

# Crash Rate Prediction from Traffic Volume Data using AI

Jonathan Le, Travis Chan, Syeda Afia Z. Hossain, Arham Asam, Devanshu Khadka

CS4624 Multimedia, Hypertext, and Information Access

Professor Mohamed Farag

Client: Dr. Mohamed Farag

Virginia Tech, Blacksburg, VA 24061

December 5, 2024

# Outline

1. Project Overview
2. System Architecture
3. Frontend Development
4. Demonstration
5. Backend Development
6. Machine Learning Model Development
7. Project Deployment Challenges
8. Deliverables
9. Future Plans
10. Lessons Learned
11. Acknowledgements
12. References

# Project Overview

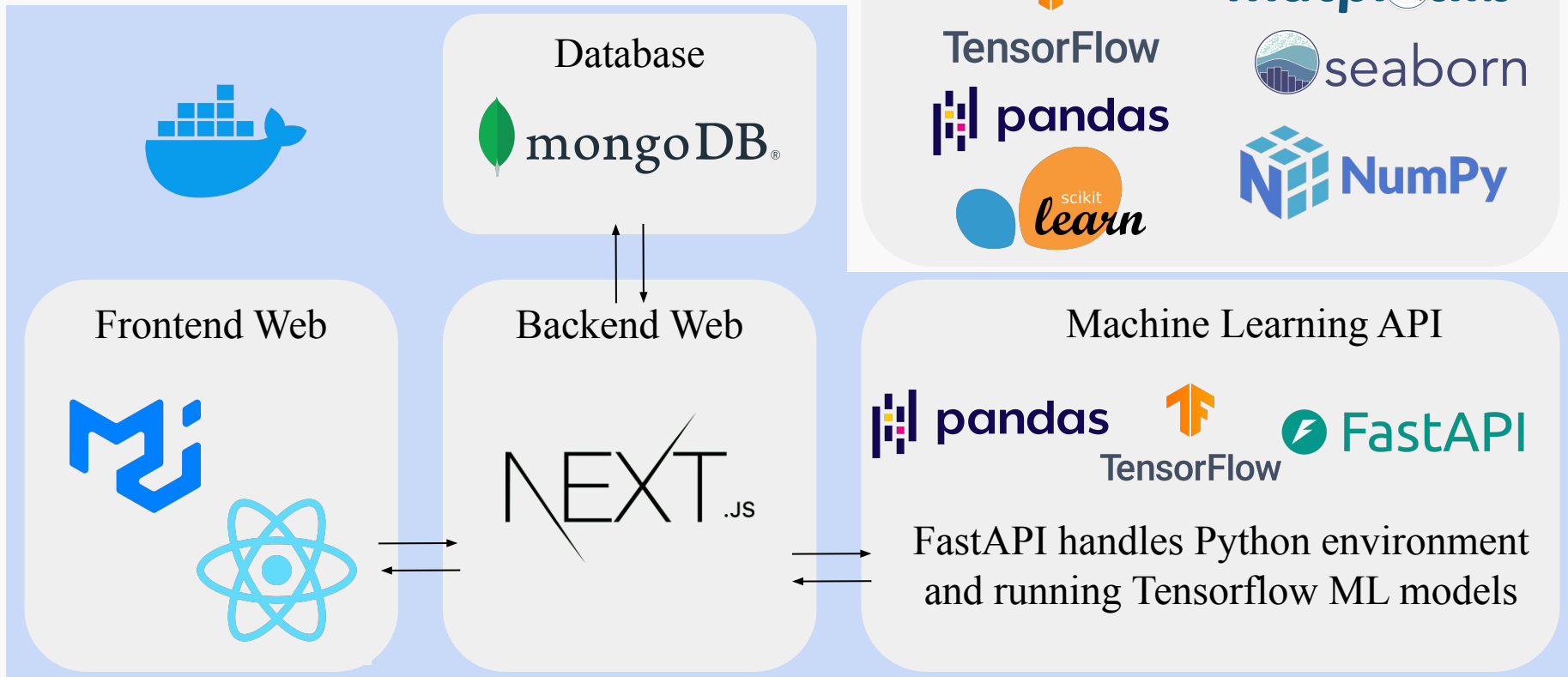
## **Problem: Highway Safety Challenges**

- Frequent Motor Vehicle Crashes; significant impact on public safety and infrastructure.
- Current methods for predicting crash rates lack advanced analytics tools
- This limitation makes it difficult for authorities to take proactive steps

## **Solution**

- Develop web application allowing users to easily leveraging ML models
- Analyze and predict crash rates
- Allow users to publish data and visualize crash predictions

