Final Project Proposal:

Joystick Journeys

4/4/2025
SE / Comp Sci 3190 / IP_24

Henry McMahan
Camden Beightler

Table of Contents

1. Introduction

We, Henry McMahan (henrymcm@iastate.edu) and Camden Beightler (cmbeight@iastate.edu), are on team IP_24. McMahan has experience with HTML and JavaScript from previous coursework and has been learning to integrate this knowledge with CSS to create websites from in-class activities in COM S 3190. Beightler, has some basic HTML, CSS and JavaScript skills from prior classes. Using these skills that have been expanded upon in COM S 3190, he has made some websites with interactive elements and plans to bring those skills into this project.

We knew some sort of media catalogue or storefront would work great with our skills and would be fun to develop for us. We decided to create Joystick Journeys because we both have a love for video games, and together we believe that we are capable of creating a great website.

2. Purpose of the Proposal

For our midterm project, we have chosen to create "Joystick Journeys", a website that serves as both an informational and commercial platform for video game enthusiasts. This decision was influenced by our shared passion for gaming and interest in web technologies. Video gaming has evolved into the dominant entertainment industry, and our website will serve as a comprehensive resource for both historical and modern gaming content.

We aim to create an engaging and interactive website that allows users to:

- Explore and find a game that might suit their tastes.
- Browse lists of curated selections of games.
- View a ranked list of the top-selling games of all time.
- Purchase video games (physical and/or digital versions) through an online storefront.

3. Goals and Objectives

Goals

- Develop a visually appealing, user-friendly website about video game history and culture.
- Provide an interactive and informative experience for users interested in gaming.
- Demonstrate proficiency in React, Node.js, and MySQL by creating a functional web-based platform.
- Implement a structured and easily navigable interface for browsing gaming content.

Objectives

- Design and develop a single page with multiple interactive views, focusing on different types of video games.
- Implement a store section where users can browse and potentially purchase games.
- Functional cart, payment, and order confirmation system.
- Have options for filtering or sorting games.
- Integrate visual assets like game covers, console images, and infographics.

4. Project Description

Joystick Journets will have the following key components making up our website for video game enthusiasts:

Website Sections:

1. **Home Page** – The landing and main page of the website. Features include a game showcase carousel, display of games with filters and buttons to control the category, and an about page for the website.
2. **Game page** – Displays the specific game and more details about it.
3. **Shopping Cart –** Displays games in the shopping cart, and a total cost
4. **Checkout –** Similar to the shopping cart page but replaces games for forms the user can input for payment details.
5. **Order Confirmation –** Thanks the user, displays details about the order, shows what games were purchased, and shows some similar games the user might like.

6. **Sign-In / Sign Up** – Simple login page that would bring the user to the user profile once signed in. The profile is important for storing order data.
7. **User Profile** – Once signed in brings page showcasing details about the user and their order history
8. **Team** – Shows info about the team (students) that made the website

User Interaction & Design Considerations:

- A clean, responsive layout with easy navigation.
- A filter function to help users find specific games or consoles.
- Hover effects and animations for a modern UI experience.

5. Project Path Selection

As a team we decided on Option 1: to extend the midterm project. We liked the work we did on the midterm project, but there were still a lot of improvements we could make to the site. One of the big improvements we plan to add is making the entire site a working storefront, with a shopping cart, payment, and order confirmation screens.

Redevelopment Strategy:

- ❖ Convert previous pages into react components (classicGames.html → ClassicGames.jsx)
- ❖ Make components of other smaller page pieces
  - ➢ Navbar → Navbar.jsx
  - ➢ Footer → Footer.jsx
  - ➢ Game cards → GameCard.jsx
- ❖ Data is pulled from RAWG API instead of a data.json file
  - ➢ Implement React code to pull from RAWG

For new additions:

- ❖ Add shopping cart to navbar
- ❖ Node.js will help with the backend for game data and store functionality
- ❖ New react components for new page views (cart, checkout, order confirmation, etc.)
- ❖ MySQL database will keep track of (games, users, cart items, orders)

6. Feature Ownership & Responsibility

Feature Name: (e.g., Login Page, Product Management, Order History)
• Feature Description: What it does and why it's important
• Assigned Developer: Full name of the team member responsible

• Tech Involvement: Confirm that the developer will handle both frontend (React) and backend (Node.js/Express) aspects of this feature

Rawg.io implementation

- Feature Description: The Rawg API will be used to access a large video game database. It will be implemented so the data can be seamlessly stored into our websites database and then easily accessed and displayed on any store page, or if a user searches for an item.
- Assigned Developer: Camden Beightler
- Tech Involvement: Frontend will be used to have the content from the API displayed in a nice format for the user. Backend will be used to request data from the API, store the data into the database, and then send data to the frontend.

Shopping Cart / Checkout

- Feature Description: Functional shopping cart that users can add and remove games to. They can checkout and pay to receive the games and be brought to the Order Confirmation screen.
- Assigned Developer: Camden Beightler
- Tech Involvement: The front end will provide a clean UI to add games to the cart as well as remove them, and then the cart's items can then be viewed at checkout. Finally the user has a UI to input payment information and confirm the payment to receive the games. The backend will manage all the data being sent to the cart, ensuring that the cart only contains the items added, and correctly removes them when a remove is requested. Then at checkout the data inside the cart will be totaled at checkout.

