

HUNG, WEI-CHENG

Personal Site: <https://weichenghung.com> (619) 366-1344 s777610@gmail.com
<https://www.linkedin.com/in/weichenghung> <https://github.com/s777610>

COMPUTER SKILLS

Programming Language

- Python(Excellent), Javascript(Advanced), SQL(Advanced), CSS(Advanced), HTML, Shell Script, Java, PHP, R
- Application, Library & Framework
- React, Vue, JQuery, SASS, Git, Flask, MySQL, SQLite, PostgreSQL, MongoDB, Heroku, AWS, MapReduce

EDUCATION

California Lutheran University, Thousand Oaks, CA

Feb. 2019

Master of Science in Information Technology (current GPA: 3.8/4.0)

Chung Yuan Christian University, Taoyuan, Taiwan

Sep. 2014

Bachelor of Mechanical Engineering

WORK EXPERIENCE

Software Engineer Intern - Tap Series, Thousand Oaks

Sep. 2018 — Current

- In charge of scraping financial data, visualizing data, and generating reports with python. Working on testing MySQL and SQLite database of online course platform. Testing the online course app with React.js.

Full Stack Developer - Locomotives, Changhua, Taiwan

Nov. 2016 — May. 2017

- In charge of SQL database and backend development with python. Build front-end with HTML5, SCSS, JS. Work on customer data visualization with python.

MAIN PROJECT EXPERIENCE

Stock Market Web App Personal Blog (Full Stack Development and Data Visualization: <https://weichenghung.com/>)

This Web-App shows some of my investment. I used Flask as framework to build the web and deployed it on AWS. It retrieves data from "IEX" API and scrapes data from some financial site by BeautifulSoup4. It analyzes data and makes interactive charts on the site by using Pandas and Plotly. It allows users to search any security in the US market. All data are stored in SQLite. I use this App very often to check my portfolio. (Python, SQL, Javascript, SCSS, HTML5, AWS).

Social Media Blog (Full Stack Development https://github.com/s777610/Social_Media_Blog)

This App, which was built by flask, allows users to register and login first. Then, users are able to create the blog post, update the profile picture, edit and delete the blog post. All data of users are stored in SQLite database for testing only. This app takes an advantage of flask-sqlalchemy to create database object. It uses flask_migrate to upgrade the database in case of needs. (Python, Flask, SQL, HTML5, CSS).

Half Half Coin System (Blockchain System Development https://github.com/s777610/blockchain_currency)

It allows users to create wallets first. The key pair(public key and private key) is generated when creating the wallet. Then, users can start mining the new block in order to get rewards. The level of mining difficulty is adjustable. Then, users can send cryptocurrency to other users by assigning a public key of the recipient. All transactions data will be stored in open transactions temporarily. After mining a new block, the data of open transactions is stored in the new block, and open transactions will be an empty list. All node can make HTTP requests to their peer nodes in order to share their data. (Python, HTML5, CSS, Vue).

RESTful API with SQL (Database & Back End Development https://github.com/s777610/Flask_RESTful_API_with_SQL)

This API is deployed on Heroku, provide GET, POST, PUT, DELETE requests. The data can be stored in SQLite locally or PostgreSQL remotely. This API allows users to store some online store data in the database. It can generate access-token and refresh-token for users. The point of this API is implementing authentication. (Python, Flask, Flask_RESTful, Flask-JWT-extended, Flask-SQLAlchemy).

Price Alerts for Online Store (Full Stack Development: <https://pricing-alerts.herokuapp.com/>)

It is able to track the price of items from the online store. It can notify users when the prices of items drop. The application checks the prices every 10 minutes. All data is stored in MongoDB. All passwords of users are encrypted. Utilizing RE to find the specific store for users. The price of item could be found by using BeautifulSoup to parse html. Only admin can add the new store. (Python, MongoDB, CSS, HTML, Mailgun, pbkdf2).