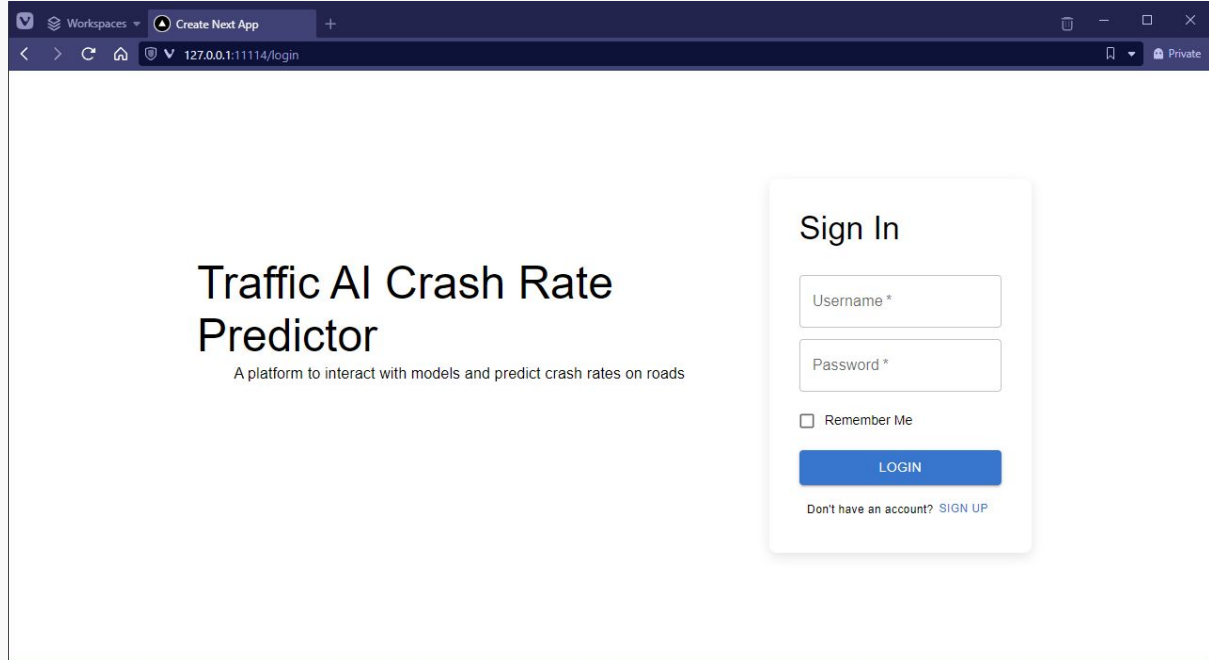
# System Architecture



# Web Application Development

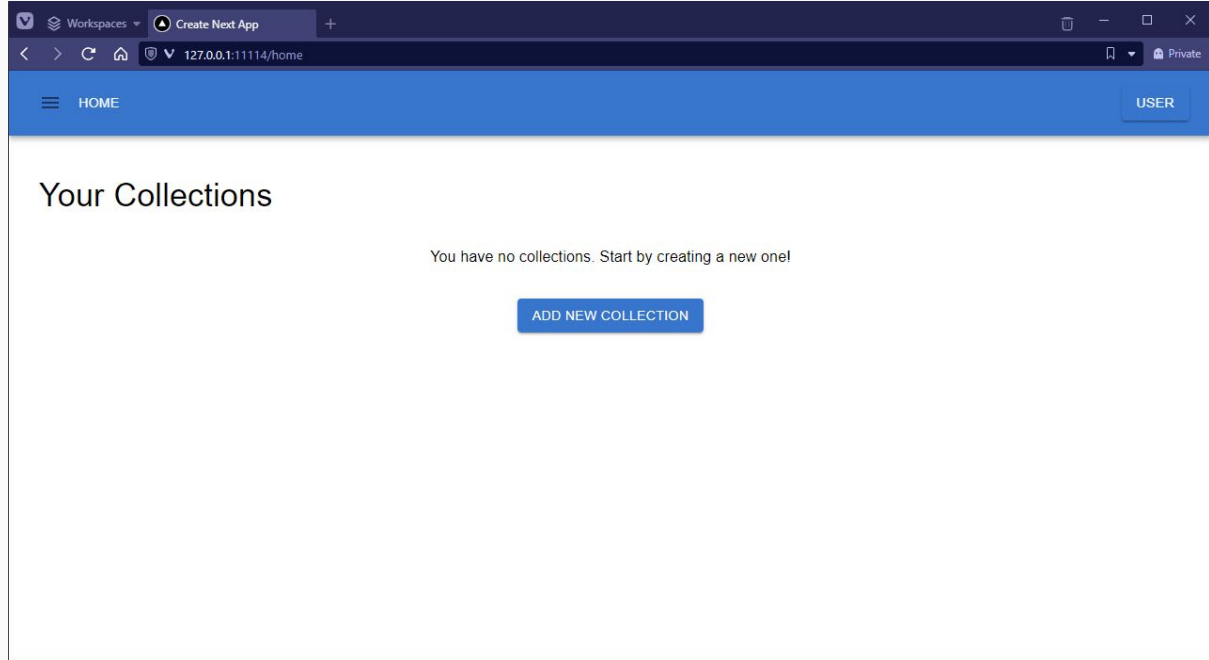
# Login/Sign up

- User
- Admin



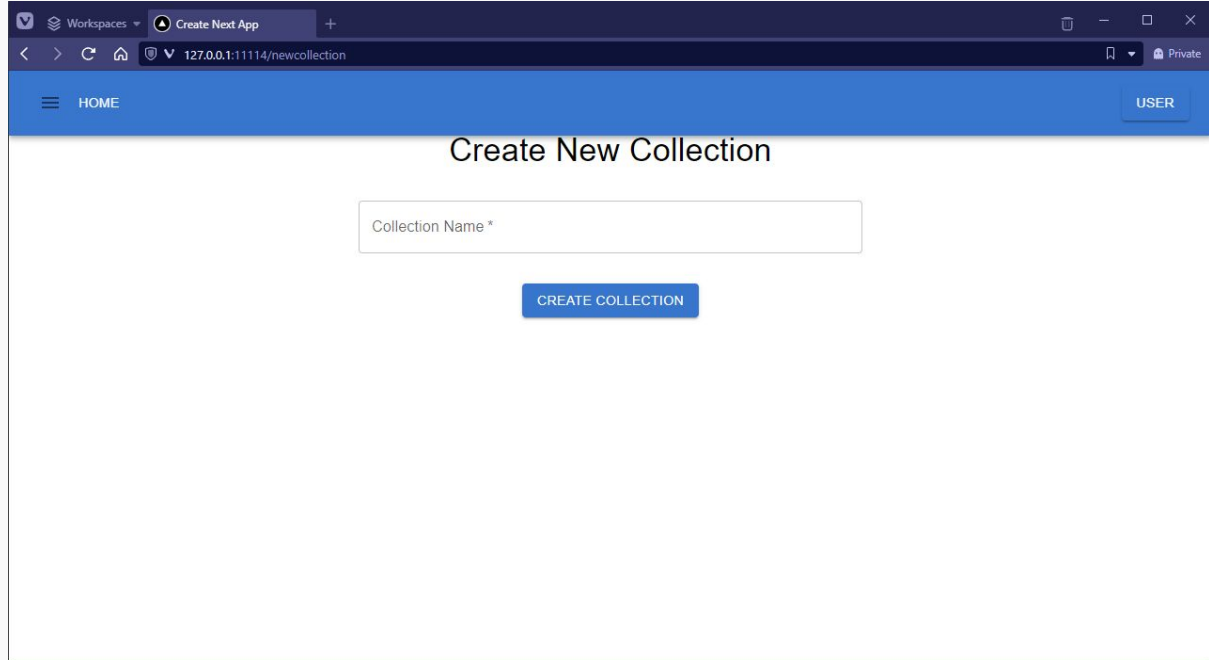
# Home (User/Admin)

- ‘Add New Prediction’



# New Collection Name (User/Admin)

- Create a name
- ‘Create Collection’

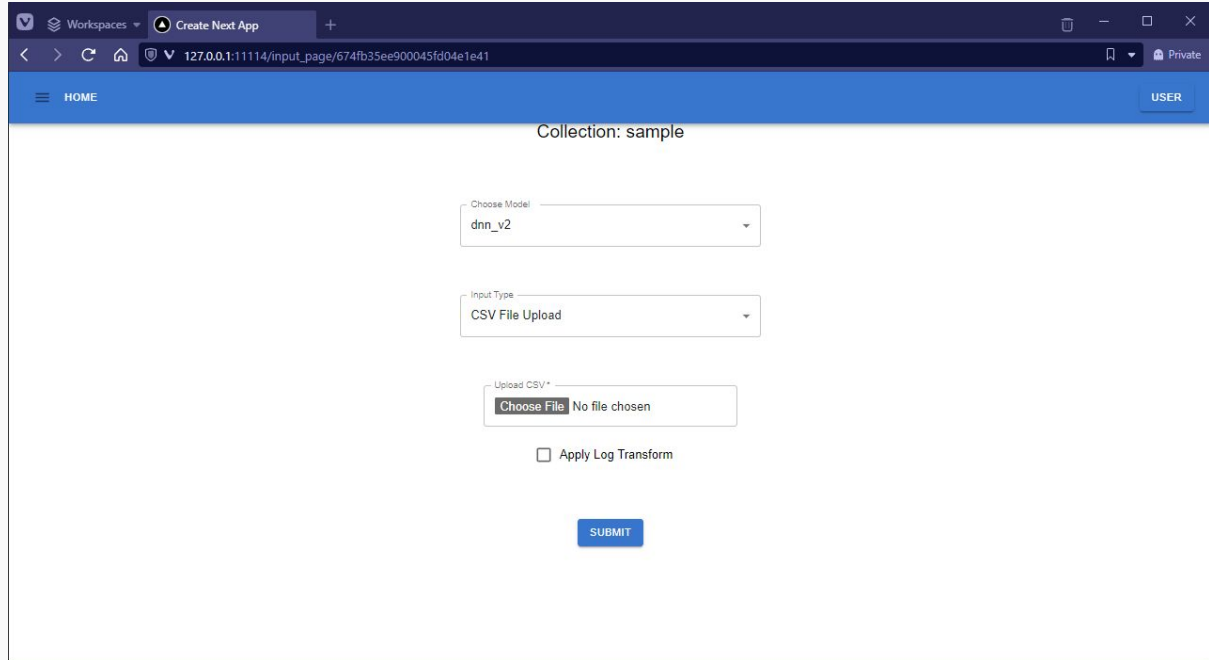


The screenshot shows a web browser window with the address bar displaying '127.0.0.1:11114/newcollection'. The page title is 'Create New Collection'. The interface features a blue header bar with a 'HOME' link on the left and a 'USER' profile icon on the right. Below the header, the main content area is white and contains a single text input field labeled 'Collection Name \*'. Below the input field is a blue button with the text 'CREATE COLLECTION'.



