CMPS350 Web Development

E-Commerce Platform

Trend

1. QUID: 201805167 STUDENT NAME: Shama Al Ahmed Effort given: 33.3%
2. QUID: 202002679 STUDENT NAME: Fatima Saleh Effort given: 33.3%
3. QUID: 201804370 STUDENT NAME: Anfal Ahmad Anfws Effort given:33.3%

CMPS350 Web Development – Spring 2024

Sections: L52

Submitted: 11th May 2024

Submitted to: Dr. Mucahid Kutlu

**• DECLARATION:** We hereby certify that no part of this project or product has been copied from any other student’s work or from any other sources except where due acknowledgement is made in the project. No part of this project/product has been written/produced for us by any other person.

* **Introduction**

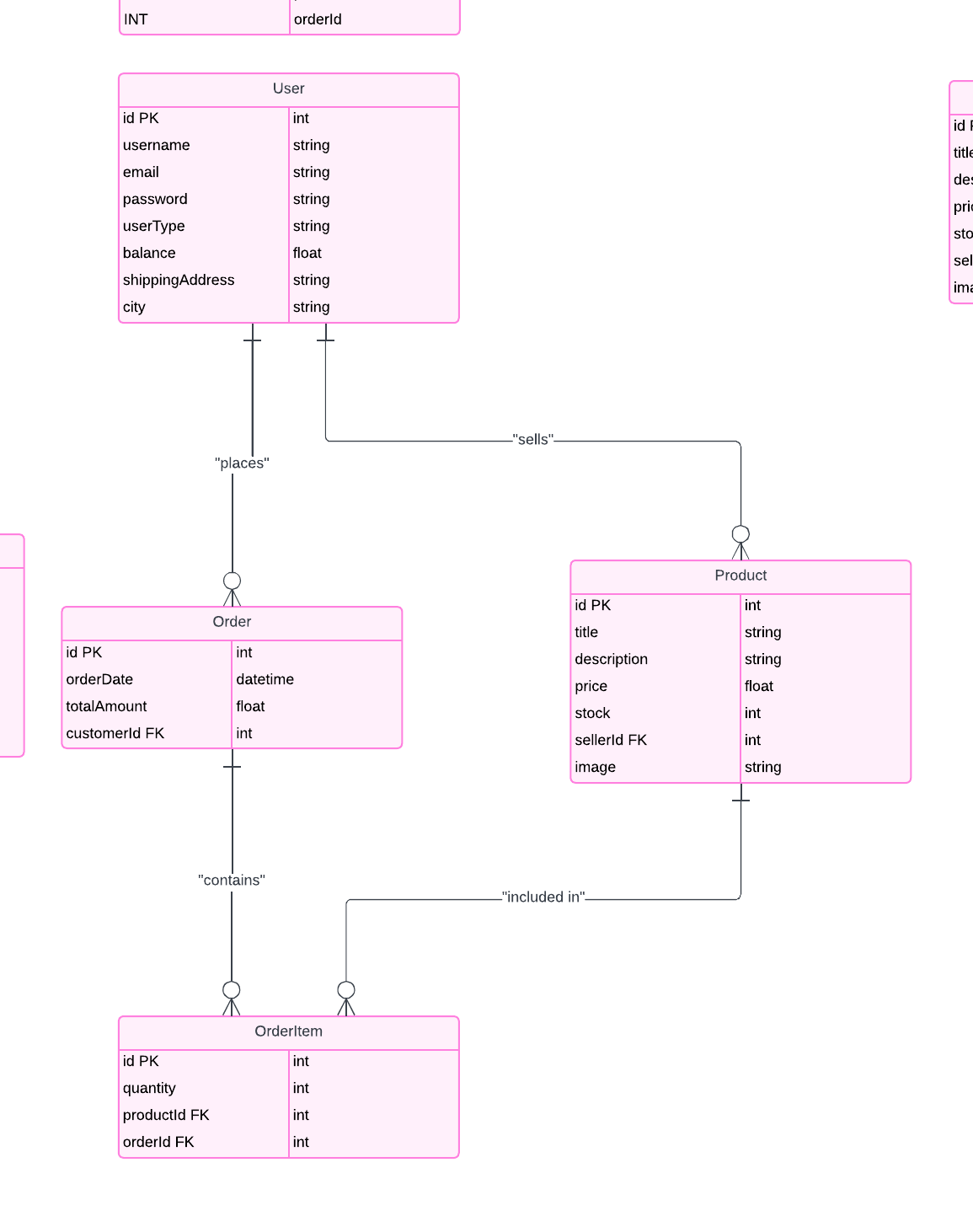
The E-Commerce Web App project seeks to improve the functionality and performance of our current application by including critical features and deep data analysis. This report describes the steps done to meet the project requirements indicated in the provided guidelines.   
  
During this phase of development, we focused on moving the data management system from local storage to a real database, implementing a strong data model, and integrating APIs to enable seamless data interaction. In addition, we launched a new use-case for extracting valuable statistics from our E-Commerce platform, which will contribute to a better knowledge of user behavior and market trends.

