MICHIGAN STATE
UNIVERSITY

Project Plan Presentation
Visualizing Neural Network Gradients

The Capstone Experience

Team Magna VNNG
Devin Dematto
An Le
Arhan Mulay
Sarah Regan
Alex Stornant
Don Truong

Department of Computer Science and Engineering
Michigan State University

Fall 2024

From Students…
…to Professionals

Project Sponsor Overview

- Magna is the largest automotive parts manufacturer in North America
- Specializing in various parts such as body exteriors & structures, powertrains, chassis, electronics, and more
- Leaders in innovation and sustainability in the automotive industry
- Employs over 150,000 people

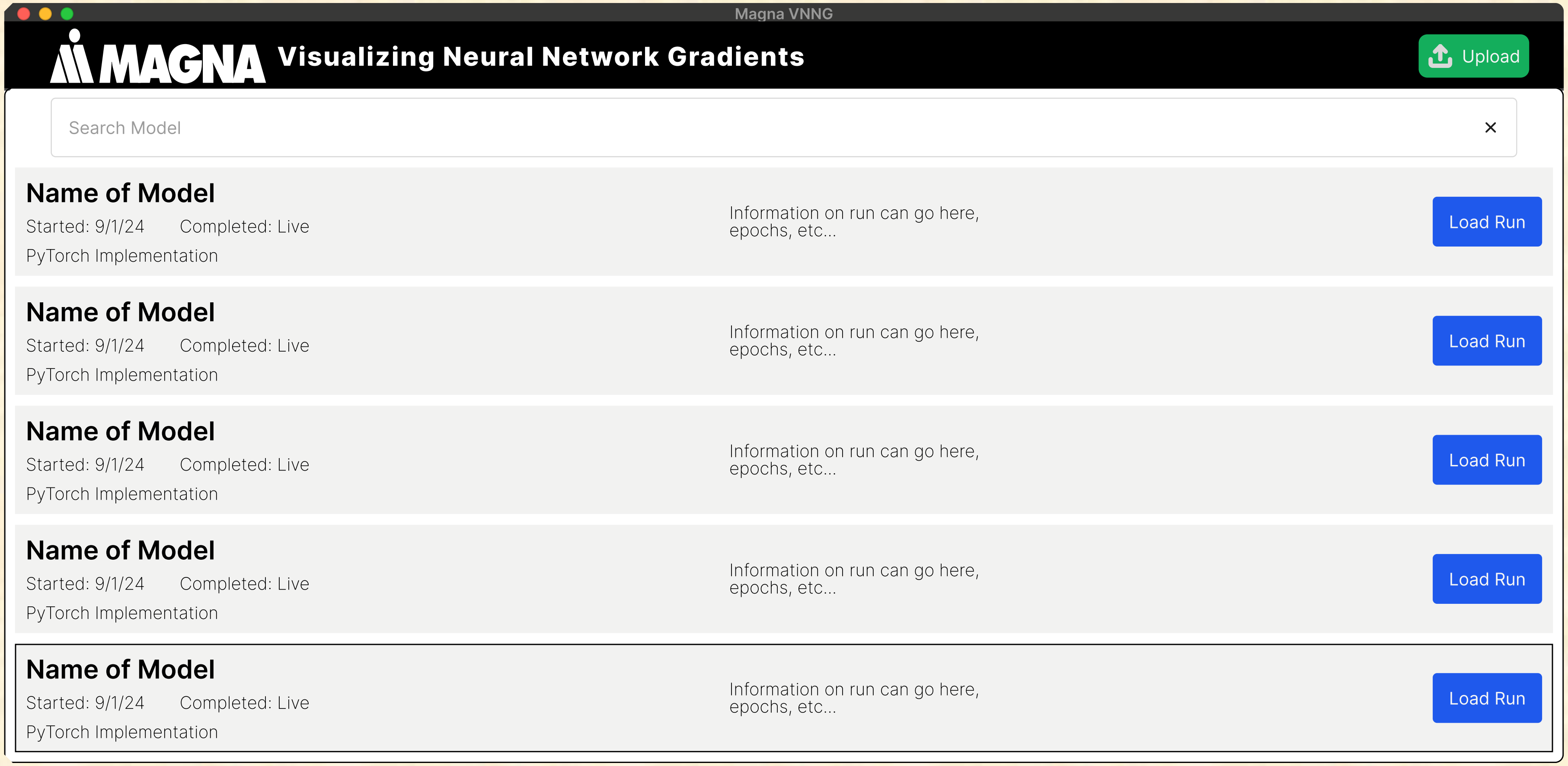
Project Functional Specifications

- Develop an interactive UI for visualizing the training of machine learning models
- Various ML models operate as “black boxes” and we hope to allow a better understanding of the behind the scenes of these models
