William Wendt

Long Beach, CA 90703
562-316-9759 | williamwendt31@outlook.com
Github: github.com/williamwendt31
Portfolio: william-wendt.tech
LinkedIn: linkedin.com/in/william-wendt-/

EDUCATION

California State University, East Bay **Aug 2018**
B.S. Computer Science
Notable Courses:

- Programming Language Concepts | Analysis of Algorithms | Web Site Development
- Database Architecture | Software Engineering | OOP & Design

Coding Dojo (Certificate) **Dec 2018**

- An immersive 14-week program, completing over +1200 hours of hands-on coding in Java, Python and MEAN/MERN stacks. *(Earned perfect scores in all three stacks.)*

SUMMARY

- Recently obtained a bachelor's degree in Computer Science combined with a certificate in full stack development from Coding Dojo. Equipped and eager to tackle tough challenges to continue evolving as a developer building innovative web applications with increasing proficiency in *Python*, *Java*, *MEAN* and *MERN* stacks with the ability to develop from scratch.
- Completed 5 projects creating websites in e-commerce, social media, chatroom, and Single-Page Web Application Design using *MVC/ MVT* architecture and implementation of appropriate technology stacks for responsive design and scalability in an *AWS* cloud environment.

TECHNICAL SKILLS

Programming Languages: JavaScript, Python, Java, C++, TypeScript, HTML5, CSS3
Frameworks: Django, Flask, Spring Boot, Angular 6+, Materialize, Bootstrap, Express.js
Servers: Apache Tomcat, Node.js, NGINX, Gunicorn, PM2
Dev Tools: VS Code, Spring Tool Suite, NetBeans, Linux Terminal, Postman, MySQL Workbench, DB Browser for SQLite, Git/ Github, AWS EC2, AWS S3, AWS IAM, AWS Route53
Methodologies: OOP, MVC/ MVT, Responsive Design, WebSockets, AJAX, RESTful API
Databases: MySQL, MongoDB, SQLite
Libraries: Socket.io, React.js, jQuery, Mongoose, JSTL

PROJECT EXPERIENCE

Coding Dojo – Burbank, CA **Aug 2018 – Present**
Jr. Full Stack Developer

Project: Chalk It Up
A chat app that allows users to communicate with each other through chat rooms. Users can send messages, change their username and create/ destroy chat rooms.

- Developed an opensource, single-page web application from scratch following the MVC pattern with a responsive UI design using *HTML5, CSS3, Bootstrap, jQuery, and Angular 7.*

- The server and client communicate and exchange JSON data through *WebSockets (Socket.io)* for instant messaging and real-time updates.
- Stored and manipulated data from chat rooms through Mongoose/ MongoDB due to its data representation (JSON), fast access and scalability.
- Deployed application with *AWS EC2/ Route53, PM2 and NGINX.*

Technologies Used: Node.js, Express.js, Angular 7, Bootstrap, jQuery, Socket.io, Mongoose, MongoDB, AWS EC2/ Route53, PM2, NGINX, Git

Project: PetNation

A pet adoption site allowing users to find all pets, domestic or exotic, a home. Users have the power to put pets up for adoption, offering all information to potential owners or to adopt any pets they see on the website.

- Built an opensource, single-page web application following the *MVC* pattern to divide the application into three interconnected parts with responsive UI development utilizing *HTML5*, *CSS3*, *Bootstrap* and *React*.
- Connection between the server and client established through *WebSockets (Socket.io)* for full duplex communication and lower latency interaction, which allows for real-time updates.
- Stored and manipulated user data through *Mongoose*/*MongoDB* due to its high performance and data representation *(JSON).* Deployed the application with *AWS EC2/ Route53, PM2 and NGINX*. Connected to *AWS S3* to save/ load assets.

Technologies Used: Node.js, Express.js, React.js, Bootstrap, Socket.io, Mongoose, MongoDB, AWS EC2/ S3/ Route53, PM2, NGINX, Git

Project: 229th

An eCommerce site where consumers can browse the site looking for their favorite products ranging from clothes, to footwear and accessories. It utilizes a shopping cart in order to keep track of consumers' desired products integrating with Stripe API to complete their transactions. Also, it incorporates an admin UI back-end where authorized users can view/ edit orders and products.

- Designed and implemented an opensource site with *MVC* architecture with responsive front-end development using *HTML5*, *CSS3* and j*Query*. Worked with jQuery to handle user interactions for product search sorting, while using Materialize for object animation. Used *AJAX* to send requests to the server to query the *SQLite* database.
- Used *JSON* data to make requests to the Stripe API. Utilized *Django-ORM* to create, read, update and delete user-data in *SQLite*. Configured and set up a virtualized environment with *AWS EC2/ Route53, Gunicorn and NGINX.*

Technologies Used: Python, Django, SQLite, Stripe API, jQuery, AJAX, Materialize, AWS EC2/ Route53, Gunicorn, NGINX, Git

Project: Roots

A social-media website allowing users to upload and share "user- generated content." Users can interact with their uploads by either commenting, liking, or disliking while allowing interaction with other users by following them or sending them direct messages. Roots allows for creative posts and creative profiles.

- Engineered an opensource website with *MVT* architecture using responsive UI development with *JSTL*, *HTML, CSS3 and Materialize*. Implemented a search sorting feature for user generated content and realtime data loading with *AJAX*. Used *JQuery* to handle user interactions and *Materialize* for object animation.

- Programmed the logic and data layers using *Java*, *Spring framework*, *JPA*, and *MySQL*. Implemented *Spring Boot Security* for authentication and authorization. Deployed the application with *AWS EC2/ Route53 and Apache*. Connected to *AWS S3* to save/ load assets.

Technologies Used: Java, Spring Boot, Spring Boot Security, MySQL, AWS EC2/ S3/ Route53, Apache, jQuery, AJAX, Materialize, Git

Project: Trip Buddy

A web app that keeps track of all users' travel plans. Everyone's travel plans are public and anyone can join in. Users have full control over their own planned trips.

- Developed an opensource web app following the MVC design pattern with a responsive front-end utilizing HTML5, CSS3 and Bootstrap.
- Used Django-ORM to create, read, update, and delete user-data in SQLite. Authorized and authenticated users using bcrypt library. Configured and set up a virtualized environment with *AWS EC2/ Route53, Gunicorn and NGINX.*

Technologies Used: Python, Django, SQLite, Bootstrap, AWS EC2/ Route53, Gunicorn, NGINX, Git

California State University, East Bay – Hayward, CA **May 2018**

Student Programmer

Project: Epsilon (4 - Person Team)

Epsilon is a '9GAG' clone website that was developed in an Agile environment. All members contributed to the detailed software requirements specification and software design document. Implemented functionalities such as: login/sign-in, user uploads, and all user profile functionality (CRUD). Also designed the database schema.

- Responsible for user authentication/authorization with b*crypt* for login and registration. Created user upload features using *JavaScript*. Designed and implemented a robust database using *SQL*.

Technologies Used: Node.js, Express.js, MySQL, Handlebars.js, Agile Development Lifecycle, Git - Version Control