Login / User Signup

- Feature Description: Login/Sign up page that users can use to first create an account on our website, and then login to that account in future uses. This way their information can be saved for future uses, such as games owned, game wish list payment information.
- Assigned Developer: Henry McMahan
- Tech Involvement: Use of front end for the user to input account information, and backend to then appropriately store that data within the database to be accessed at any time when needed.

Order Confirmation

- Feature Description: Order confirmation screen shows that games were purchased, has similar games section, and links back to homepage.
- Assigned Developer: Henry McMahan
- Tech Involvement: Front end would have to communicate with cart / checkout data to see what was purchased for the showcase. Backend would request data from Rawg.io to showcase similar games and make use of MySQL for data related to the user and order history.

Game catalog browser

- Feature Description: Main display of games that users can sort by certain lists of games. The user can also filter based on things like price, console, release year, etc. Clicking on a game would bring the user to a separate screen that would display most of the details about the game and have a button to add to cart.

- Assigned Developer: Henry McMahan, Camden Beightler
- Tech Involvement: On the front-end game art and price details would have to be showcased. The game screen would show similar details from data that would have to be retrieved from the backend, with use of MySQL and the Rawg.io API.

7. Resources and Tools

- Web development programming languages: HTML, CSS, JavaScript.
- External API Rawg.io provides a database of over 500,000 video games, we can use this to display games to be sold on our website.
- Use of Discord for effective team communication and collaboration.
- Careful time allocation to ensure work is consistently getting done on a week-by-week basis, so there is no crunch time nearing the deadline.
- Use GitHub as our version control remote repository to push and pull our work to keep our team's work up to date and allow members to work from anywhere without need for direct in-person collaboration.
- Use MySQL to manage and store user data flowing into the website for user login and purchasing information.
- Node.js can be used to handle fetching data from rawg.io for user searching.
- Use React to handle our UI, allowing for a smoother user experience.
- Use Bootstrap and to provide a well-constructed catalog template to display our video games on multiple pages in a user-friendly arrangement.
- Use tailwind to create a much sleeker and modern looking version of our website
- Use Excalidraw to create wireframes for the pages of our website. By doing this we can make a roadmap for when we start the implementation phase and ensure that it will be neat and user friendly.
- Make use of GitLab's features to keep track of commits made to the repository.
- Discord will be used to keep track of current tasks that need to be completed by each weekend.
- Work will be divided up evenly and posted on our private discord channel and pinned to it in order to keep track of what needs to be done and what has been done.

8. File Structure and Project Organization

• frontend/ – Contains all React-related files.
  – src/
    ∗ assets/ – Images and external data.
    ∗ components/ – React components.
    ∗ pages/ – Full page views like login, Store, gameinfo, checkout.
    ∗ api/ – Helper functions for calling RESTful APIs (login, fetch games).

  * App.jsx – Application
  * main.jsx – for ReactDOM.

- backend/ – Express server, API routes, and database logic.
  * config/ – Configuration files (DB connection, environment).
  * routes/ – routes (auth.js, games.js, checkout.js).
  * controllers/ – route handlers
  * models/ – Database access and queries.
  * middleware/ – Authentication, and logging
  * index.js – app entry point.

- Documents/ – Planning sketches, Software Architecture Document, final report, and video.
  * sketches/ – UI sketches and wireframes.
  * architecture.pdf – Software Architecture Document.
  * final_report.pdf – Final written report.
  * demo_video.mp4 – Final project demo.

- .env – Environment variables
- .gitignore – Ignore node_modules, .env, etc.
- package.json – project dependencies.
- tailwind.config.js – Tailwind CSS configuration.

9. Data Sources and Management

Our current plan to access data is through Rawg.io, a large game database. We will access it from the backend fetching and storing data within MySQL, so whenever a user searches for a game it will grab the data and pull it to the website.

Here is an example of a JSON file from information pulled from our API, Rawg:

```
{
  "results": [
    {
      "id": 22509,
      "slug": "minecraft",
      "name": "Minecraft",
      "released": "2011-11-18",
      "background_image": "https://media.rawg.io/media/...jpg",
      "rating": 4.48,
      "platforms": [
        { "platform": { "name": "PC" } },
        { "platform": { "name": "PlayStation 4" } }
      ],
      "genres": [
        { "name": "Sandbox" },
        { "name": "Survival" }
      ]
    }
  ]
}
```

This would be an example of information provided to us from Rawg. It gives all important game info and is stored within the JSON file to be displayed to the user.

Here is how we plan to implement CRUD:

First, with Create, we will add the game to the store.
Step 1: User interacts searching for a game (frontend).
Step 2: Backend then sends a request for the data.
Step 3: Data is retrieved and then saved to MySQL.
Step 4: Finally the data is converted to the JSON format, now the game is offically created and useable.

