Zoie Bonnette

📍 Lubbock, TX   ✉ bonnettezoie@gmail.com   📞 (949) 449-3192   🔗 zoiebonnette03.github.io

in zoie-bonnette   ○ zoiebonnette03

Summary

Bachelor of Science in Computer Science candidate, and experience developing front-end web components using Angular, collaborating with software teams, and testing API controllers in C#. Passionate about using software development as a problem-solving tool to drive improvements within the public sector.

Education

Texas Tech University   *Sept 2021 – May 2025*
BS in Computer Science, minor in Mathematics

- GPA: 3.9/4.0 (a link to somewhere ↗)
- **Coursework:** Computer Architecture, Comparison of Learning Algorithms, Computational Theory

Experience

Software Engineer   *Lubbock, TX*
Tyler Technologies   *September 2024 – Present*

- Developed an API controller and created endpoints to handle data transactions between the client and server
- Designed and implemented front-end components, integrating API endpoints, enabling data entry and updates to the SQL database
- Enhanced data presentation and functionality by redesigning tables with AG Grid, improving sorting capabilities and user experience

Software Engineer Intern   *Lubbock, TX*
Tyler Technologies   *June 2024 – Aug 2024*

- Implemented a mobile-friendly "next" button with enable/ disable functionality for a web application using Angular and BroswerStack for cross-browser testing
- Developed and executed unit tests in C# using Moq library, xUnit framework, and Visual Studios debugging tools to ensure code quality and functionality
- Built a real-time "user is typing..." indicator with animated ellipses using event listeners in Angular
- Participated in recurring Scrum meetings (stand-up, sprint planning, sprint reviews, retrospectives), gaining hands-on experience in collaborative software development practices and team workflow

Projects

Multithreading Matrix Processing   *github.com/Multithread ↗*

- Processed matrix data using Conway's Game of Life rules on Texas Tech's High Performance Computing Center, utilizing multi-threading for efficient computation for large-scale simulations
- Tools Used: Python, TTU HPCC

Recursive Descent Parser   *github.com/Parser ↗*

- Developed a lexeme parser to analyze and validate input phrases or words based on provided rules using efficient parsing algorithms while ensuring no ambiguity
- Tools Used: C, Makefile, VS

Red-Black Tree   *github.com/RBTree ↗*

- Developed a self-sorting red-black tree, integrating algorithms to balance the tree during insertions and deletions
- Optimizes search time by inserting numbers into a self-sorting stack
- Tools Used: C++

Technologies

Languages: C++, Python, Java, C#, SQL, HTML, SCSS, TypeScript, JavaScript/ JSX

Technologies: Git, Jira, Docker, VS/ VS Code, SQL SMS, AWS, Figma, Expo