

# MANNAN KASLIWAL

(347) 255-0998

New York, NY

[mkasliw1@binghamton.edu](mailto:mkasliw1@binghamton.edu)

[PORTFOLIO](#)

[GITHUB](#)

[LINKEDIN](#)

## TECHNICAL SKILLS

React, Redux, JavaScript, Rails, Ruby, HTML5, CSS3, Git, PostgreSQL, SQL, jQuery, AWS S3, TDD, RSpec, C++, Java, Agile, OOP, Mongoose, MongoDB, D3.js, SVG, Node.js, Express, Python, Webpack, Heroku, SQLite

## EDUCATION

**BS in Computer Science** - Binghamton University, State University of New York | 2014 - 2018

- Magna Cum Laude (GPA: 3.7/4.0), Dean's List (2014 - 2018), Chi Alpha Epsilon National Honor Society
- Relevant Coursework: *Software Engineering, Database Systems, Programming for the Web, Design & Analysis of Algorithms*

**Web Development** - App Academy | 2020

- Immersive 1000-hour full-stack web development bootcamp with <3% acceptance rate
- Teaches full-stack web development topics such as TDD, scalability, security, single-page apps, and web development best practices

## PROJECTS

### Superflix

live | [github](#)

*A superhero themed Netflix clone developed with React, Redux, JavaScript, Ruby on Rails, AWS, and PostgreSQL*

- Engineered a debounced search feature which, upon changes to user input, filters media based on fully matching and partially matching video titles and genres in optimal time
- Implemented custom back and frontend user authentication by combining Rails conventions, Active Record, and customized React-Router components to enable a personalized experience for each user, such as being able to have their own watchlist
- Leveraged AWS S3 for cloud video and image storage to reduce load times for displaying media and to allow for graceful application scalability when new content is added
- Optimized database usage and eliminated N+1 queries by observing PostgreSQL and Active Record best practices

### NBA Stat Race

live | [github](#)

*An NBA stat visualizer to track the route a stat took through a season for chosen players; implemented with JavaScript, D3.js, and SVG*

- Designed asynchronous functions utilizing JavaScript 'fetch' calls to obtain NBA statistical data and NBA players data from the *balldontlie* API based on dynamic user input and parameters
- Charted line-graph representations of the data received from API calls using D3.js to give users a superior way to visualize the players' statistics in comparison to one another and also relative to how far into the season the players are
- Revamped the D3.js graph with stroke-dash animations and D3.js transitions to illustrate the growth and decline of the statistics as the season progressed, thereby producing a more immersive user experience

### UpNext

live | [github](#)

*A mobile-first app to track movie interests and getting recommendations; built using MongoDB, Express, React, Redux, and Node.js*

- Headed a four man team by monitoring the development of the different pieces of the project, delegating frontend and backend tasks as needed, and reviewing & merging git pull requests to keep a stable git repository
- Employed the Mongoose ORM to filter and map results of *TMDb* API calls into documents stored in MongoDB to prevent future redundant API calls and shift reliance to our own backend whenever possible
- Utilized CSS media queries and responsive settings in React components to create a fully responsive, device agnostic design

### Gamerverse

live | [github](#)

*A website that aims to help gamers meet new friends and find gaming events in their area; created using Ruby on Rails and SQLite*

- Leveraged Active Record 'through' associations to optimize backend data extraction from multiple tables in a single query
- Composed RSpec tests to follow a Test Driven Development process, resulting in code that has been thoroughly tested
- Applied the Agile process model and usage of PivotalTracker to simulate a realistic development cycle with multiple developers working in unison