# New Prediction (User/Admin)

- Choose Model
- Input Type
  - **CSV file (default)**
  - Manual Input
- Apply Log Transform
- ‘Submit’

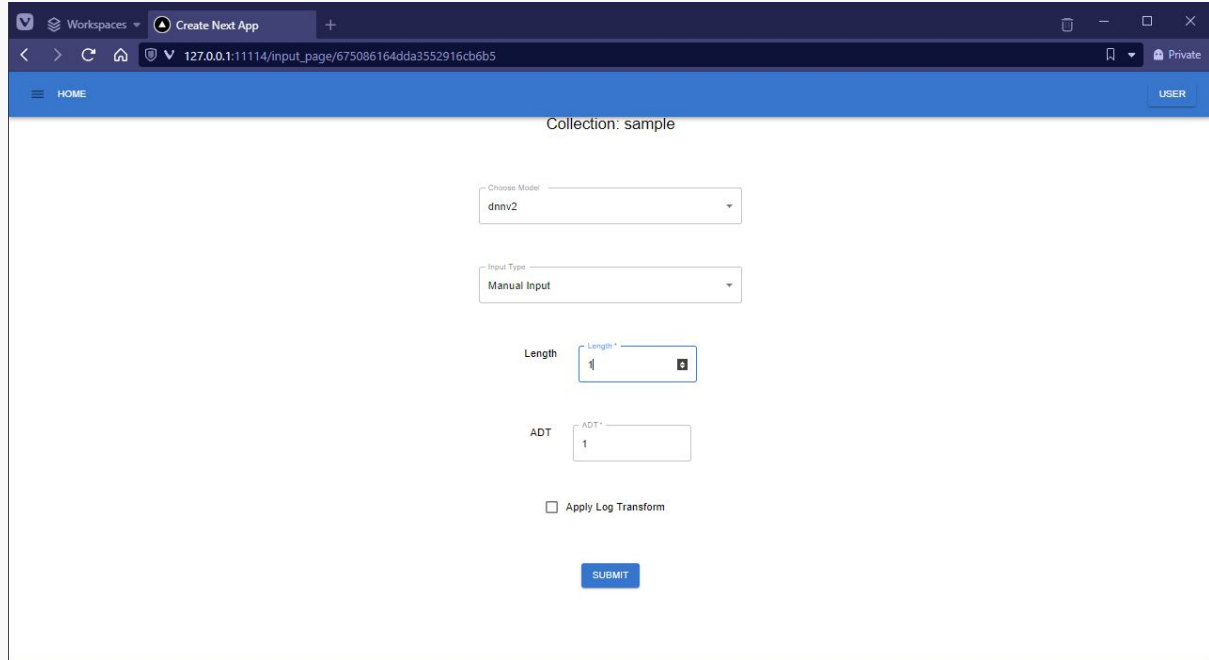


The screenshot shows a web browser window with the address bar displaying '127.0.0.1:11114/input\_page/674fb35ee900045fd04e1e41'. The page has a blue header with 'HOME' on the left and 'USER' on the right. Below the header, the text 'Collection: sample' is centered. The main content area contains a form with the following elements:

- A dropdown menu labeled 'Choose Model' with 'dnn\_v2' selected.
- A dropdown menu labeled 'Input Type' with 'CSV File Upload' selected.
- An 'Upload CSV\*' section with a 'Choose File' button and the text 'No file chosen'.
- An unchecked checkbox labeled 'Apply Log Transform'.
- A blue 'SUBMIT' button at the bottom.

# New Prediction (User/Admin)

- Choose Model
  - CSV file (default)
  - **Manual Input**
- Apply Log Transform
- ‘Submit’

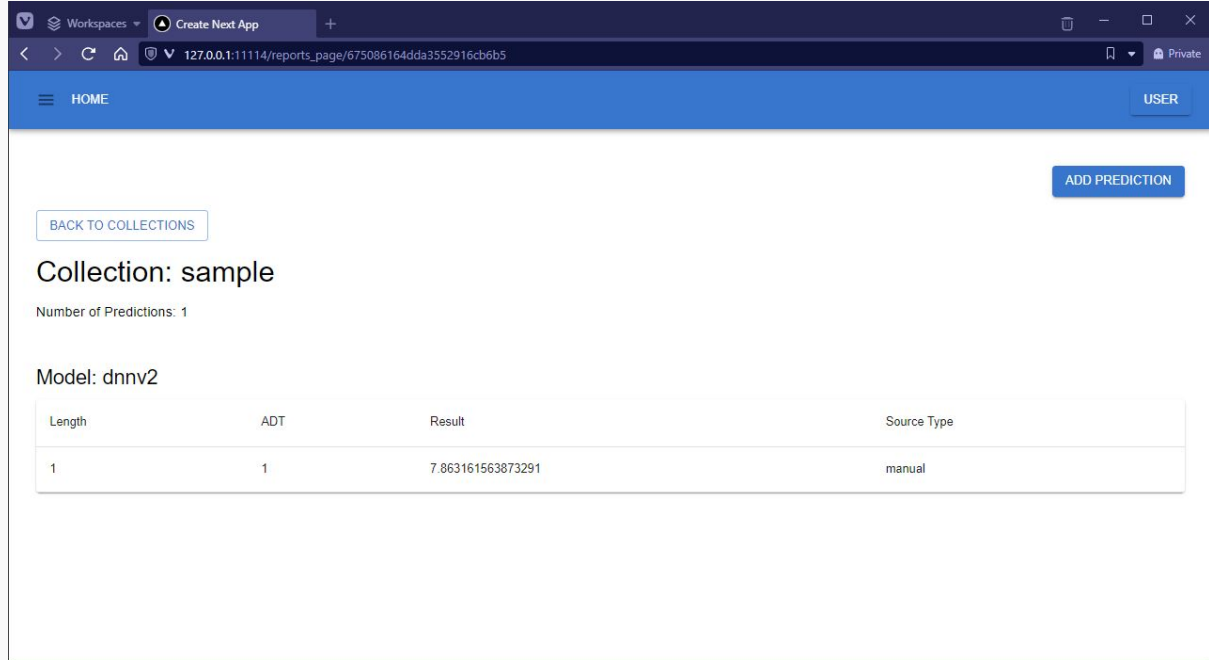


The screenshot shows a web browser window with the address bar displaying "127.0.0.1:11114/input\_page/675086164dda3552916cb6b5". The page title is "Create Next App". The main content area has a blue header with "HOME" on the left and "USER" on the right. Below the header, the text "Collection: sample" is centered. The form contains the following elements:

- A dropdown menu labeled "Choose Model" with "dnnv2" selected.
- A dropdown menu labeled "Input Type" with "Manual Input" selected.
- A text input field labeled "Length" with the value "1" and a small square icon to its right.
- A text input field labeled "ADT" with the value "1".
- A checkbox labeled "Apply Log Transform" which is currently unchecked.
- A blue "SUBMIT" button at the bottom.