**Github link:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **%** | **Functionality**\* | **Quality of the implementation** | **Grade** |
| Design and implement the Data Model. | 10 | 10 |  |  |
| Init DB: populate the database with the data from the json files in seed.js | 5 | 5 |  |  |
| APIs and Repository Implementation to read/write data from the database | 25 | 25 |  |  |
| Statistics use-case with NextJS | 40 | 40 |  |  |
| **Documentation**   * Data Model diagram. * UI Design with screenshots and description. * Database queries. * Conducted tests and evidence. * **Contribution** of each team member [-10pts if not done] | 20 | 20 |  |  |
| **Total** | 100 | 100 |  |  |
| Bonus - successful deployment of the app and the Database to a cloud hosting service such as <https://vercel.com/> | 5 |  |  |  |
| Copying and/or plagiarism or not being able to explain or answer questions about the implementation. | 0 |  |  |  |

https://github.com/anfal-aa1804370/Web-projec

1. **Data Model diagram:**



1. **Database queries:**

* **Order:**

async saveOrder({ orderDate, totalAmount, customerId, orderItems }) {

    try {

      const newOrder = await prisma.order.create({

        data: {

          orderDate,

          totalAmount,

          customerId,

          orderItems: {

            create: orderItems.map((item) => ({

              quantity: parseInt(item.quantity),

              productId: item.id,

            })),

          },

        },

        include: {

          orderItems: true,

        },

      });

      return newOrder;

    } catch (error) {

      throw new Error(`Failed to save order: ${error.message}`);

    }

  }

  async getAllOrders() {

    try {

      const orders = await prisma.order.findMany({

        include: {

          orderItems: {

            include: {

              product: true,

            },

          },

          customer: true,

        },

      });

      return JSON.stringify(orders);

    } catch (error) {

      console.error(error);

      throw error;

    }

  }

  async getOrderById(id) {

    try {

      const orders = await prisma.order.findMany({

        where: {

          customerId: parseInt(id),

        },

        include: {

          orderItems: {

            include: {

              product: true,

            },

          },

          customer: true,

        },

      });

      return JSON.stringify(orders);

    } catch (error) {

      console.error(error);

      throw error;

    }

  }

  async getOrdersBySellerId(sellerId) {

    try {

      const orders = await prisma.order.findMany({

        where: {

          orderItems: {

            some: {

              product: {

                sellerId: parseInt(sellerId),

              },

            },

          },

        },

        include: {

          orderItems: {

            include: {

              product: true,

            },

          },

          customer: true,

        },

      });

      return orders;

    } catch (error) {

      console.error(error);

      throw error;

    }

  }

* **Products:**

  async getAllProducts() {

    const products = await prisma.Product.findMany();

    return JSON.stringify(products);

  }

  async getProductById(id) {

    const product = await prisma.product.findUnique({

      where: {

        id: parseInt(id),

      },

    });

    return JSON.stringify(product);

  }

  async getProductBySellerId(id) {

    const product = await prisma.product.findMany({

      where: {

        sellerId: parseInt(id),

      },

    });

    return JSON.stringify(product);

  }

  async saveProduct(product) {

    const { title, image, price, stock, description, sellerId } = product;

    const newProduct = await prisma.product.create({

      data: {

        title,

        image,

        price: parseFloat(price),

        stock: parseInt(stock),

        description,

        sellerId: parseInt(sellerId),

      },

    });

    return JSON.stringify(newProduct);

  }

* **User:**

async checkUser(obj) {

    const checkedUser = await prisma.user.findFirst({

      where: {

        username: obj.username,

        password: obj.password,

      },

    });

    return JSON.stringify(checkedUser);

  }

  async getAllUser() {

    const users = await prisma.user.findMany();

    return JSON.stringify(users);

  }

  async getUserById(id) {

    const user = await prisma.user.findUnique({

      where: { id: id },

    });

    if (!user) {

      throw new Error(`User not found`);

    }

    return JSON.stringify(user);

  }

1. **Document App Testing using screenshots.**

**A screenshot of a login page

Description automatically generated**

**A screenshot of a person's shirt

Description automatically generated**

**A screenshot of a website

Description automatically generated**

**A screenshot of a login page

Description automatically generated**

**A login screen with a person in the middle

Description automatically generated**

**here if the user not logged in and select item he will be directed to the login page**

**A screenshot of a person's shirt

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A close-up of a shirt

Description automatically generated**

**A screen shot of a checkout

Description automatically generated**

**A screen shot of a computer

Description automatically generatedA screen shot of a shirt

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a checkout

Description automatically generated**

**A screenshot of a website

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a person's shirt

Description automatically generated**

**A screenshot of a website

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a menu

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Summary Report – (Reflection):**

* **Anfal Ahmad Anfws:**

During this phase, we all worked closely together, using what we learned about React, Prisma, and Next.js to make our project better.

Together, we tackled each part of the project, sharing our skills to improve everything. Working as a team, we helped each other out and learned from each other along the way.

By collaborating, we gained valuable experience in building web apps and using new technologies. Our teamwork made our project stronger and more successful.

* **Fatima Saleh:**

In this project, I gained a lot of experience, and one of them is that I have improved my skills in web development, and how to work with HTML, CSS, JAVASCRIPT, React, next.js and Prisma. It was an enriching experience. I worked on every use case in this phase as it was challenging, and we needed to work together to be able to do it. I faced many problems during this phase, and I get some help from my teammates. Also, it was a good experience to work with teams to get from their experiences and if I have a problem, they are ready to help me to solve the problem. It was a fun experience to work with my teammates.

* **Shama Al Ahmed:**

In the second phase of the project, we improve and develop the platform with what we have learned from the last topics in the course such as using Prisma ,React ,etc…In this phase we mostly worked together, as this phase has advanced topic to implement, and a bit challenging, but this enhanced teamwork to express our ideas about what we understood from the course and share them together to ensure the success of this project.