(Joe) Xiaohong Chen

LinkedIn: https://www.linkedin.com/in/xiaohong-joe-chen-704b9b148/ | Mobile: (678) 557-8140 | Email: xchen671@gatech.edu

EDUCATION

Georgia Institute of Technology, Atlanta, GA **Jan.2018 – Dec.2020**

- Bachelor of Science in Computer Science

Georgia State University, Atlanta, GA **Aug.2015 – Dec.2017**

- Bachelor of Science in Computer Engineering

TECHNICAL SKILLS

- **Programming Language**: Python, JavaScript, Java, GraphQL, CSS, HTML, Sass
- **Frameworks/Library:** React/Hooks, React Native, AWS Amplify, Redux, Node.js, Express.js
- **Databases/Package /Tools:** Git, AWS AppSync, npm, Postman, Amazon DynamoDB, MongoDB, Firebase, NumPy, SendGrid, mailtrap

WORK EXPERIENCE

Software Engineer Summer Intern | QianLi Travelling **May.2019 – July.2019**

- Developed full stack travel booking website application based on **RESTful** API with **Express.js** and **MongoDB** as database in **Agile** software development.
- Built **responsive** user interface booking section and tours section using **SASS Mixins** to write media queries.
- Implemented the authorization permission to delete the tour posts only for administrators.
- Used **nodemailer** sends password reset token via email to let user reset the password and used **mailtrap** to safety test. Updated the user new information into MongoDB.
- Implemented email template with **Pug** and used **SendGrid service** to send welcome emails to the clients. Used **ndb** to debug **Node.js**.

PROJECTS

AWS Cloud Based Free-Food React Native Application **Feb.2020 – April.2020**

- Implemented **Responsive** & **Adaptive** React Native APP adjusting to different sizes & platforms that allows college students and college event holders to post and search where /when to get free food along with their event detail based on **GraphQL API** with **AWS Amplify** Framework.
- Used **Amazon DynamoDB** database to store free-food posts, user comments and user adding like for the specific post.
- Implemented the frontend user interface with **React/hooks**, built an interactive product with **materialize-design** and **font-awesome**. Used **Amazon Cognito** to handle user authentication.
- Implemented **Native Device Features** about displaying nearby free-food posts on an interactive Google Map navigating user to the event.
- Used a mix of local storage and **Redux** to manage and persist states.

MERN Based Full-stack Contacts Keeper Web Application **Sep.2019 – Dec.2019**

- Developed full stack contact manager application allows authorized user to create, read, update, delete and filter their own contacts based on **RESTful** API with **Express.js** and **MongoDB** as database. Used **Axios** to consuming API.
- Used **mongoose** as mongo driver, provided schema-based solution to model application data, managed relationships between contacts and users.
- Implemented the frontend user interface with **React/hooks**, built beautiful, usable products with **materialize-design** and **font-awesome** for **Ajax** based dynamic web pages.
- Used **JWT** to securely implement user authentication symmetrically signed by a shared secret using the HMAC algorithm.
- Used **bcryptjs** to hash plain password with salt so that protected against rainbow table attacks, Used **Express-validator** as middleware to validate user input on the login and register page.
- Used local storage and React **Context** to manage application state, meanwhile to keep users in logging status. Deployed the application on **Heroku**.

Factor Graph Based Estimate Vehicle Poses | Python, NumPy, Google Colab **March.2020 – May.2020**

- Implemented the **Iterative Closest Points (ICP) algorithm** to estimate transform between two dense sets of points and used it alongside **GTSAM** to perform simultaneous localization on Lidar scans.
- Dataset was composed of 180 Lidar scans captured by Argo AI from car's front camera, which making left turn at an offset T-intersection. Used **tracking landmarks** to compose the initial guess which of two ICP input clouds are the start state and the captured 0.1 seconds after start state.
- Applied **gtsam.Pose3** transform on each point in the cloud. Converted each point in the point cloud into homogeneous points and applied a homogeneous transformation on each point with **NumPy**. Assigned closest points pairs by **kd_tree algorithm** of **sklearn's NearestNeighbors**.
- Implemented a **factor graph** to estimate the pose of vehicle in world coordinates with ICP transforms are the factors and used GTSAM to construct the factor graph.