Dataflow:

- Game is selected from Rawg API
- Backend retrieves data and then inserts it into the database

Next with Read, we get the information for the user.
Step 1: User visits the store and makes a search (frontend).
Step 2: The backend queries the database and then retrieves the info.
Step 3: MySQL queries the games information table.

Dataflow:

- Users can view a list of games, or click on a game to view its information.
- Backend fetches data from MySQL database and returns data to the frontend

Next with Update, we edit the game information.
Step 1: First some data about a game may be changed, such as a price change. (Frontend)
Step 2: The backend will the process the update request and update it in the MySQL.
Step 3: MySQL sends an update query to the games information table.

Dataflow:

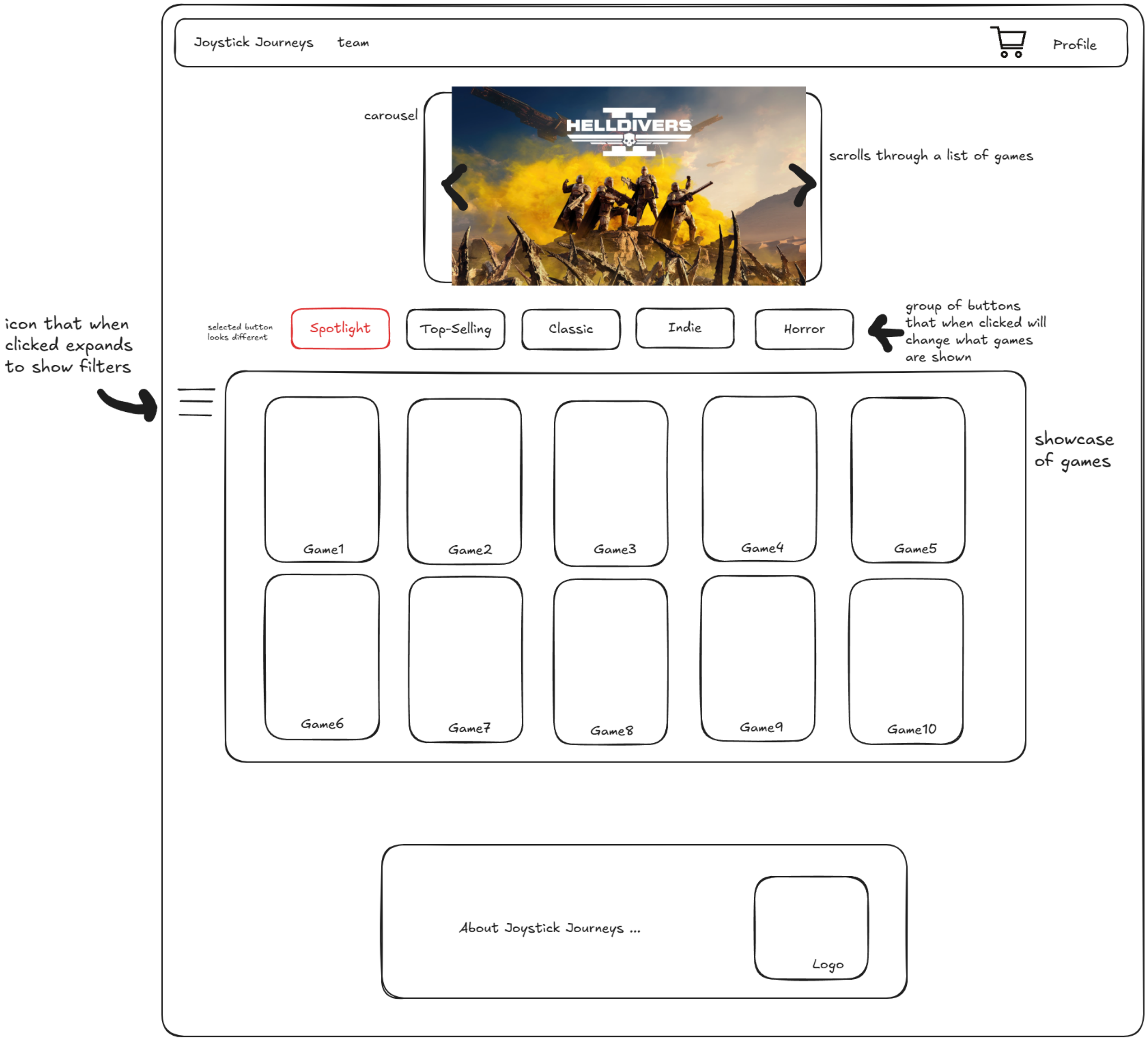
- Game information is changed.
- The backend receives the updated information and updates the database.

Finally with Delete, we remove games from the store.
Step 1: A game is deleted and all information along with it. (Frontend)
Step 2: The backend will then request a delete, specifying what game to delete.
Step 3: MySQL receives it and removes it from the table.