- Engineers working with all types of machine learning will use our service to debug and understand their models better

Project Design Specifications

- VNNG application serves as a hub for accessing and managing visualization files
- Users easily integrate a library into their ML training scripts to create visualization files
- Upon selecting a visualization file, users can see high level metrics and information, and have access to more specific tools for visualization
- Our primary tool for visualizing gradient and weight changes is our interactive 3D heatmap, which allows users to explore the model in a 3D space

Screen Mockup: Home Page



Magna VNNG

MAGNA Visualizing Neural Network Gradients

Upload

Search Model ×

Name of Model
Started: 9/1/24 Completed: Live
PyTorch Implementation
Information on run can go here, epochs, etc...
Load Run

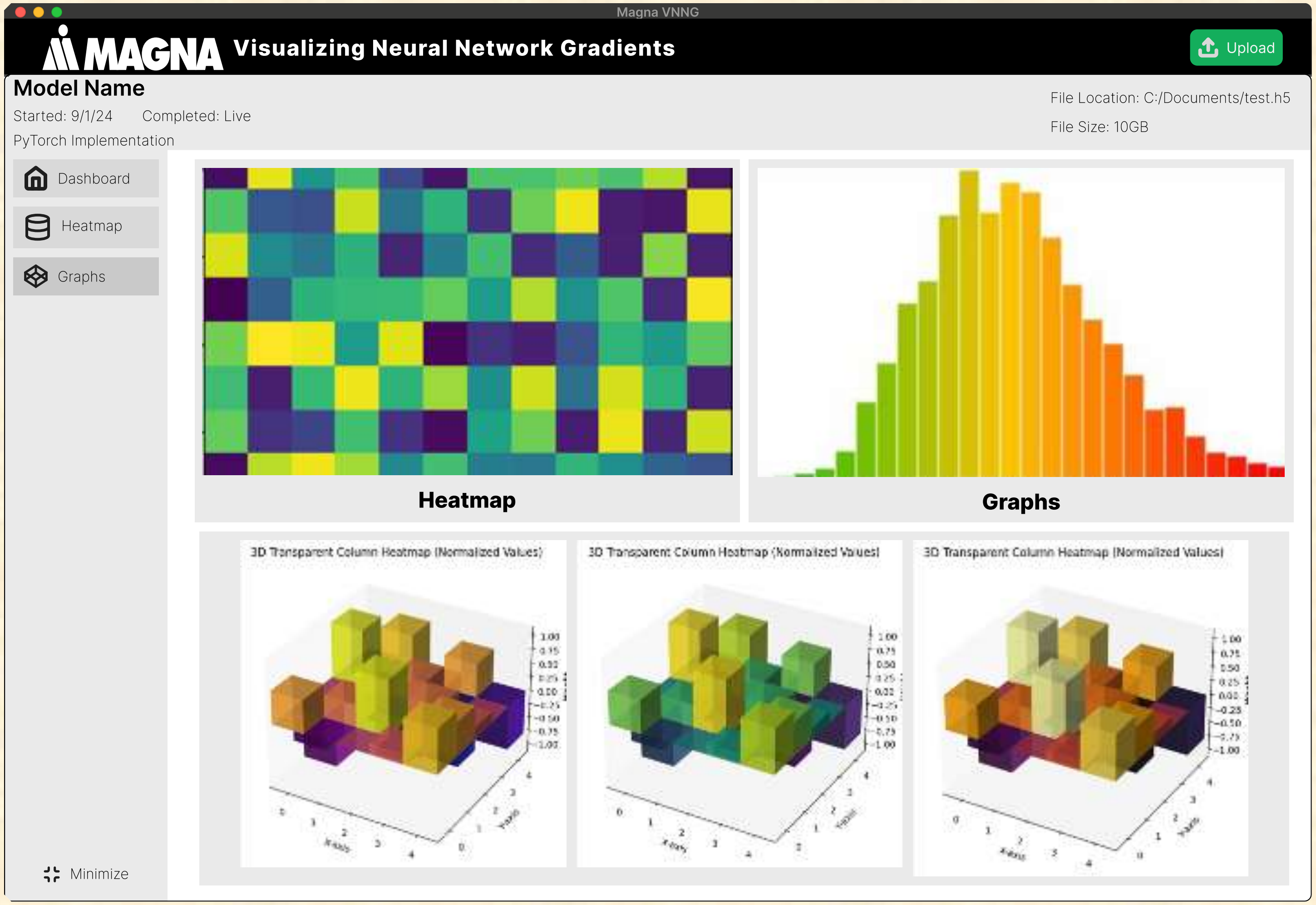
Name of Model
Started: 9/1/24 Completed: Live
PyTorch Implementation
Information on run can go here, epochs, etc...
Load Run