# View Prediction Report (User/Admin)

- Attributes
- Result
- Input/Source Type
- ‘Add Prediction’
- ‘Back To Collection’



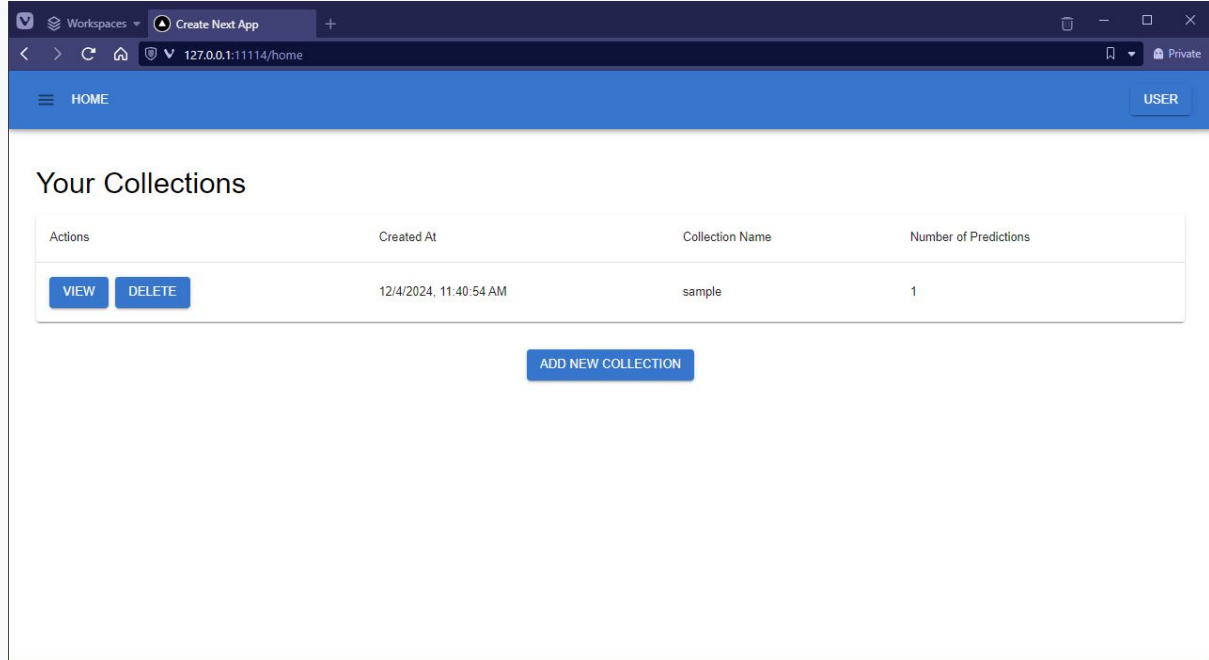
The screenshot shows a web browser window with the following elements:

- Browser tabs: "Workspaces" and "Create Next App".
- Address bar: "127.0.0.1:11114/reports\_page/675086164dda3552916cb6b5".
- Header: "HOME" on the left and "USER" on the right.
- Buttons: "BACK TO COLLECTIONS" (light blue) and "ADD PREDICTION" (dark blue).
- Text: "Collection: sample" and "Number of Predictions: 1".
- Text: "Model: dnnv2".
- Table with 4 columns: Length, ADT, Result, Source Type.

Length	ADT	Result	Source Type
1	1	7.863161563873291	manual

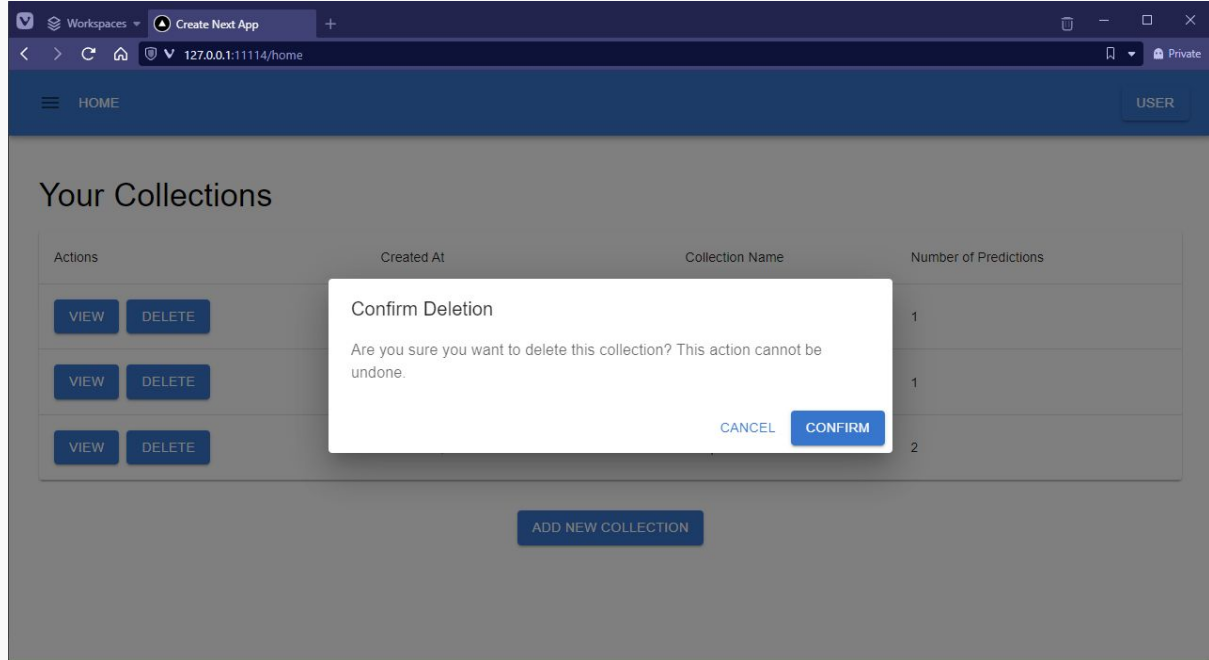
# Home (User/Admin)

- New Collection Added
- Date, Time
- Name
- No. of Predictions
- ‘View’
  - Prediction Report Page
- ‘Delete’
  - Popup to confirm deletion