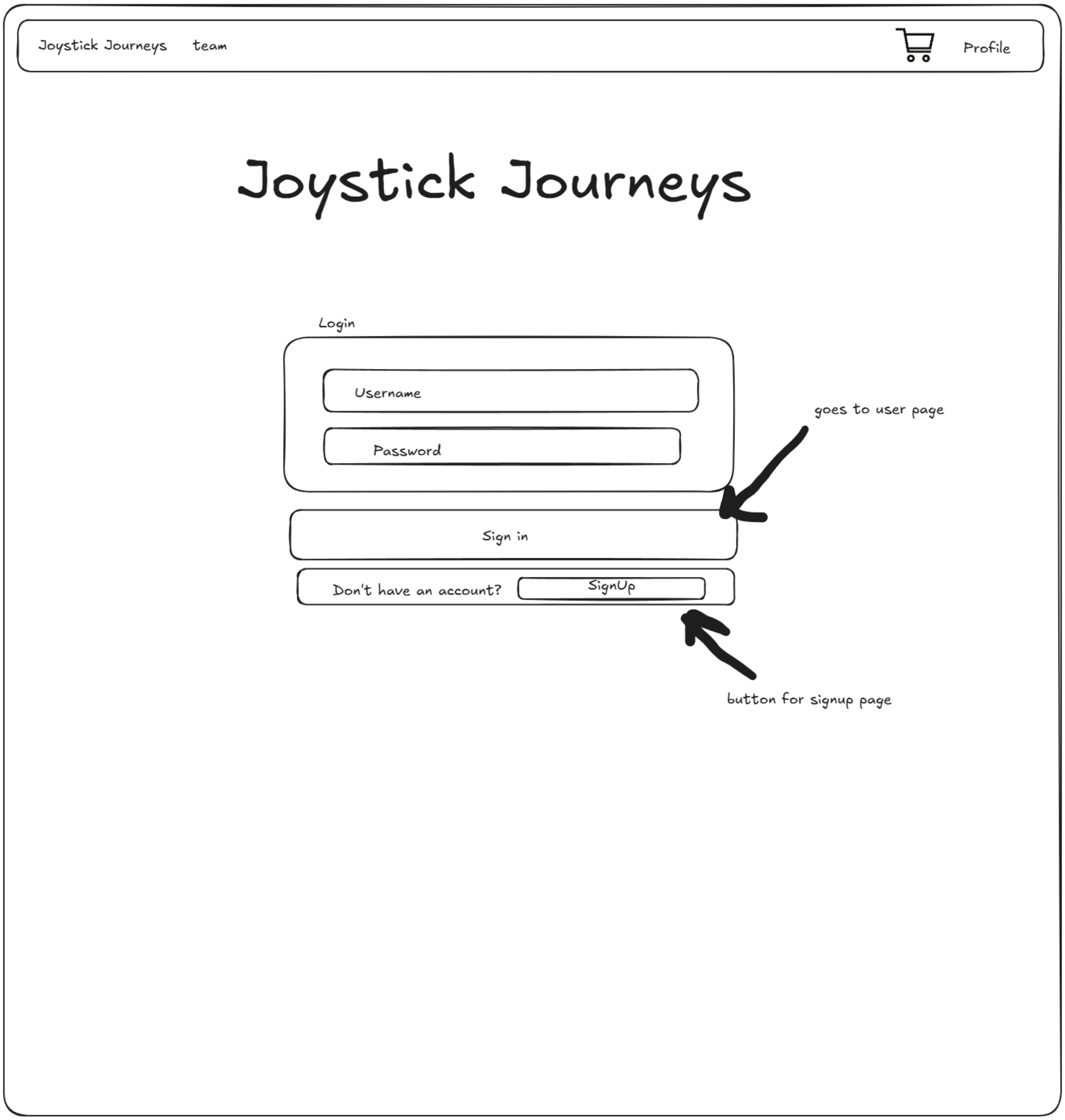
Dataflow:

- Game information is deleted.
- The backend receives the delete request and removes it from the database.

