Samuel Hale

samuelhalebusiness@gmail.com | samuelhalebusiness.wixsite.com/portfolio/projects | github.com/SamuelHaleDev

Education

University of Michigan - Dearborn – BS in Computer and Information Science 2024

Skills

Languages: JavaScript, HTML, CSS, C#, Java, SQL
Frameworks: .NET, React, Node, Flask, JSwing, JUnit, PyTorch
Tools: AWS, Docker, Git, Jira, MongoDB, PostgreSQL, SQLite, Workato, Digital Ocean

Experience

Software Engineer Intern, WoodWing – Detroit, MI 2023 – 2024

- Authored technical specifications for the static summary feature, ensuring alignment with constraints by converting user stories into detailed technical documentation following AGILE methodologies
- Implemented a new static summary feature that builds reference links from static summary titles to the corresponding headers in the chapters using JavaScript, Workato and Assets API, saving an estimated 100+ hours of manual editing
- Reduced manual HTML validation time by 100% via integrating a JavaScript that uses W3C API to validate all HTML pages
- Ensured the delivery of 40 books to the Brazilian government by providing robust user acceptance testing through communication on ZenDesk and Jira tickets and testing locally via AWS Lambda JSON testing configuration
- Enhanced the security of customer production environments by incorporating AWS Secrets Manager in automations, reducing the risk of unauthorized access or data breaches
- Developed an archival solution that triggers off of issue status in Studio and processes article metadata to generate XML metadata files, reducing manual archival time by 100% and streamlining issue release processes

Projects

PlayPal AI github.com/ww-samuel-hale/playpal

- Created an aesthetically pleasing Game Card, by coding a React rendered HTML skeleton of the component and styling with CSS, facilitating quick browsing of game recommendations for users
- Devised a content recommendation algorithm, that utilizes SKLearn cosine similarity to create user and game vector profiles and perform element-wise comparisons, resulting in game suggestions customized to user preferences
- Cut recommendation algorithm speed by 80%, by implementing a capped First in, First out (FIFO) data structure that stores recommendations and past interactions to prevent re-recommendations

StorAI storai.net

- Integrated course metric tracking for over 20 users, by sending REST HTTP requests with course progress data, storing it in a PostgreSQL database, and developing a user dashboard in React to display this data in a loading bar and completion cards
- Programmed a normalized PostgreSQL database by modeling Entity-Relationship diagrams, merging multi-value attributes into distinct related tables, and writing SQL execution scripts, resulting in a scalable backend architecture
- Built a REST API in Node.js utilizing Express.js to stand up the server and all CRUD endpoints and passport for secure endpoint authentication, supporting efficient data flow between server-client and managing 100s of API calls each day
- Slashed over 20 hours off of development time by configuring continuous integration with a GitHub Actions script that SSHd into our Digital Ocean linux droplet and copied project files over, which eliminated manual deployment

WhatsChat github.com/SamuelHaleDev/Chat-Application

- Engineered a push design pattern with WebSockets and UML that allows users to communicate in less than 1 second
- Designed the GUI chat application's layout and used Java Swing to implement it, making the user experience flawless
- Automated testing by creating unit test cases in JUnit that cover class methods, achieving over 90% coverage across tests

Loan Eligibility Predictor github.com/SamuelHaleDev/LoanPredictionModel

- Estimated loan eligibility with 90% accuracy by training a PyTorch standard neural network on 613 rows and 11 features
- Increased model prediction accuracy by 15% via normalizing pre training data to fall within the range of -1 to 1
- Addressed NAN values by imputing missing data with average value, balancing quality and quantity of the data set