Name of Model
Started: 9/1/24 Completed: Live
PyTorch Implementation
Information on run can go here, epochs, etc...
Load Run

Name of Model
Started: 9/1/24 Completed: Live
PyTorch Implementation
Information on run can go here, epochs, etc...
Load Run

Name of Model
Started: 9/1/24 Completed: Live
PyTorch Implementation
Information on run can go here, epochs, etc...
Load Run

Screen Mockup: Dashboard


Magna VNNG
MAGNA
Upload
Dashboard
Heatmap
Graphs
Minimize
Heatmap
Graphs
3D Transparent Column Heatmap (Normalized Values)
3D Transparent Column Heatmap (Normalized Values)
3D Transparent Column Heatmap (Normalized Values)

Visualizing Neural Network Gradients

Model Name

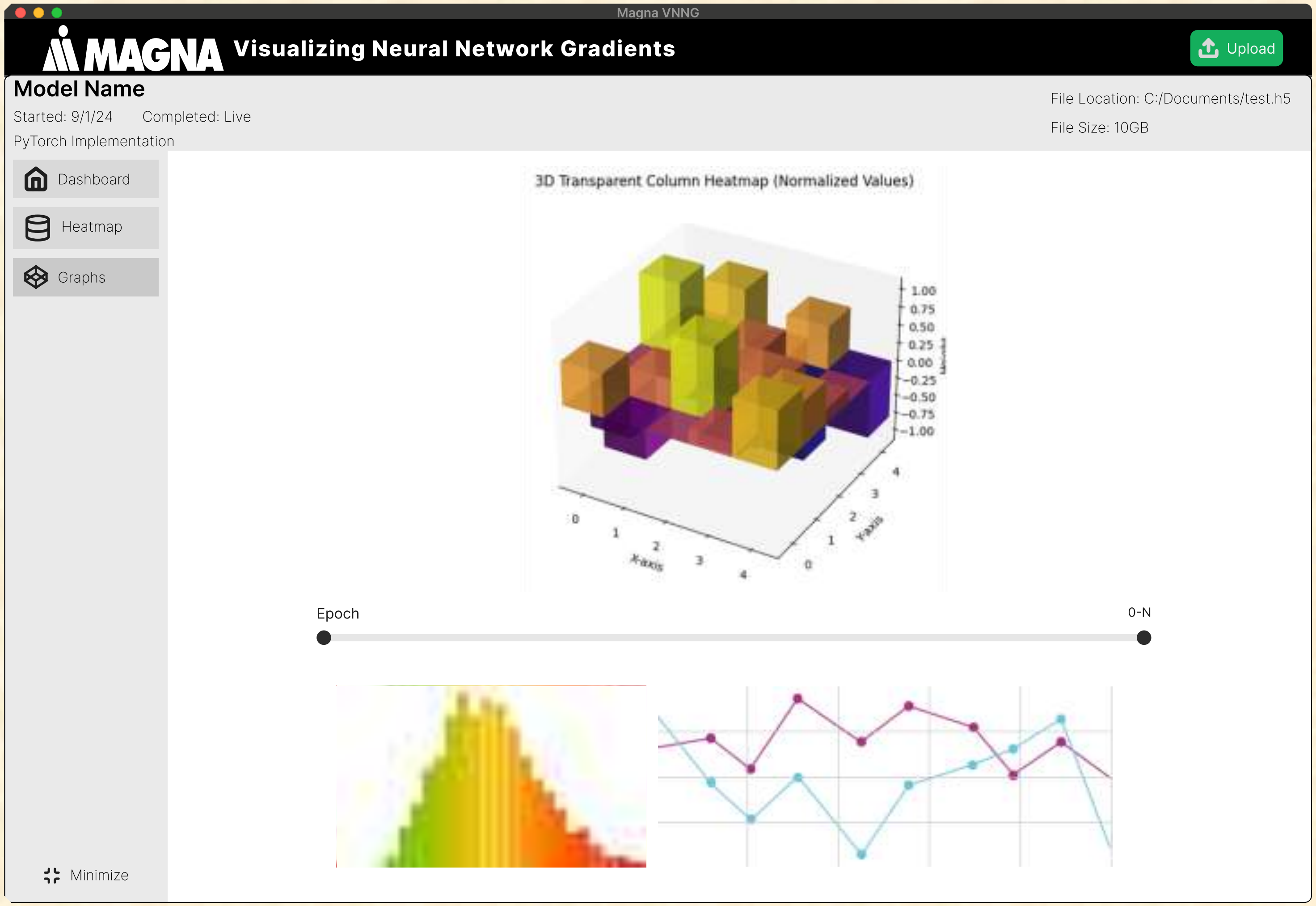
