Christopher Wendland

Newport Beach, CA | 949.910.4342 | chris.r.wendland@gmail.com
github.com/cwendland | chriswendland.com

Skills

Javascript, React, Node.js, Bootstrap, JQuery, AJAX, Angular, Express, Java, C++, Python, Heroku, MongoDB, SQL, Assembly, HTML, CSS, RESTful API, GraphQL, MySQL, Agile Development, SCRUM, MERN Stack, XSLT, VBScript, VSCode, Eclipse, XCode, MATLAB , Git, POSTGRES

Experience

UCLA / Peer Learning Facilitator
October 2020 -June 2021, Los Angeles

Held tutoring sessions for fellow undergraduate students. Taught Python for modeling dynamics of multivariable nonlinear systems. Instructed up to 6 students at a time.

Transcepta / Software Engineering Intern
May 2017 - October 2017, Aliso Viejo
Programmed specialized logic into the document parsing system for customers. Redrafted training SOPs for new hires after witnessing inefficient onboarding. Utilized VBScript, XSLT, MS SQL, and Stored Procedures to complete assigned projects.

Education

University of California, Los Angeles / Molecular, Cellular, and Developmental Biology B.S
September 2018 - June 2022, Los Angeles

University of California, Los Angeles / Software Engineering Program / Full-stack Web Development
Full time, immersive training program with an emphasis on Fullstack Web Development
August 2022 - November 2022, Los Angeles

Course Highlights

Data Structures and Algorithms, Discrete Mathematics and Probability Theory, Memory Management and Performance Enhancement, Computer Architecture, Software Design Patterns, Computer System Architecture

Projects

Wordalle/ React, CSS, Bootstrap, POSTGRES

Wordle-esque game to guess what prompt was used in generating Dall-e AI images. Database built using POSTGRES. Deployed using Firebase.

Build-A-Bet/ React, CSS, MongoDB, GraphQL

Sports betting website to create a parlay with RESTful API, built using MERN stack. BCrypt used for secure authorization as well as sessions and cookies. Deployed with Heroku.

Harmonic Analyzer/ Python, Arduino

Fourier transform performed on raw audio data from the Arduino to get the frequency spectrum. Python then used to classify the frequency spectrum into individual notes, allowing tuning of multiple types of instruments.