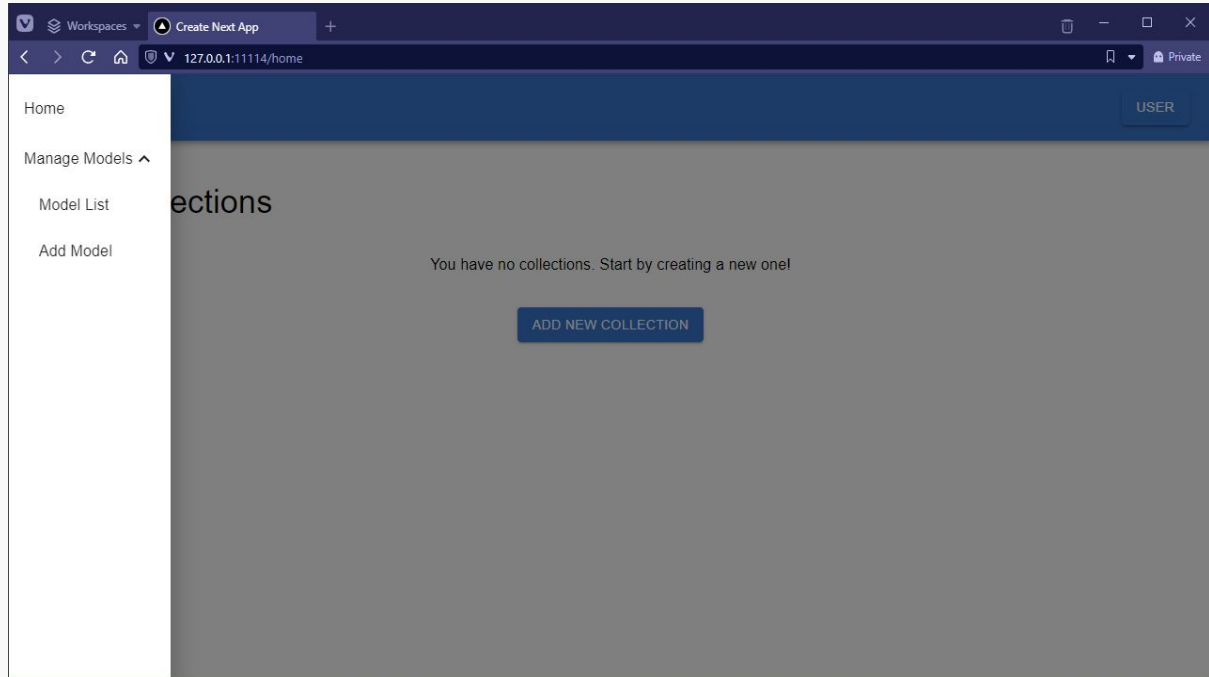
# Home (User/Admin)

- New Collection Added
- Date, Time
- Name
- No. of Predictions
- ‘View’
  - Prediction Report Page
- ‘Delete’
  - **Popup to confirm deletion**



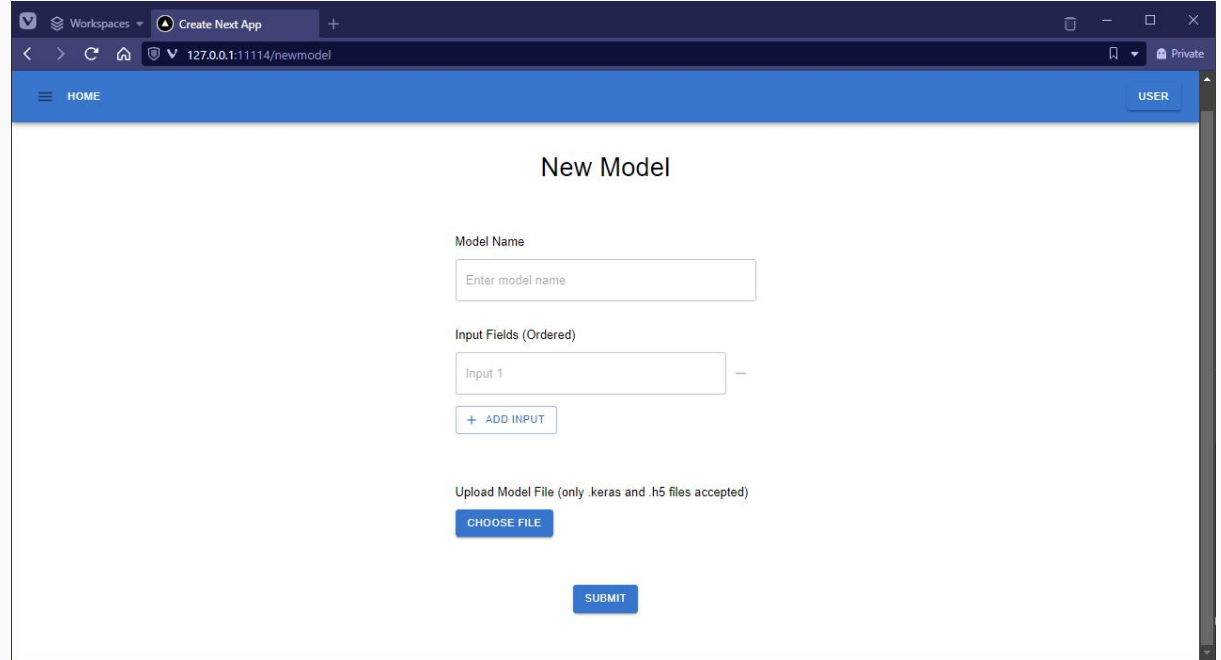
# Home (Admin)

- Add a new model
- Manage current models



# Add Model (Admin)

- Model name
- Ordered list of input fields
- Choose Model file(.keras and .h5)

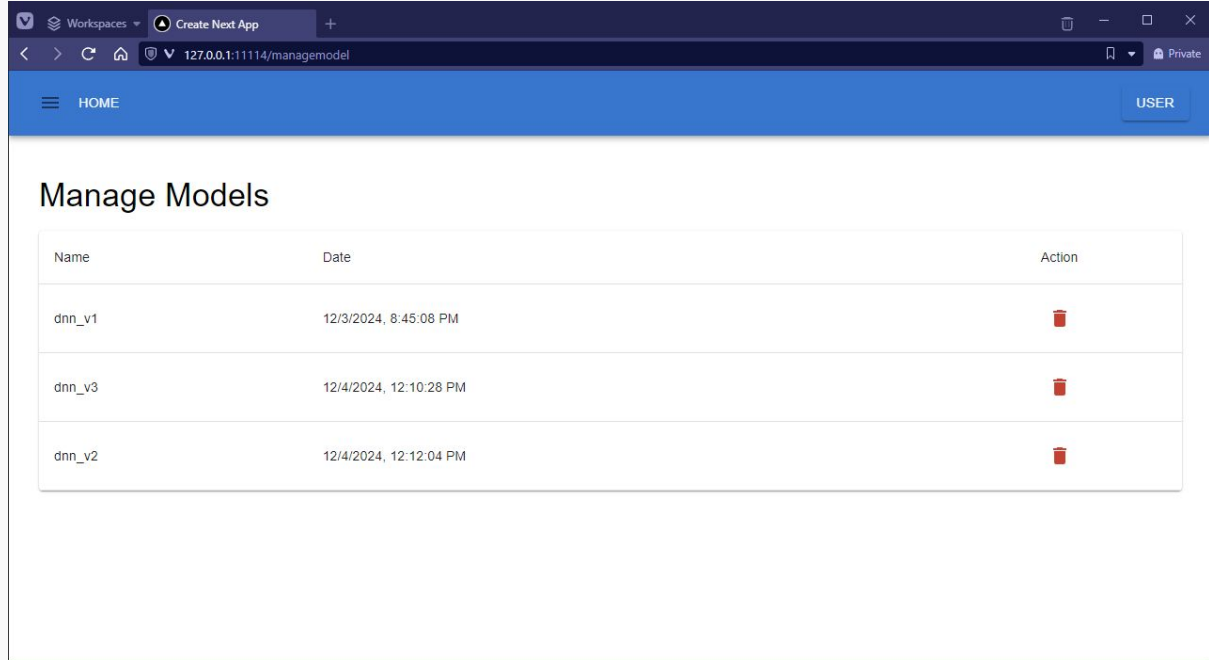





The screenshot shows a web browser window with the address bar displaying "127.0.0.1:11114/newmodel". The page has a blue header with "HOME" on the left and "USER" on the right. The main content area is titled "New Model" and contains the following form elements:

- Model Name:** A text input field with the placeholder text "Enter model name".
- Input Fields (Ordered):** A text input field with the placeholder text "Input 1" and a minus sign on the right. Below it is a button labeled "+ ADD INPUT".
- Upload Model File (only .keras and .h5 files accepted):** A blue button labeled "CHOOSE FILE".
- SUBMIT:** A blue button at the bottom center of the form.