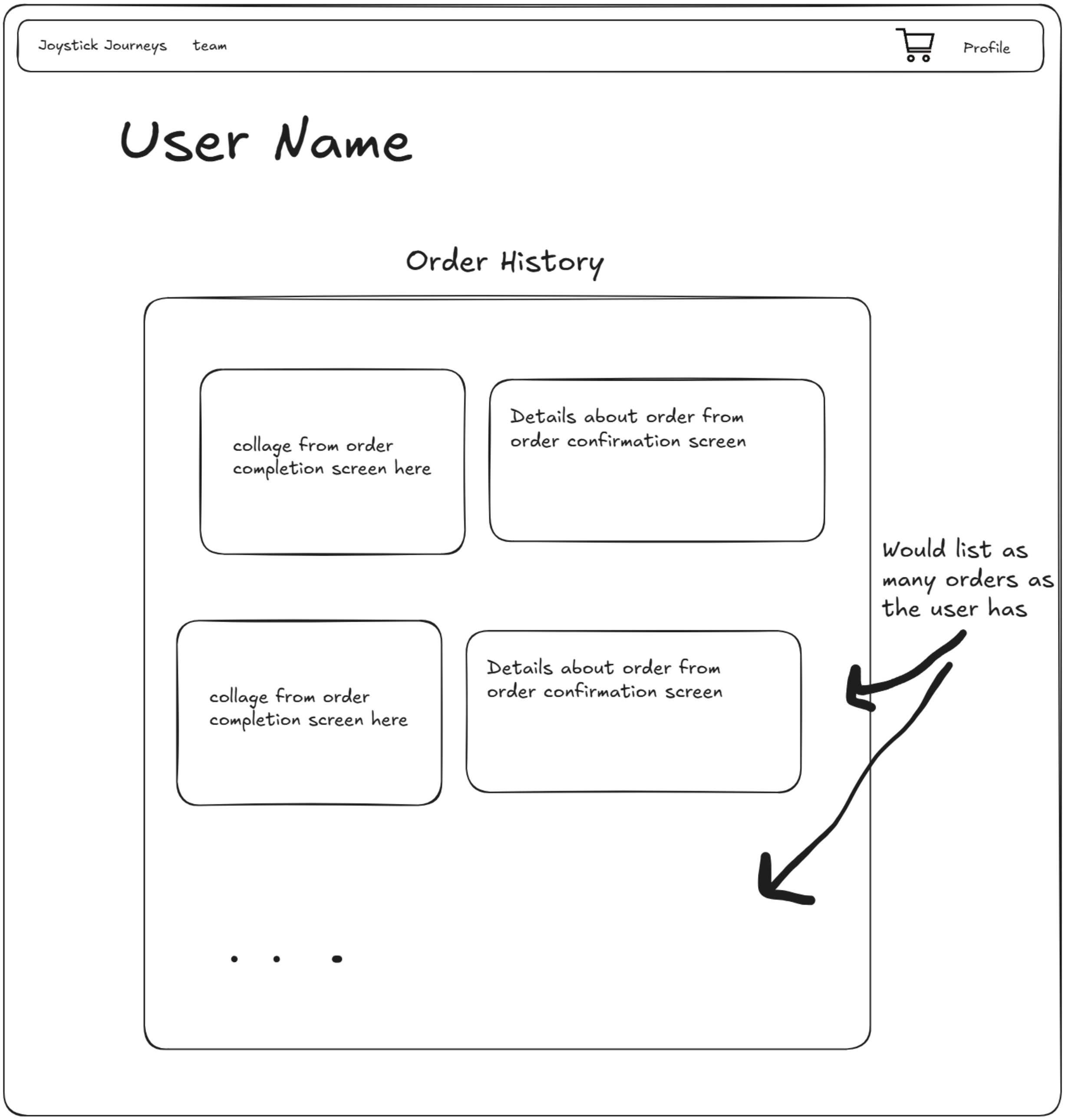
10. User Experience Views


Joystick Journeys
team
Profile
carousel
HELLDIVERS II
scrolls through a list of games
icon that when clicked expands to show filters
selected button looks different
Spotlight
Top-Selling
Classic
Indie
Horror
group of buttons that when clicked will change what games are shown
showcase of games
Game1
Game2
Game3
Game4
Game5
Game6
Game7
Game8
Game9
Game10
About Joystick Journeys ...
Logo

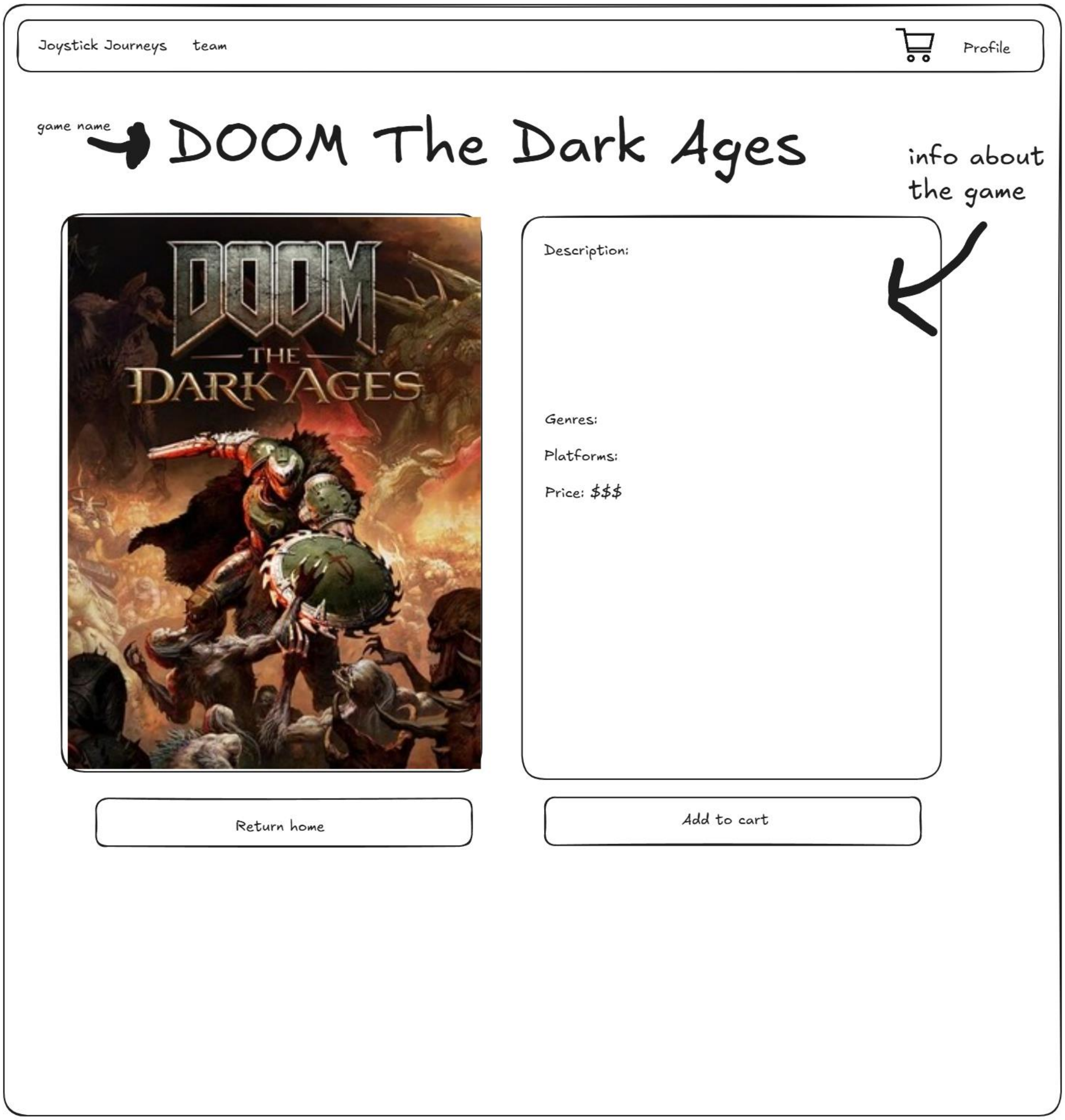
This is the main user page that will be the main display if the games. There is a navbar at the top, carousel, buttons for users to click to see different games, the actual display of the games, and an about section at the bottom.



