AutoCar

EE461L Group 13

Amy Chang, Julian Fritz, Shanil Jasani, Madilyn Sikorski, and Renzo Teruya

Meet the Team!

Amy Chang
Database

Renzo Teruya
Backend

Shanil Jasani
Frontend

Julian Fritz
Backend

Madilyn Sikorski
Cloud

What is AutoCar?

Easy & Convenient Access to self-driving cars

Short & Long Term, Individual & Group rentals

HW Sets = Cars | Projects = Carshares

DEMO

autocar.pythonanywhere.com

REFLECTIONS

Scalability

- Database
  - Can easily add a new field/attribute to each entry in the database
    - Example: Want to implement billing with credit card information
      - Add a credit card attribute to the User entry
      - Encrypt the number
  - Add more cars to the catalog
- User Interface
  - Dynamic loading allows us to have as many or as little cars available, with as many users and rentals as possible
  - Mobile-friendly UI could be turned into an iOS/Android app
  - Search functionality

Modularity

- Backend
  - Main autocar module that creates the app and blueprints for additional modules
  - MongoDB module easily accessible from any file
  - New features can easily be added to the app with a new blueprint
- Frontend
  - HTML templates all extend one site template
  - Each page has its own template for easy backend access
- Refactoring
  - Refactored database entries to add billing, real time tracking
  - Refactored UI to make visual improvements
  - Refactored backend python files for better organization

Testing

- Pytest unit tests
  - Tests covered all backend python functions
- Manual testing
  - All MongoDB functions tested manually using MongoDB Compass and function calls
  - All backend functions tested manually using the website
- Integration testing
  - Manually tested by trying out all the website's features
  - All the website's features working properly indicates modules working together correctly

Improvements

- Database
  - Can create more functions that will help clean up the flask application code
  - More documentation for easier readability
    - When calling a database function, it prompts the type of argument needed and provides a description of what the function does
- Frontend
  - UI enhancements including clearer directions and animations
  - Minimize contrasting colors (ex. Dark blue on a dark orange background)
- Backend
  - More features to implement
    - Examples:
      - Google Maps Integration
      - Billing in dollars (vs credits) with credit card merchant like Stripe API

What went well?

Frontend

- Consistent branding of site with colors, fonts, and layouts

Teamwork & Dynamic

- Working as a team, delegating work to team members
- Completing tasks early and efficiently, leaving much time for testing and adding aesthetics

Deployment

- Deploying the application to PythonAnywhere, over other platforms
- Integrating each of the components of the project to form the deployed application

What could we have done better?

- More object-oriented approach to the backend
  - Classes
- More consistent use of Github
  - Branches, commits, issues
- Maintain more accurate logs of everything that is going right/wrong throughout the process
- Brainstorm more detailed tasks to do at the start of the project
- Have the database perform more intuitive functions
- Ex. checkpoint 2

What did we learn?

- A great deal about Flask, HTML, MongoDB, and cloud deployment
- What may be an obvious workflow to one, may be a complicated and hidden one to another user
  - Importance of good documentation
- Communication skills as a team
  - Weekly meetings, daily text updates
- When one approach is not working out for the integration of the code, there are thousands more to try
- Do not close ourselves off to one option of platforms to use, explore all options
  - Example: Google Cloud & Heroku → PythonAnywhere

Attributions

- https://vpic.nhtsa.dot.gov/api/vehicles/
  - vehicle datasets & example API calls
- https://www.freecodecamp.org/news/learn-bootstrap-4-in-30-minute-by-building-a-landing-page-website-guide-for-beginners-f64e03833f33/
  - Frontend Design
- https://unsplash.com/@introspectivedsgn
  - Free images
- https://www.pythonanywhere.com/forums/
  - Documentation used to configure issues with cloud deployment
- https://flask.palletsprojects.com/en/1.1.x/
  - Documentation and tutorial for flask
- https://jinja.palletsprojects.com/en/2.11.x/
  - Documentation for Jinja with examples
- https://pymongo.readthedocs.io/en/stable/
  - Pymongo Documentation for database API interaction
- https://docs.mongodb.com/manual/reference/method/js-collection/
  - MongoDB Collection methods & examples
- https://stackoverflow.com/questions/17057191/redirect-while-passing-arguments
  - How to pass messages through url and as session variables
- https://www.w3schools.com/html/
  - HTML tutorial & examples

Q&A