Matthew Francis | 289-772-7465 | Burlington, ON
Portfolio| GitHub | mattfrancis888@gmail.com | LinkedIn

Technical Skills

Frontend: React, Next.js, React-Spring, Redux, Typescript, React- Testing-Library, Enzyme, Jest, HTML, SCSS, BEM, CSS, Bootstrap, JS, Figma, Lighthouse, axe | **Backend:** Authentication, Database Normalization, Docker, CI/CD, Node.js, Express.js, PostgreSQL, MongoDB, GraphQL. | **Tools:** Git, AWS, Gitlab, Azure DevOps, TravisCI, GitHub Actions, Shopify, Microsoft Teams, Slack, Heroku, Vercel .

Experience

Non-Incorporated Company (My Startup) | Founder & Full-Stack Software Developer — Aug 2021 – Present

- Collaborated with partner for ideas, design and engineering to create a platform for people to collaborate. Built with GitHub Actions, Next.js, React-Spring, Typescript, HTML, SCSS, BEM. CSS, JS, Express.js, PostgreSQL, Figma, Lighthouse, axe.
- Collaborated with mentors in accelerator to challenge decisions, address market research and pivot on ideas.
- Collaborated with Ontario's Co-operative Association to comprehend and apply co-operative principles.

Habanero | Frontend Software Developer — Sep 2021 – Jan 2022

- Collaborated with design, operations and engineering team to develop a web application for over a dozen of clients with Azure DevOps, React, React-Spring, Typescript, HTML, SCSS, BEM, CSS, JS, Figma, Lighthouse, axe.
- Involved in agile methodology that included, biweekly scrum, backlog refinement, check-ins, QA, and retrospective.
- Used Git and Azure DevOps to conduct code reviews, raised pull requests, pushed code, and operate a CI/CD pipeline,

Badal | Full-Stack Software Developer — May 2021 – Sep 2021

- Collaborated with design, operations and engineering team to develop a Shopify application that automates sales for Shopify stores with over hundreds of products and users with GitLab, React, Redux, Typescript, PostgreSQL, GraphQL, Express.js, Docker, HTML, SCSS, BEM, CSS, JS., Lighthouse, axe.
- Involved in agile methodology that included biweekly sprints, backlog refinement and daily stand-ups.
- Used Git and Gitlab to conduct code reviews, raised pull requests, pushed code, and operate a CI/CD pipeline,

Laurier's Computer Science Club| Lead Software Developer — Sep 2020 – May 2021

- Led 15 team members in designing and engineering a web application for the club with React, HTML, SCSS, CSS, JS, Figma.
- Held weekly meetings to explain technology stack and version control with Git; used agile methodology that included biweekly sprints and backlog refinement; used Git and GitHub Actions to conduct code reviews, raised pull requests, pushed code, and operate a CI/CD pipeline.

Education

Honours Bachelor of Science, Computer Science | Wilfrid Laurier University, Waterloo — Sep 2018 - Present

Projects

Blizzard (Link)

- Replicated Blizzard's website, known for publishing games like Call of Duty and World of Warcraft. Users can view pages for each game. Used React Spring to create physics-based animations like accordion, image parallax, and card flip.
- Built with React, React Spring, Redux, Typescript; HTML, SCSS, CSS, JS, tested with React Testing Library.

Steam (Link)

- Replicated Steam's website, the largest video game retailer for PC games. Users can browse for games and discounted games; add games to their cart; create, edit, delete reviews for games; authenticate; upload profile picture and edit their profile.
- Data is normalized to BCNF and is stored in PostgreSQL. Authentication is done with cookies, JWTs and refresh tokens. Used security practices against XSS / CSRF attacks.
- Built with React, Redux, Typescript, HTML, SCSS, CSS, JS, Express.JS, PostgreSQL, tested with React Testing Library.

Kijiji (Link)

- Replicated Kijiji's website. Users can create, edit, and delete their listing; upload a display picture to their listings, search listings based on category and location; view personally created listings; authenticate.
- Data is normalized to BCNF and is stored in PostgreSQL. Authentication is done with cookies, JWTs and refresh tokens. Used security practices against XSS / CSRF attacks.
- Built with React, Redux, Typescript, HTML, SCSS, CSS, JS, Express.JS, PostgreSQL, tested with React Testing Library.