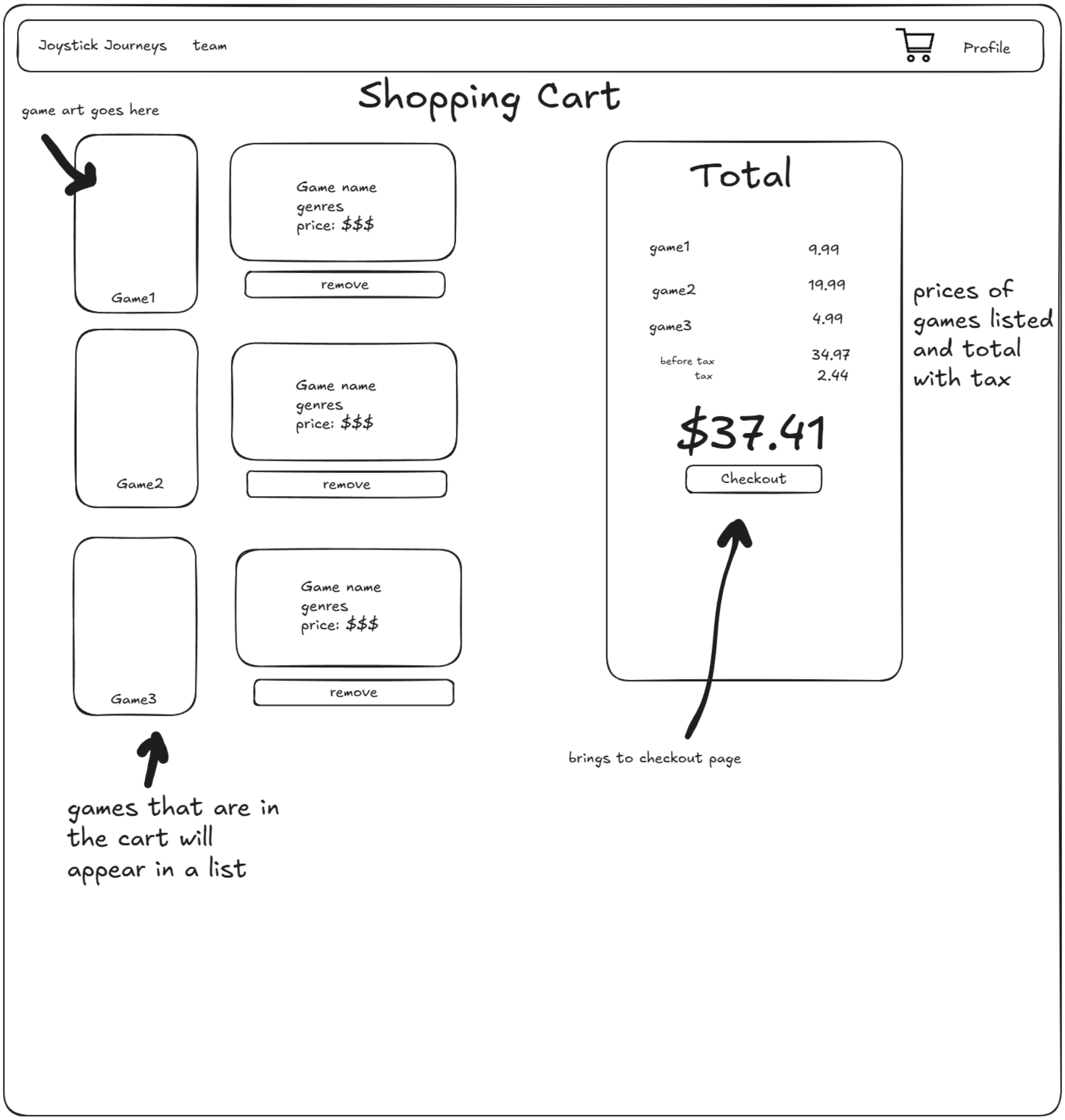
Clicking on profile brings up the login screen. This is where users can login to their account or sign up for a new one.


