Festo:

AI Assisted Rapid Prototype Generator for DevOps

Luke Fitzpatrick, Mreedul Gupta,
James Hautekeete, Erik Hirschmann,
Harika Kondur, Zane Perry

Objective

We were tasked with creating an AI generator for React and Python applications to aid DevOps with the initial prototyping phase of development.

Solution

Upload

Users can upload Epics, User Stories, and any other desired code files.

This includes files generated by the previous Team!

Generate

Through GPT, code is generated.

Users interact with a React frontend that connects to a Python server.

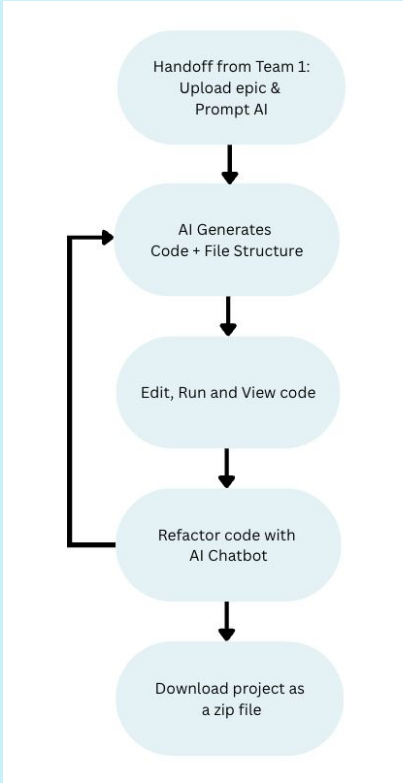
View and Edit

Implemented a functioning code environment to view and edit all generated code files.

Export

Separated Code into a pre built file structure so it can be downloaded and run with ease.

Architecture & Flowchart




Demo

Challenges

Fine-tuning AI outputs

- Much of our time was spent fine-tuning AI output and input prompts in order to have consistency across all generated code files.

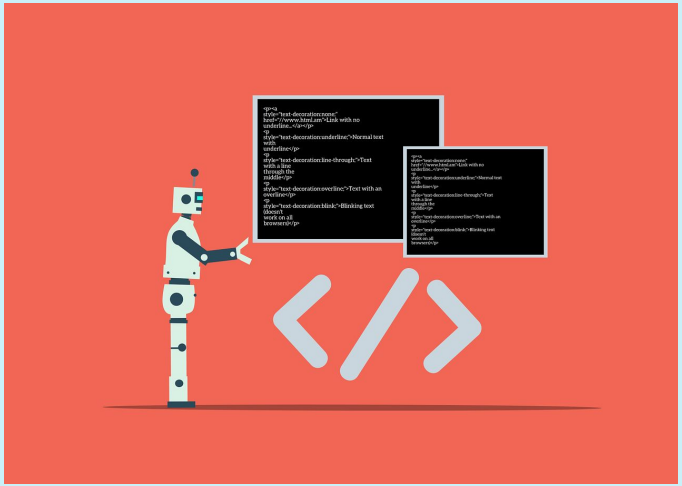
Implementing a live in-browser development environment

- This was an essential part of our application. Users should easily and quickly view the generated prototype. Lot's of time went into research how to achieve this. And even more time into debugging to get it effectively running.
- Generated code had to also be in a format that could be run on the fly.

Lessons Learned

Artificial intelligence is a great tool! Through this project we discovered it is not perfect on it’s own.

Generated code often contained many small errors with simple solutions that developers can resolve.



Future Recommendations

AI Improvements

- Try different models
- Increase Token Limit
- Incorporate the Festo skillground

Model Validations

- Test AI outputs to ensure correctness
- Possible Multi-Agent Learning

Support for other codebases

- Challenges with the viewer environment
- Presents challenges with cross-over

Enhanced Iteration

- User edits passed back to AI
- Prompted suggestions from the model

Any Questions