KAIJUN ZHU

kaijun.zhu0629@gmail.com | github.com/zhukaijun0629 | LinkedIn.com/in/kaijun-kevin-zhu | +1-6073795124

EDUCATION

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN GPA 3.75/4.00
Master of Science, Civil Engineering May 2016

SOUTHWEST JIAOTONG UNIVERSITY, CHENGDU, CHINA (HONOR STUDENT) GPA 3.78/4.00
Bachelor of Science, Civil Engineering June 2014

SKILLS

Programming/tools: • python • TypeScript • Java • Matlab • React • NodeJS • SQL • MongoDB • Git
Technical Skills: • Algorithms • Data Structures • Machine-Learning • Object-Oriented Programming

PROJECTS

KAIPLACE (kaiplace.web.app) – A Photo Sharing Social Networking Service (Web/IOS/Android)

Front-end (deployed on Firebase Hosting): React, CSS **Back-end** (deployed on Heroku): NodeJS, Express
Database: MongoDB, Mongoose **Image Storage**: Amazon S3, AWS-SDK

- Implemented **React SPA** front-end and **NodeJS/Express** back-end with **RESTful** API to provide smooth social networking service for customers.
- Designed **schema** models for the business logic of Customer-Oriented Networking system to support all the functionalities and connected to **MongoDB** database using **Mongoose**.
- Managed user **authentication** and **authorization**: Encoded user password and compared **hashed** password using **bcryptjs**; Authorized user with **JWT** stored in browser local storage which would expire in certain period.
- Facilitated photo uploading experience by auto-filling required fields, such as date taken and location, by extracting EXIF metadata and then fetching **Google Reverse Geocoding API** and **Bing Time Zone API**. On server-side, utilized the extracted data to further organize the photos based on time and location.
- Reduced image size using **imagemin** and stored images to **Amazon S3** with the help of **AWS-SDK.**

PROFESSIONAL EXPERIENCE

WALTER P MOORE, WASHINGTON DC – A $114M/year multinational engineering consulting firm
Structural / Software Engineer Feb. 2018 – Jan. 2021

KEY PROJECT: STEEL CONNECTION DESIGN AUTOMATION (Major programming language: **python**)

- In charge of developing a comprehensive digital workflow for steel connection design, which consists of 4 phases: A) Data Extraction, B) Analysis and Bucketing, C) Design Automation, and D) Result Visualization, reducing connection design project timeline by **40%** on average (varies from weeks to months per project scale)

A)
- Developed add-ons for Autodesk Revit – a building information modelling (BIM) software – using python, **pyRevit** and **Revit API** to read/write user-specified data of structural elements via .csv files, enabling data exchange between applications.
- Expanded the add-ons to perform **version comparison** on multiple .csv files to indicate created, updated, and deleted structural elements, saving **20%** of manual-comparing efforts.

B)
- Designed **classification** rules based on global and local 3-d geometric relationships among structural elements to analyze the raw .csv data, in order to identify joints and finally bucket the qualified joints into 16 different types of steel connections.
- Improved classification algorithm for large-scaled 3-d models with **k-d tree** data structure and **nearest neighbor searching** to reduce runtime by **70%**.

C)
- Implemented service automating the connection design process by
  - ▪ **auto selecting** the most eligible stored design template based on section sizes and loading data,
  - ▪ **optimizing** design parameters according to the **dynamical** configuration setup per design criteria and customer requirement, and
  - ▪ sending requests to engineer if no existing design would work.
- Stage C reduced **90%** of manual design efforts, about **80 hours** per project (mid-sized)

D)
- Combined and pushed design results to Tekla BIM model via .csv file and visualized detailed connections in 3-d simulation, demonstrating all plates and bolts using **Tekla API.**