Joystick Journeys
team
Profile
User Name
Order History
collage from order completion screen here
Details about order from order confirmation screen
collage from order completion screen here
Details about order from order confirmation screen
Would list as many orders as the user has
. . .

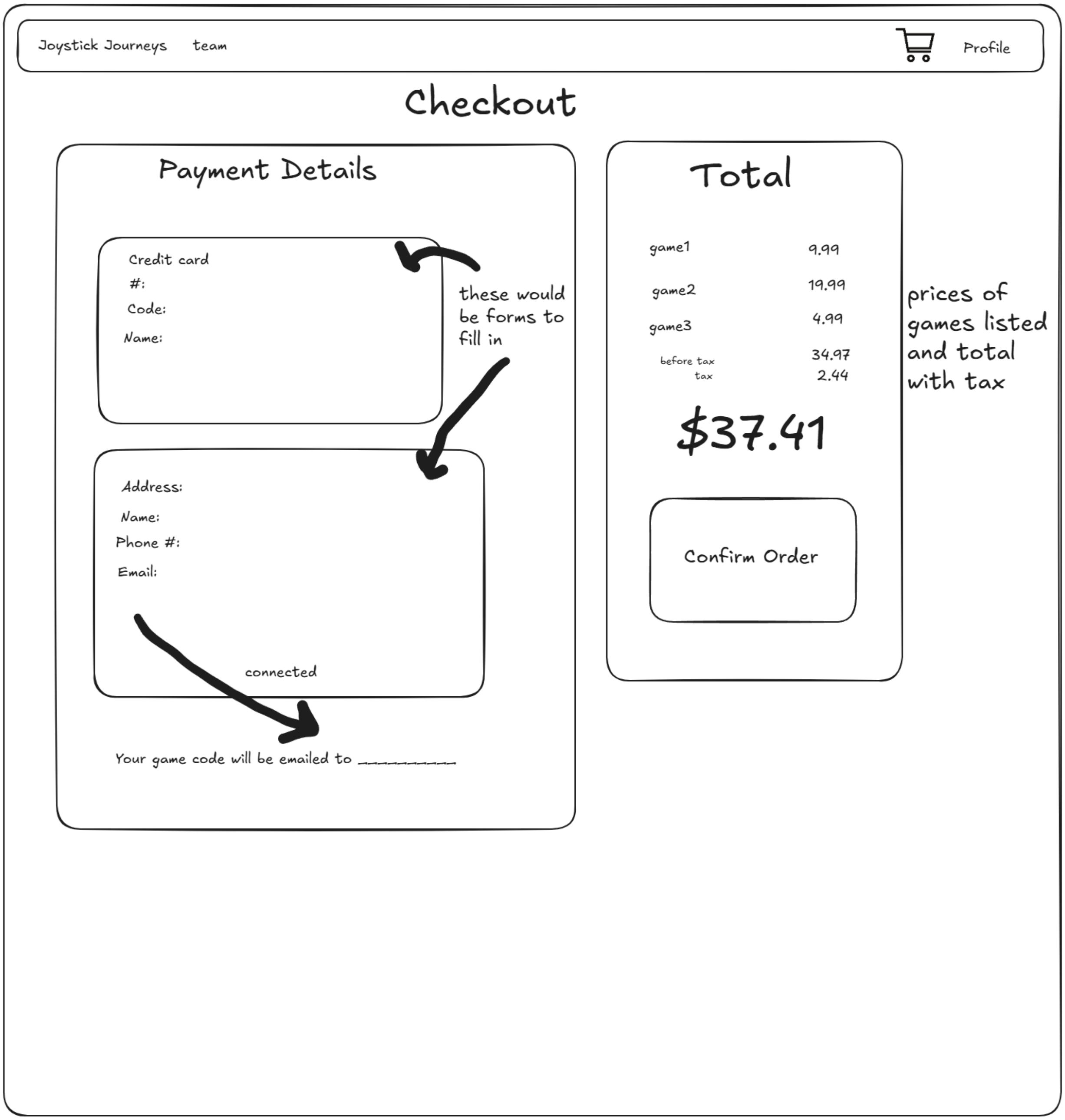
A successful login would bring up the Order History screen. This will showcase the same details and images seen on the Order Completion screen.



