# External Dependencies

This project was built using **Node v13.12.0** and **MongoDB v4.2.5**.

# Setup

npm install needs to be executed in the following directories:

* transact-games
* transact-games/rest-service
* transact-games/front-end

# Running the project

### Importing data

The project uses a **database named TransactGames**. Each json file in the Resources folder should be imported into that database. If you have **mongoimport**, run the following in the project root (transact-games):

mongoimport --uri=mongodb://127.0.0.1:27017/TransactGames --file=Resources\game.json

mongoimport --uri=mongodb://127.0.0.1:27017/TransactGames --file=Resources\user.json

Otherwise, you can use **Compass**. The sample user information is:

Username: Tyler

Password: mypassword

Username: Greg

Password: 12345678

Username: NewUser

Password: password

Note: All the sample user data is fake. If you want to create a new user, you will need a valid credit card number (I used https://www.getcreditcardnumbers.com/).

### Launching the project

To start the project back-end, run node rest-service/app.js from the project root. Note that the application server runs on **port 3000**. To start the front-end, navigate to transact-games/front-end and run npm run serve. For more details about how to use the site, see the included UserGuide.pptx.

# Reflection

### What would I do if given more time?

This was not a requirement for the project: I am including it for my own benefit.

1. Integrate an external API to give users a wider selection of games. This would, of course, require additional pagination logic on the games page and a more optimized approach to fetching information about a game.
2. Improve password security.
3. Fully implement the system that would be used by the warehousing location to process game shipments. My idea right now is that the user would print a QR code and tape that to their package. When the package arrived, the QR code would be scanned to obtain information about the game and the user who sent it. An additional program could be used to automatically update the user’s inventory when the QR code is scanned.
4. Allow users to have more than one a particular game in their inventory at a time. At first, limiting users to one of a particular game at a time seemed like a good idea; however, it ended up making the site more confusing to use overall.
5. Limit the number of games a user can have in their inventory and the amount of time a game can spend in a user’s inventory. If this site were real, there would need to be a way to prevent users from using it as a cheap way to store their games.