Final Project Plan

1. Project Title:

Card Trader

2. Info about app

A website where users can buy and sell cards for trading card games such as Magic: the Gathering, similar to https://www.cardmarket.com/ or https://www.tcgplayer.com/. Users will be able to list their own cards for sale, browse available cards or search for a particular card.

The database will contain every card or card variant that can be sold, while sellers can create listings for a card. This means that potential buyers can easily find every listing for the same cards. The seller can indicate that they have multiples of the same card available in one listing.

When users find a card they want to buy, they can also browse other cards from the same seller. This is often useful since it is usually economical and practical to deliver multiple cards in one shipping.

Search results will list unique cards and the lowest available price for that card. Clicking a card shows more information and each listing for that card. (A full website would contain every card in the game, while this project will contain a small sample).

When a purchase is made, the app will track it's status:
1. When the purchase is completed by the buyer, the card is marked as ordered
2. The seller can mark the card as shipped
3. The buyer can then mark the card as delivered

If a user (the buyer) orders a card it will be marked as "ordered". The user that listed the card for sale can then mark it as "sent", after which the buyer can mark it as delivered.

User Privileges

Guests (users that aren't logged in) can browse and search the cards available for sale

Members (registered and logged in users) can browse and search, buy cards or list cards for sale. They can edit or delete their listings, browse their previous or ongoing sales or purchases and manage their account.

Admins can browse and search, add new cards to the database, see previous and ongoing sales from any user and change their status, and delete listings from any user

3. Scope

At a minimum, the app will let users browse, search, list and buy cards. The app will contain a small sample of cards to buy and sell.

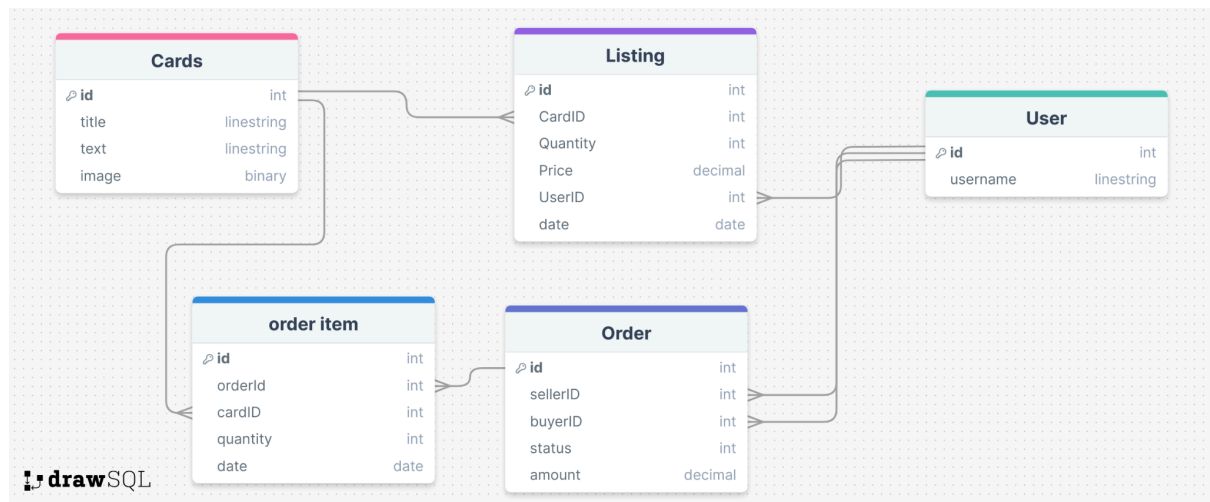If time is available, additional features may be added:

- **Credits:** Members can "buy credits" on the site, use it to buy cards, earn credits by selling cards, and "withdraw money". Credits are removed from the buyer's account when they make an order but only added to the seller's account when the order is delivered to prevent scamming. The site charges a small fee when a purchase is made and when money is withdrawn.

- **User Review:** After buying a card, a buyer can review the seller, to let other users know if the ordered cards were actually delivered, if they were damaged etc. This helps other buyers find trusted sellers and discourages dishonest sellers

- **Wishlist:** Let's a user create a list of cards they want and find a user with many or all of these cards for sale.

4. Project Template

Backend
Authentication and Authorization using ASP.NET Identity

SQL

Pages:

Visible for everyone:
Main page shows every card and the cheapest available price
Search Results shows cards that match and the cheapest available price
Card Page shows the card's image and text and every listing for the card.
User Listing Page shows every listing from the seller.

<u>Visible for members</u>
Orders Page shows the users completed and ongoing orders. Status can be changed.
Checkout Page lets users complete an order.
Account Page let's users handle their account.

<u>Visible for admin</u>
Card Database lets admin add, edit or delete cards
User List lists every user.

Members can buy cards from any page that displays listings. They can also edit or delete their own listings. Admins can delete any listing.

Header
<u>Guest</u>
- Home button
- Search Bar
- Login
- Register

<u>Member</u>
- Home Button
- Search Bar
- My listings
- My orders
- Account
- Log out

<u>Admin</u>
- Home button
- Search Bar
- Card Database
- User list
- Account
- Log Out

Footer
Social Media Links
Link to About Page

5. System

Frontend: **React**
Backend: **ASP.NET Core Web API**

6. Time plan

What do to	April 2024 Week 18					May 2024 Week 19					May 2024 Week 20				
	29	30	1	2	3	46	7	8	9	10	13	14	15	16	17
Planning	Planning	Planning		Planning											
SQL				Sql	Sql	Sql	Sql								
Backend				Backend	Backend	Backend	Backend								
Frontend								Frontend		Frontend	Frontend	Frontend			
Testing							Testing	Testing				Testing	Testing		
Flex													Flex	Flex	
Presentation															Present