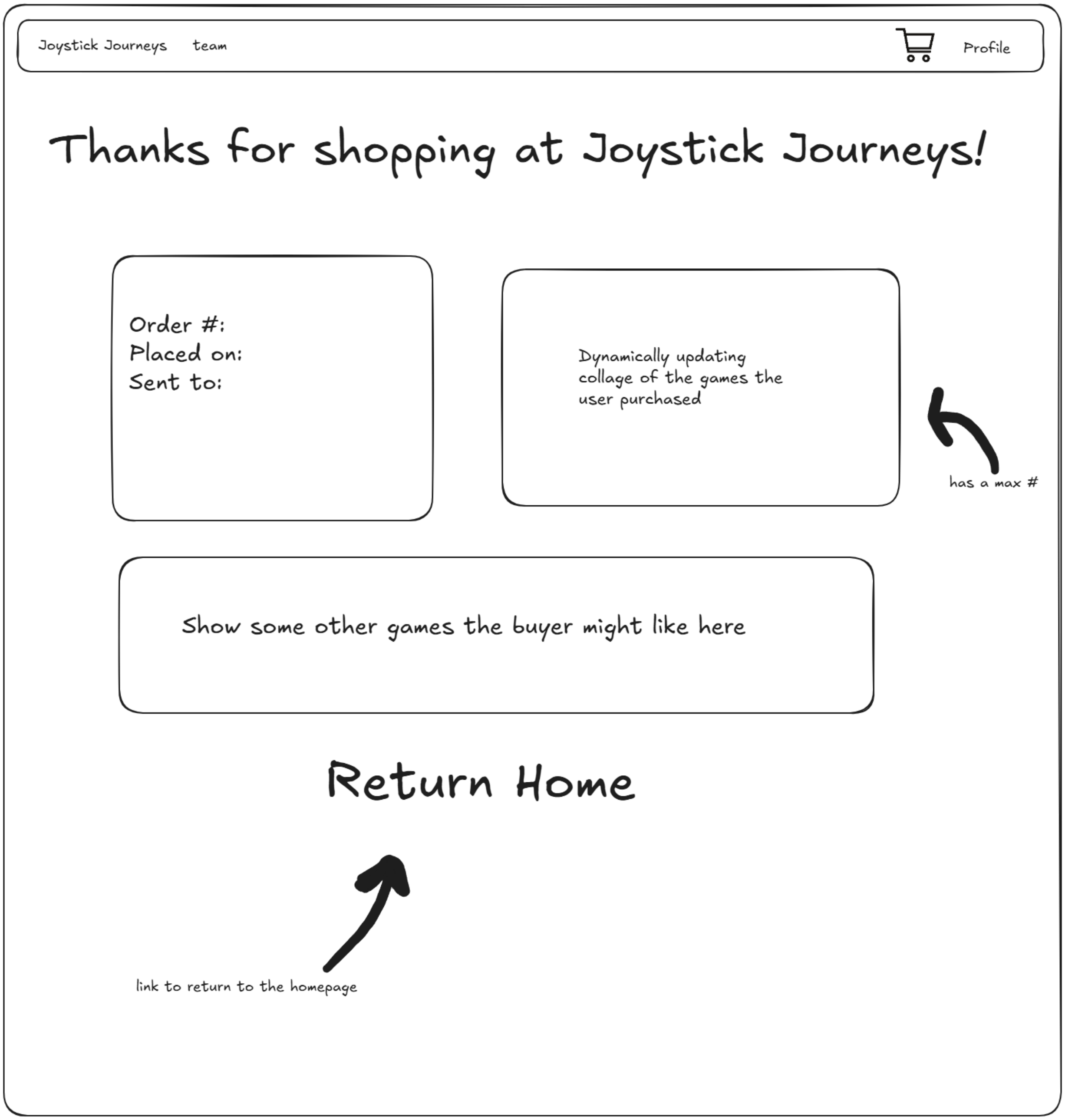
Clicking on a game on the home page brings up the game view for the specific game. This will display more specific info about the game, as well as provide a button to add the game to the cart. The Return home button returns the user to the home screen.



Clicking the shopping cart icon will bring up the shopping cart page. This will list the games in the user's cart on the left side. This is more detailed and also allows removing an item. The right side lists the total and also has a button which brings the user to the checkout page



The checkout page would be very similar to the cart page, but instead of listing a bunch of games there will be forms for the user to input their information. Hitting the confirm order button brings the user to the order confirmation screen.


Joystick Journeys
team
Profile
Thanks for shopping at Joystick Journeys!
Order #:
Placed on:
Sent to:
Dynamically updating collage of the games the user purchased
has a max #
Show some other games the buyer might like here
Return Home
link to return to the homepage

This is what the order confirmation screen would look like. It would show some order details as well as a collage of what games were bought. There would also be a section showing similar games to buy and a link to return home.

11. Final Comments

This is our proposal for the Joystick Journey's website. We believe our project will create not only an easy way for gamers to browse, discover, and purchase video games, but cultivate a community for our users to be a part of and share in their love for video gaming. We want to learn to make a website with more advanced technologies such as React and Node.Js, as well as learn to implement API such as Rawg.io to make the website even better.

Thank you for taking your time to review and consider our proposal. We are open to any feedback and suggestions you may have!

Contact Information:
henrymcm@iastate.edu
cmbeight@iastate.edu