Started: 9/1/24 Completed: Live

PyTorch Implementation

File Location: C:/Documents/test.h5

File Size: 10GB

Screen Mockup: Heatmap



Model Name

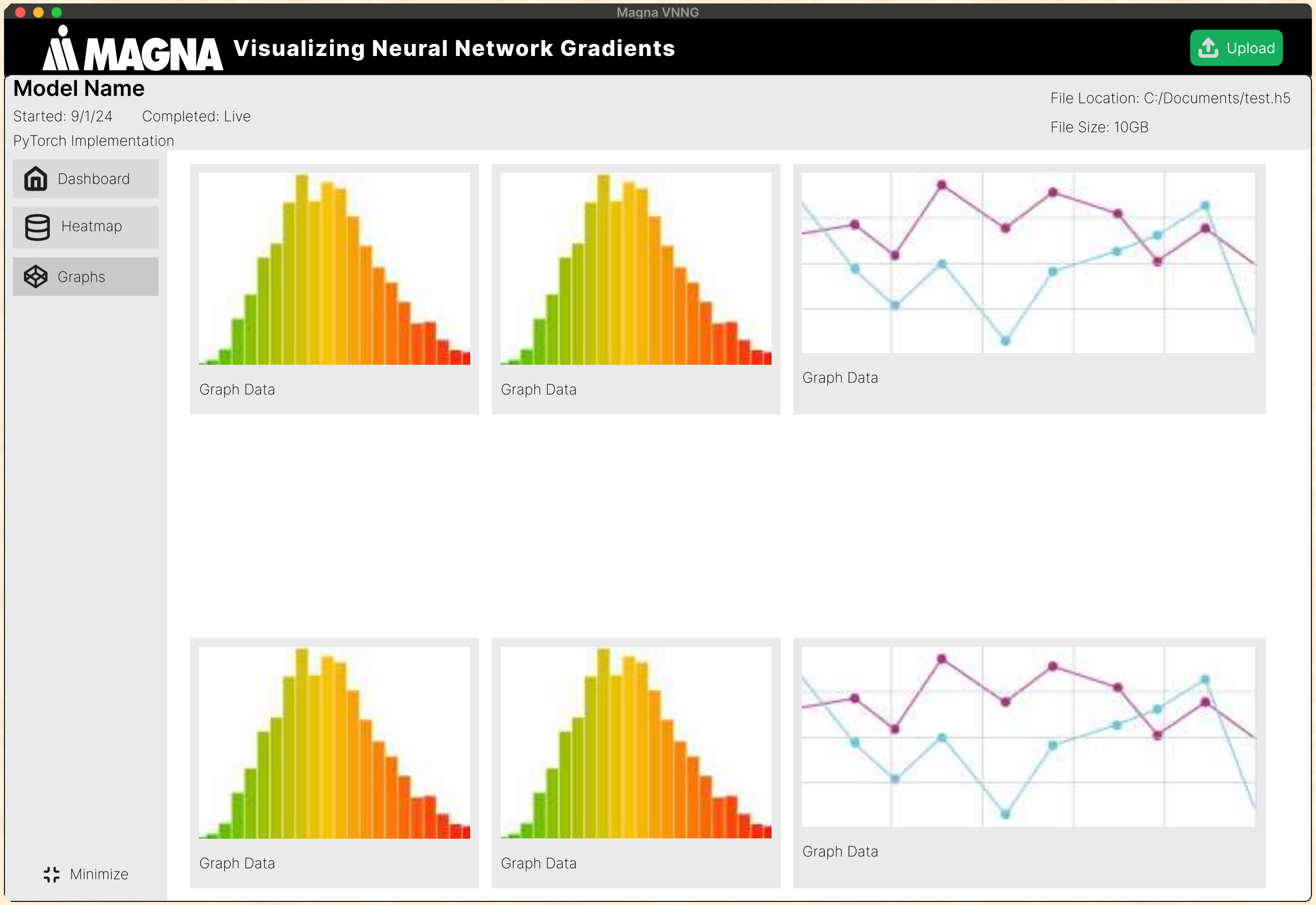
Started: 9/1/24 Completed: Live

PyTorch Implementation

File Location: C:/Documents/test.h5

File Size: 10GB

Screen Mockup: Graphs



Model Name

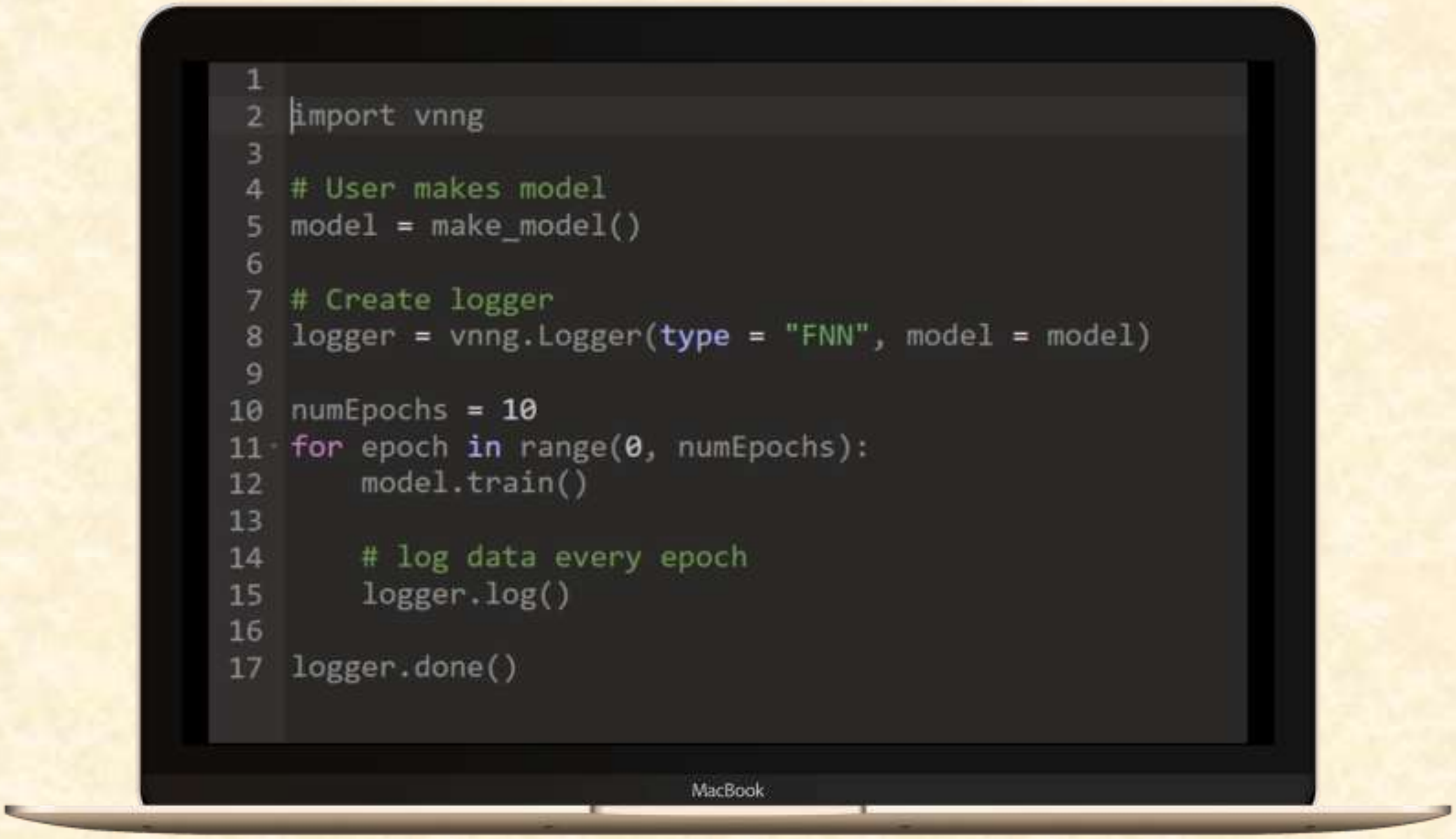
Started: 9/1/24 Completed: Live

PyTorch Implementation

File Location: C:/Documents/test.h5

File Size: 10GB

Screen Mockup: Library Pseudocode

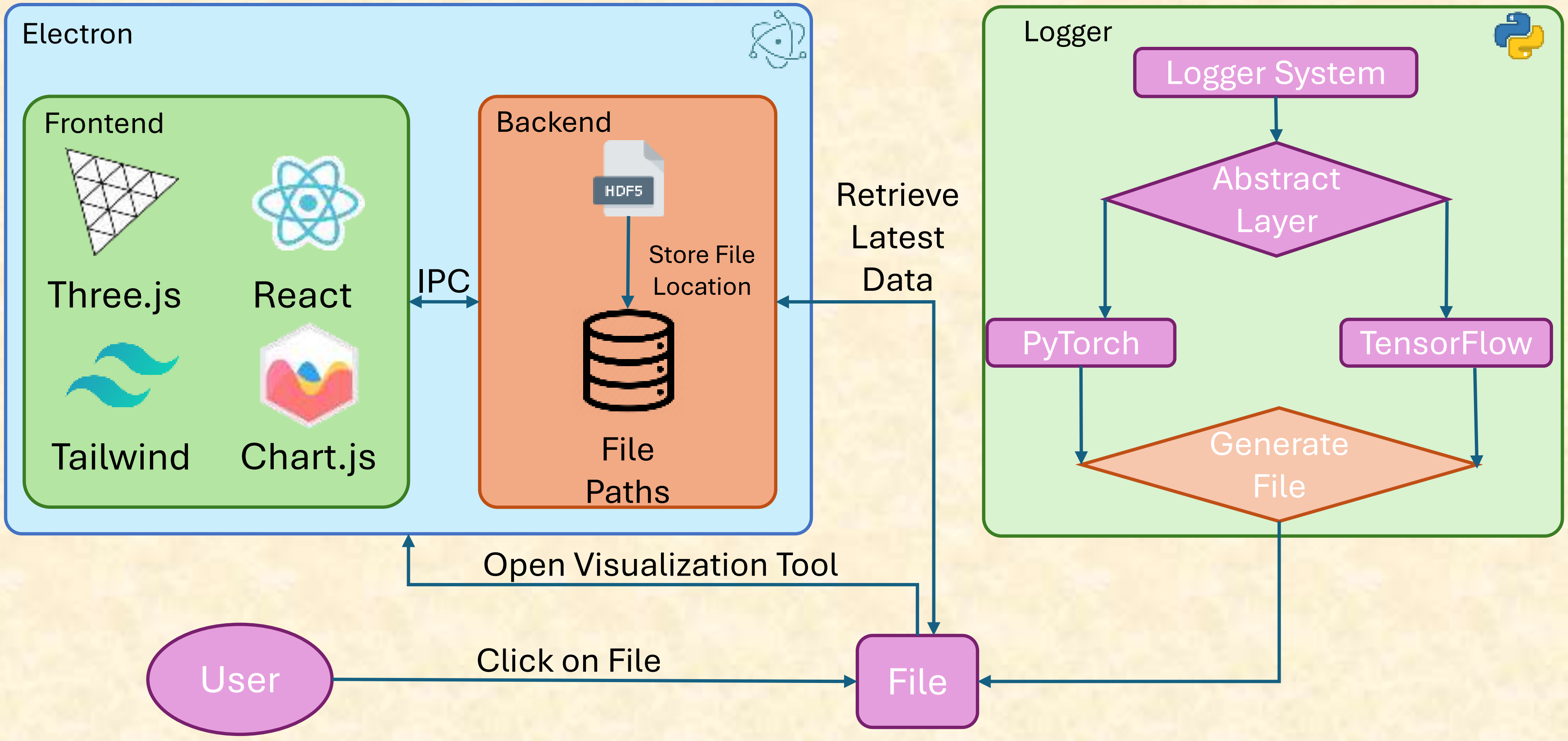


```

import vnng

# User makes model
model = make_model()

# Create logger
logger = vnng.Logger(type = "FNN", model = model)