# Manage Models (Admin)

- List of existing models
- Delete a model



Name	Date	Action
dnn_v1	12/3/2024, 8:45:08 PM	
dnn_v3	12/4/2024, 12:10:28 PM	
dnn_v2	12/4/2024, 12:12:04 PM	



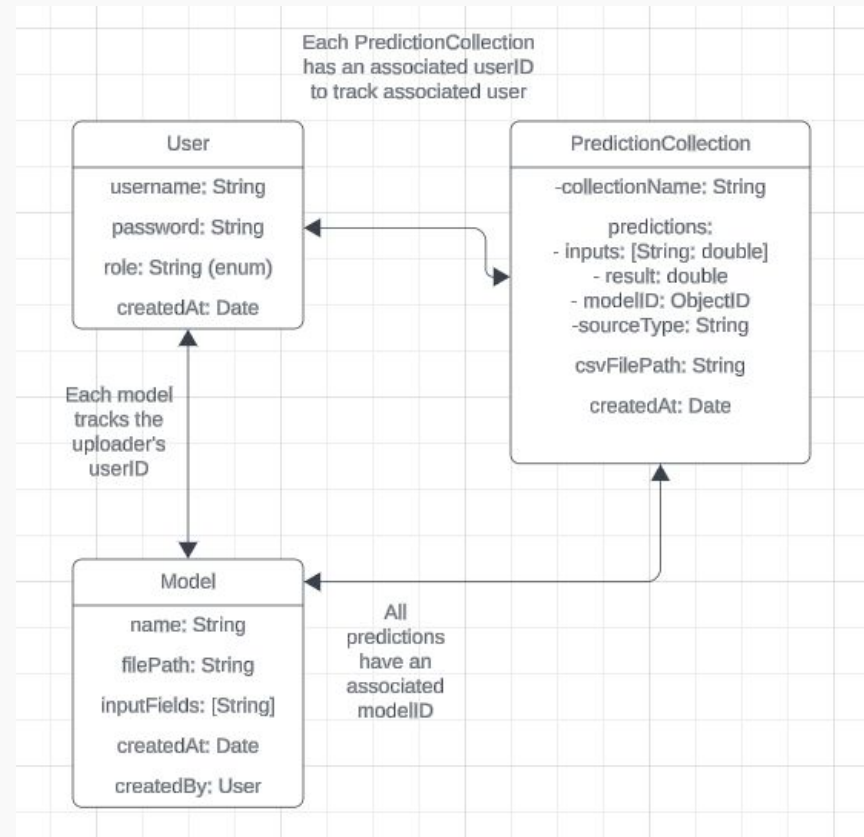
# MongoDB

## 1. Backend Database

- a. Communicates with Next JS

## 2. Schemas

- a. User Schema
- b. Model Schema
- c. Prediction Collection Schema



# NextJS

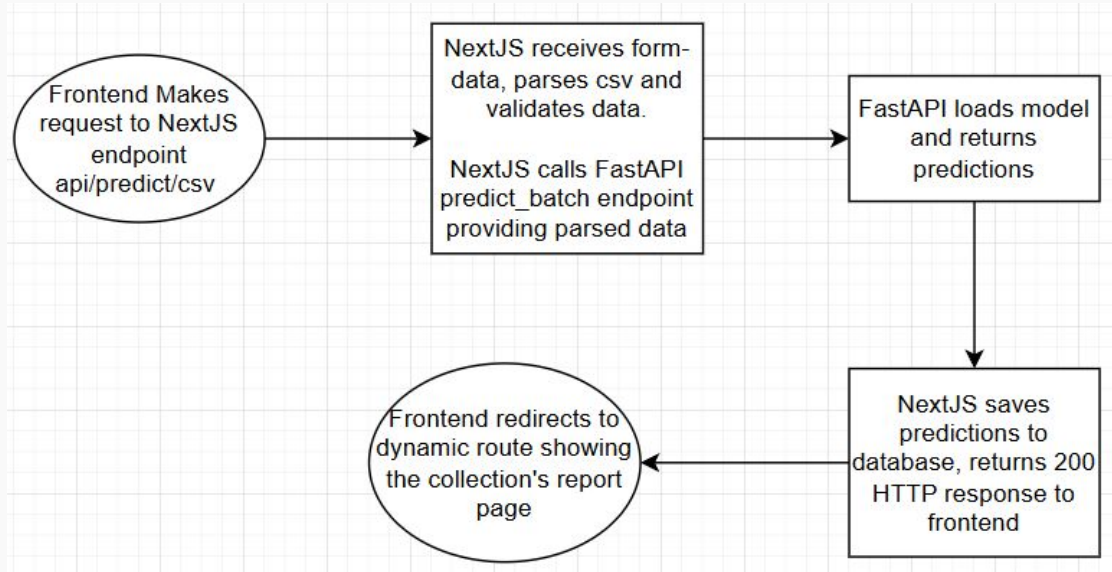
1. Web Frontend
2. Web Backend API
  - a. Communicates with FastAPI & MongoDB
  - b. User Login
  - c. User Roles
  - d. User Prediction Collections
    - i. CRUD operations
    - ii. Add predictions using FastAPI ML Backend
      1. Manual Input
      2. CSV Input
  - e. ML Model management (Admin functionality)
    - i. Upload/delete Models



# FastAPI

## 1. Backend ML Environment

- a. Runs ML Models upon request from NextJS, generates predictions
- b. Prediction API endpoints:
  - i. Single
  - ii. Batch
- c. Frameworks
  - i. NumPy
  - ii. Pandas
  - iii. Tensorflow



