we only use the Database through Prisma client. Here is the html pages (using database, not localStorage):
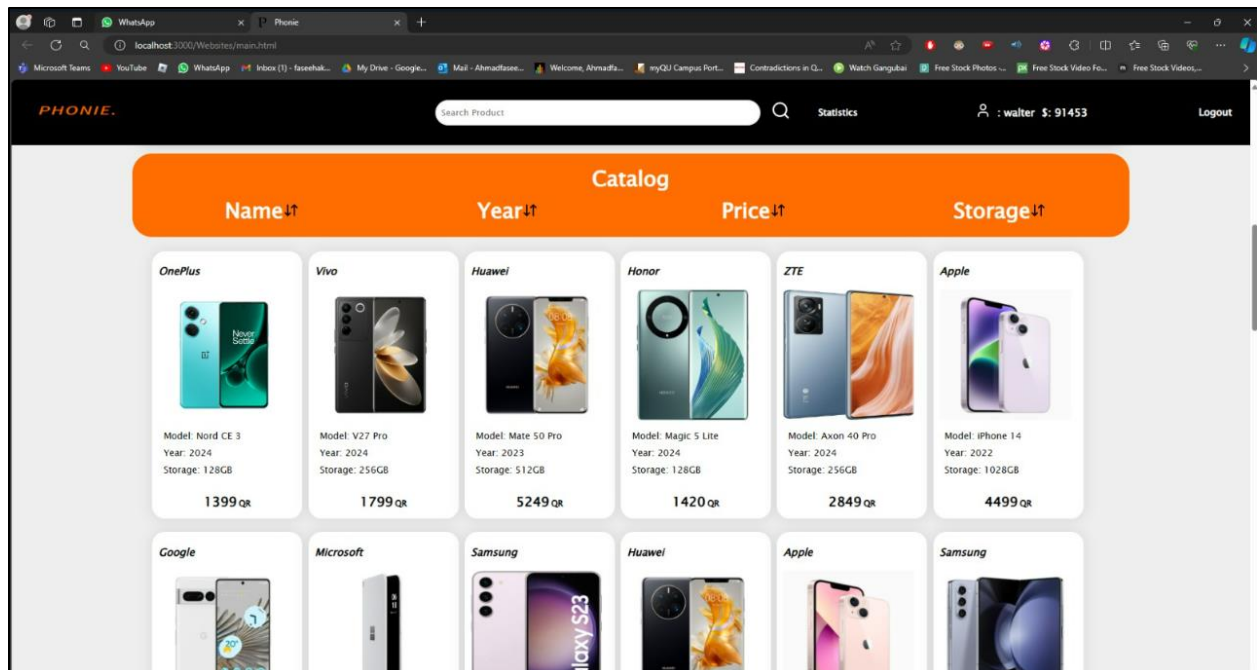