numEpochs = 10
for epoch in range(0, numEpochs):
    model.train()

    # log data every epoch
    logger.log()

logger.done()
```

Project Technical Specifications

- Logger
  - Easy to integrate Python library to extract metrics from ML training
  - Stores results in HDF5 file for easy and efficient storage
- Frontend
  - Electron app used to visualize the metrics from the logger, created with React, Chart.js, Three.js, and tailwind

Project System Architecture



Project System Components

- Hardware Platforms
  - N/A
- Software Platforms / Technologies
  - Electron
  - Node.js
  - React + Tailwind
  - Three.js + Chart.js
  - PyTorch and Tensorflow
  - HDF5

Project Risks

- Problematic layer detection
  - Identifying what is a problem with layers such as exploding gradients, vanishing gradients, nonconverging models, etc, is difficult, especially with more complex models
  - Implement algorithms to track gradients across layers and intervals and flag abnormal behavior
- Performance Issues
  - Slow processing times due to large amounts of data can degrade user experience
  - Implement efficient data storage and use data sampling techniques (e.g., mean, max, min) to reduce data size
- Scalability
  - As model complexity increases so does the difficulty of representing the model in a meaningful way
  - Use dynamic visualization techniques that adjust based on model complexity, such as adaptive zooming, filtering, aggregation, and highlighting key layers. Also, allows users to customize the visualization to focus on specific layers or metrics relevant to their needs
- Generalizing Varying Network Architectures
  - The diversity in network architectures (e.g., FNN, CNN, RNN) presents a challenge in creating a generalized log parsing and visualization approach
  - Focus initially on supporting common network architectures such as Feedforward Neural Networks (FNNs) and simple Convolutional Neural Networks (CNNs). Once these are robustly handled, work on developing more generalized solutions that can accommodate a wider range of architectures

Questions?

? ? ? ?

? ?

? ? ?