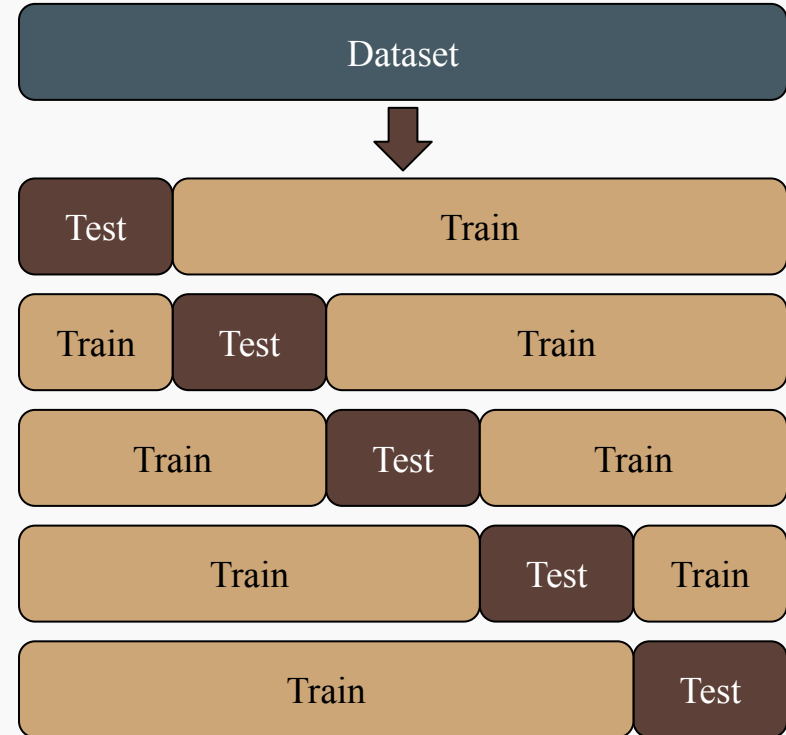


# Machine Learning Model Development



# ML Model Development: Training & Testing Split

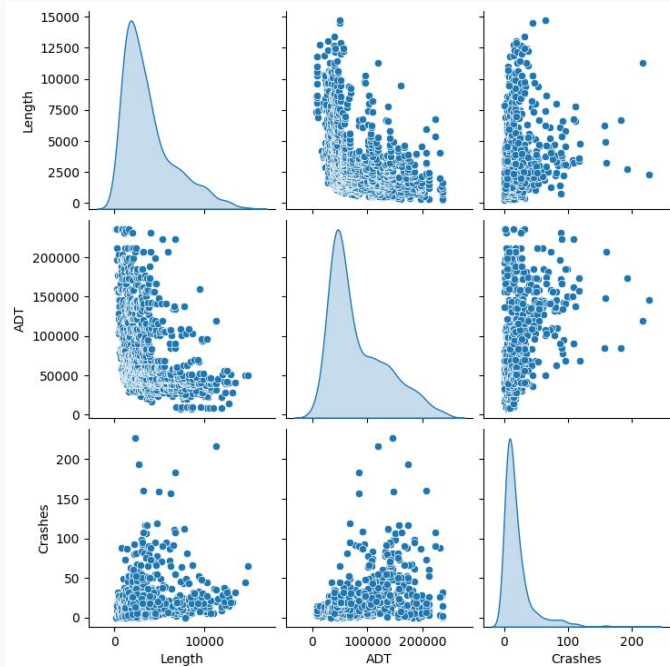
1. Data Dictionary
  - a. Length: Length of road
  - b. ADT: Average daily traffic
  - c. Crashes: Number of crashes
2. Training and testing data split
  - a. *80/20 training/test split using k-fold cross validation (k=5) [3]*



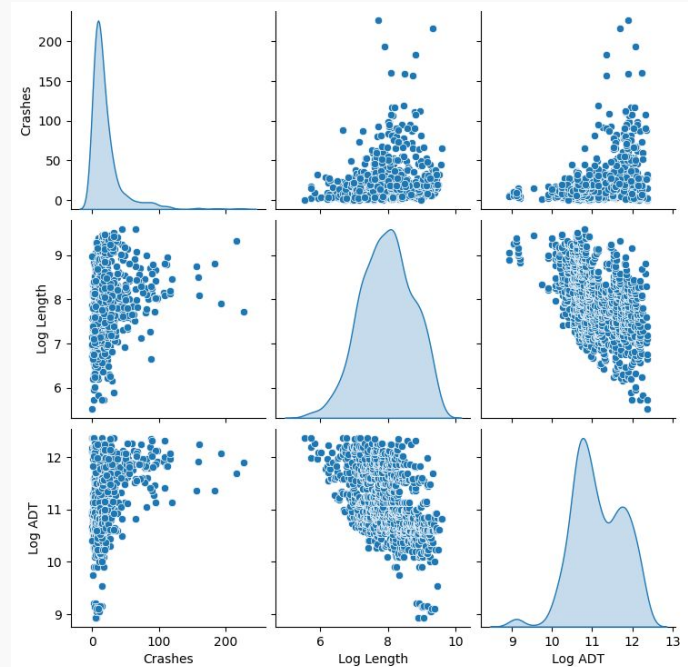
*k-fold cross validation (k=5) [3]*

# ML Model Development: Exploratory Data Analysis

Original Training Data  
(n=934)



Log Transformed Training Data  
(n=934)



# Exploratory Data Analysis (Travis)

Original Training Data

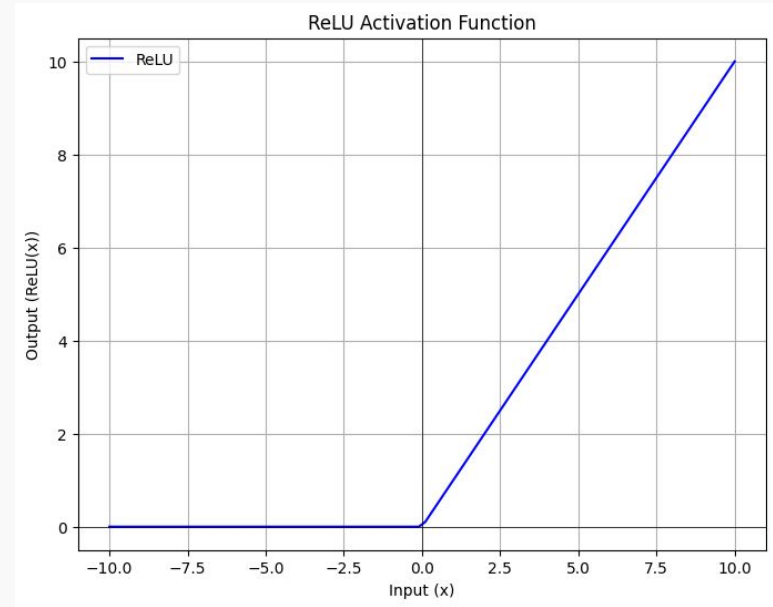
	<b>Length</b>	<b>ADT</b>	<b>Crashes</b>
<b>Length</b>	1.000	-0.423	0.136
<b>ADT</b>	-0.423	1.000	0.399
<b>Crashes</b>	0.136	0.399	1.000

Log Transformed  
Training Data

	<b>Log Length</b>	<b>Log ADT</b>	<b>Crashes</b>
<b>Log Length</b>	1.000	-0.511	0.171
<b>Log ADT</b>	-0.511	1.000	0.400
<b>Crashes</b>	0.171	0.400	1.000

# ML Model Development: Regression with DNN

1. Deep neural network layers
  - a. Normalization layer to scale data for model stability [4]
  - b. Two 64 unit layers utilizing Rectified Linear Unit (ReLU) activation function
    - i.  $f(x) = \max(x, 0)$
    - ii. Output range:  $[0, +\infty)$
  - c. One 1 unit layer *ReLU activation layer for positive output* [5]



ReLU Activation Function [5]