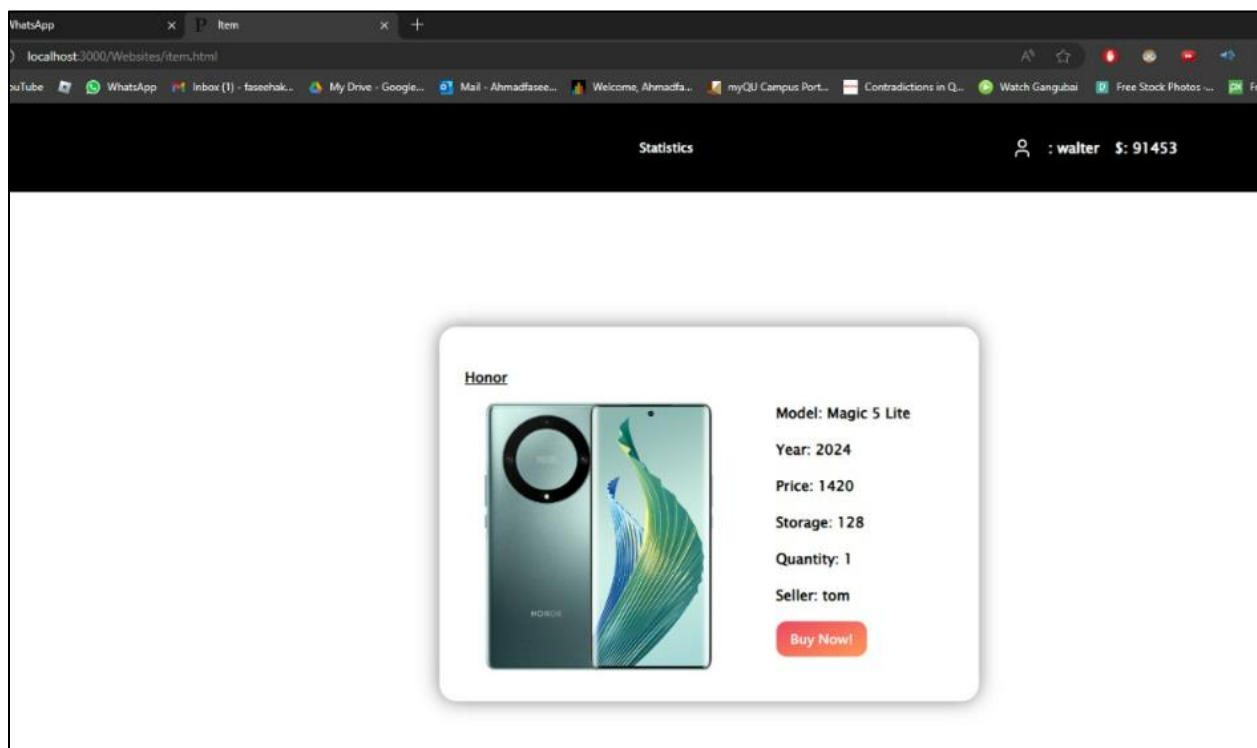
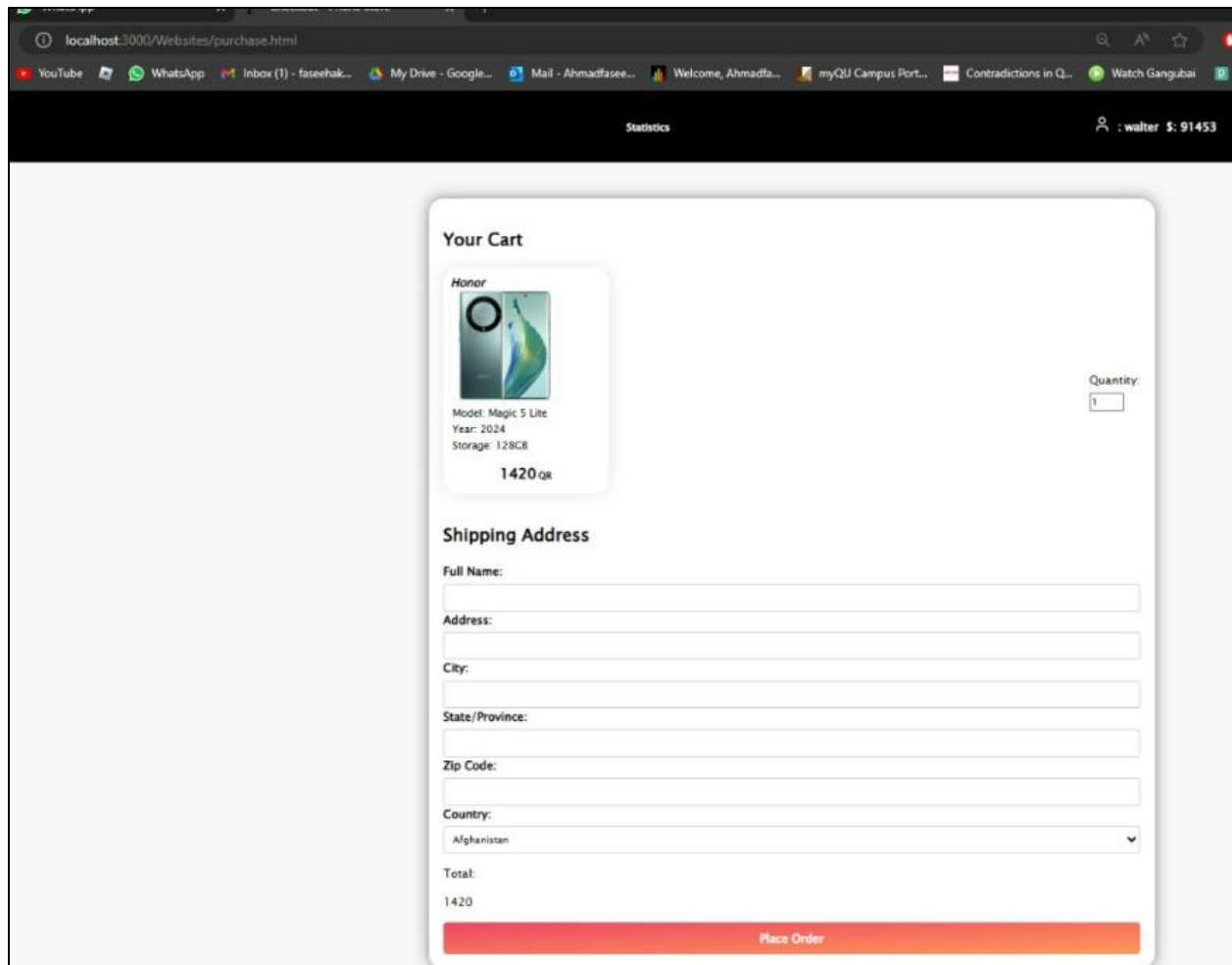*Figure 8. Home Page*

Figure 9. Items Page

Figure 10. Purchase Page

Figure 11. Sell Page