Application link: https://tiger-moupokqm4-daniel-jones-projects-81b09951.vercel.app/
Github link: https://github.com/daj6-clem/TigerTix/tree/main
Video link: https://youtu.be/YzWkOhFZz_Y

Why is CI/CD important?

Continuous integration and delivery are important because they ensure constant updates while also keeping the program in a deployable state. CI keeps the code updated, meaning all developers are using the same code at all times. CD ensures that code pushed is functional and allows bugs to be immediately located and fixed.

What have we learned?

Security

JH: I learned how to manage accounts and securely store hashed passwords for accounts.

DJ: Hashing passwords and only storing the hashes was interesting. I also learned about JWT security tokens and how they can be used for authentication within a website.

Networks

JH: I learned how to build and host a website online that many users can access.

DJ: I didn't learn a lot about networks beyond the basics, just Client-Server architecture, HTTP requests, databases over networks, and debugging the mess that is frontend/backend interaction.

Accessibility

JH: I learned about implementing ARIA labels. I also learned how to implement voice recognition.

DJ: John did the majority of the work in the accessibility area, so I didn't learn much beyond observing his implementation and making it fit with the rest of our code.

Deployment Problems:

Many of the problems we faced when deploying the project revolved around the database, the sql code, the authentication tokens, and the cors. There are a lot of special allowances that you have to make for a live deployment that you simply don't have to do for local implementations, which resulted in us not realizing we had a problem until much later in the project development. There were some difficulties in getting sqlite 3 to work with the deployed backend and frontend. The database was simply inaccessible for a while, and our eventual solution was to switch to better sqlite and redo half of the backend code. Another issue was the transition from local to deployment itself. A lot of the local urls were baked into our code and had to be replaced with online ones. Environmental variables were also a huge headache to set up, because Vercel and Railway give you a hundred different links and it's hard to tell which ones are what.

TigerTix Ticketing System

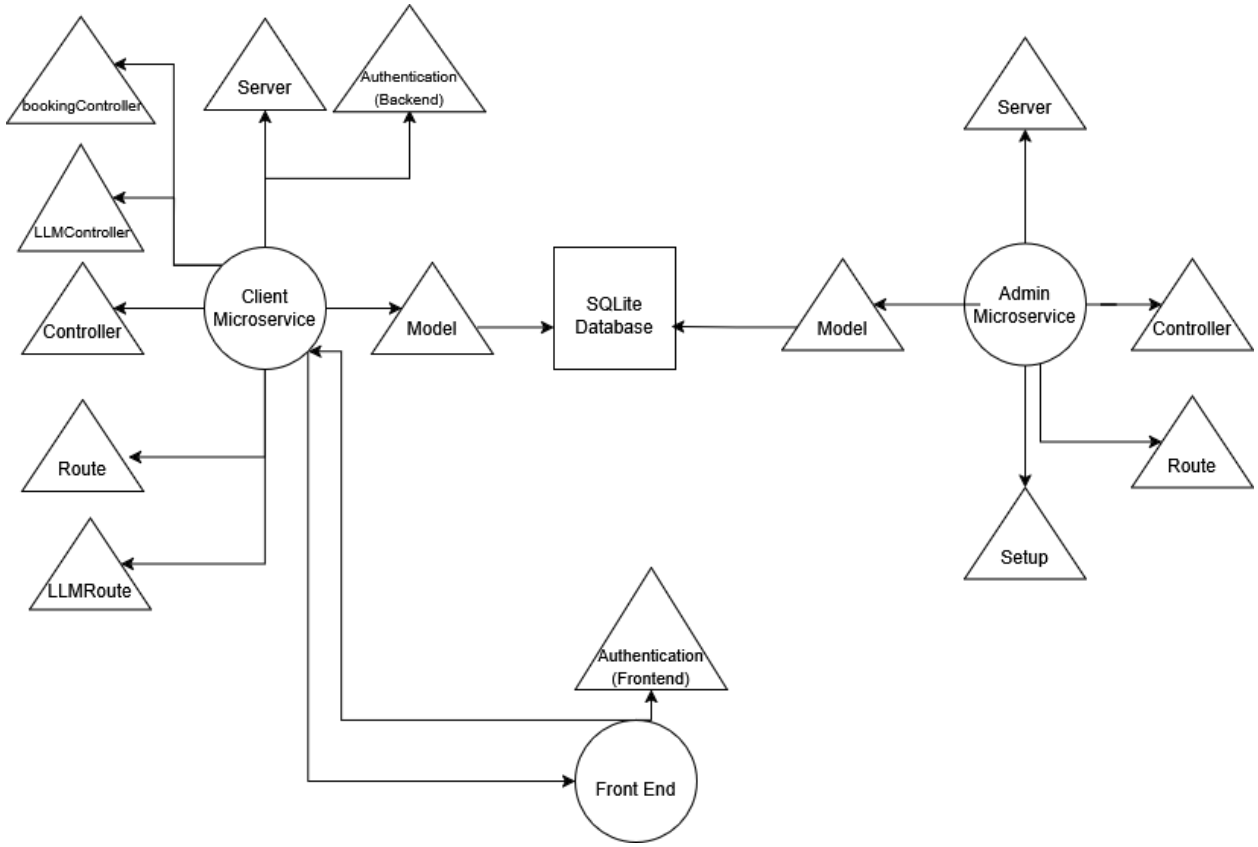
Project Overview

The TigerTix ticketing system is an online ticket and event management system for Clemson University. The system provides a way for students and faculty to purchase tickets to events hosted on campus, and for event hosters to allow tickets to their events to be purchased. The system utilizes user accounts to track tickets purchased and personalize the user experience. Accessibility features such as voice activation and an AI event lookup assistant are provided.

Tech Stack

The frontend is built using React. The backend uses Node.js with Express. Ticket and event data is stored in a shared SQLite database. The AI assistant utilizes OpenAI's ChatGPT. Tests are written using React's built-in testing library as well as Jest.

Architecture Summary



Installation and Setup

Clone the repository from github and ensure all dependencies are installed by running 'npm install' from a privileged terminal (ex. Powershell) inside of each microservice (frontend, client service, admin service). To run locally, run an instance of

each microservice in a separate privileged terminal (for backend this means ‘node server.js’ and for frontend this means ‘npm start’).

Environment Variables Setup

An OpenAI key must be provided in a .env file in the frontend and client microservices in order for the AI assistant to run.

Regression Tests

Ensure React and Jest are installed. Run ‘npm test’ in a privileged terminal inside of each microservice. Tests are also run automatically when new code is pushed to github.

Team Members

John Habakus - Developer
Daniel Jones - Developer
Dr. Julian Brinkley - Instructor
Colt Dolster - TA
Atik Enam - TA

License

Copyright 2025 John Habakus, Daniel Jones

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.