# ML Model Development: Training & Evaluation

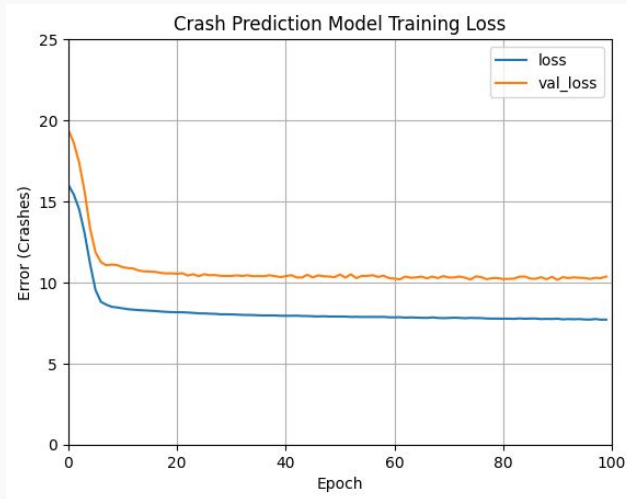
1. Mean Absolute Error (MAE) used as loss function given *robustness to outlier* [6]
2. MAE and Mean Squared Error (MSE) employed as evaluation metrics

	<b>MSE</b>	<b>MAE</b>
<b>DNN model 1</b>	25.455	3.638
<b>DNN model 2</b>	39.146	4.084
<b>DNN model 3</b>	86.305	5.524
<b>DNN model 4</b>	241.368	8.792
<b>DNN model 5</b>	1886.828	26.891

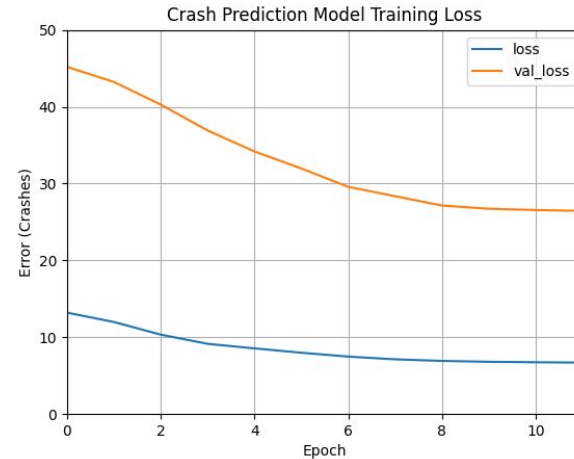
Model Evaluation Table

# ML Model Development: Model Training

1. Train model on train features and train labels
2. 80/20 training/validation split
3. *Train for 12 epochs (where val loss first reaches a minimum)* [7]

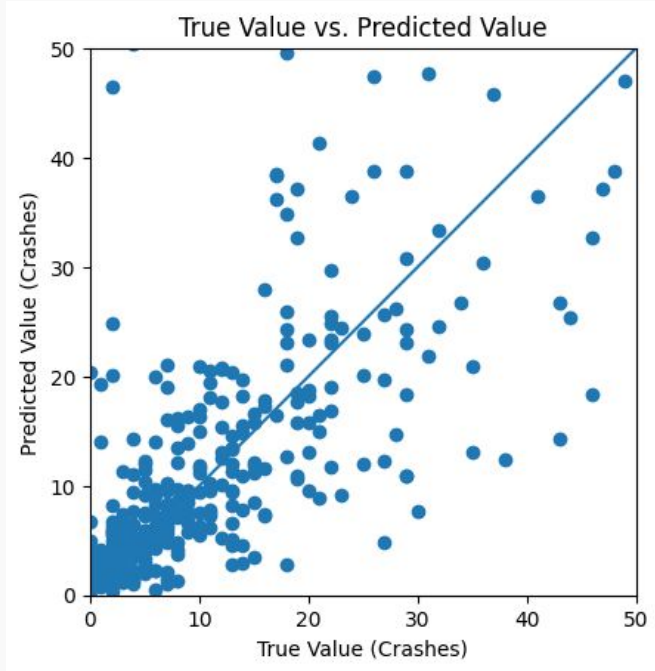


Version 2 Model

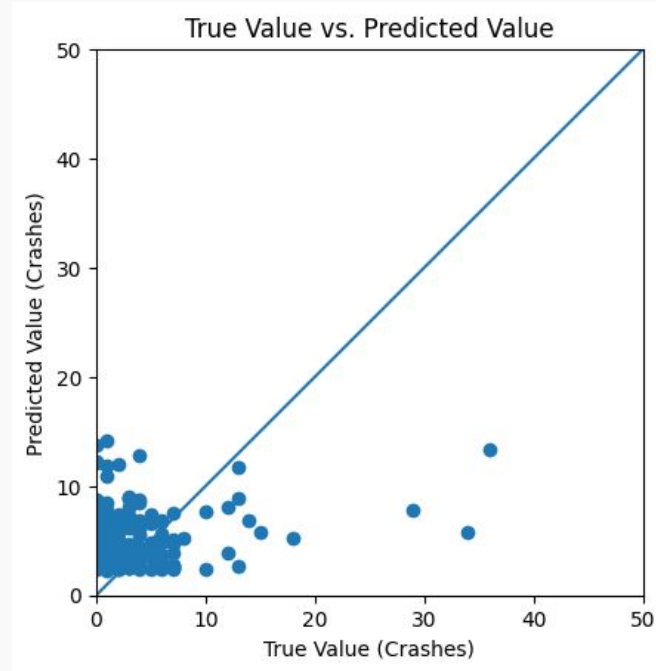


Version 3 Model

# ML Model Development: Performance

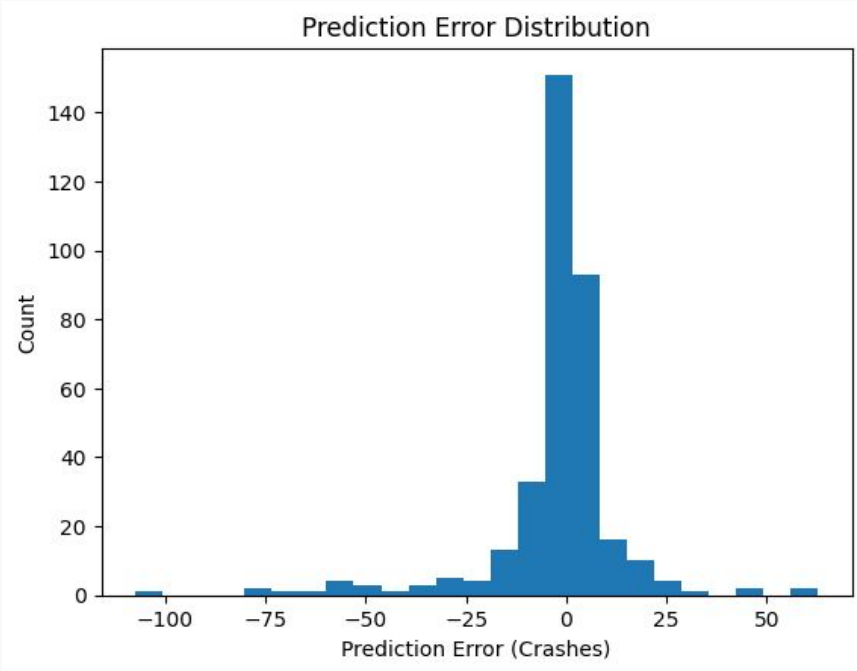


Version 2 Model

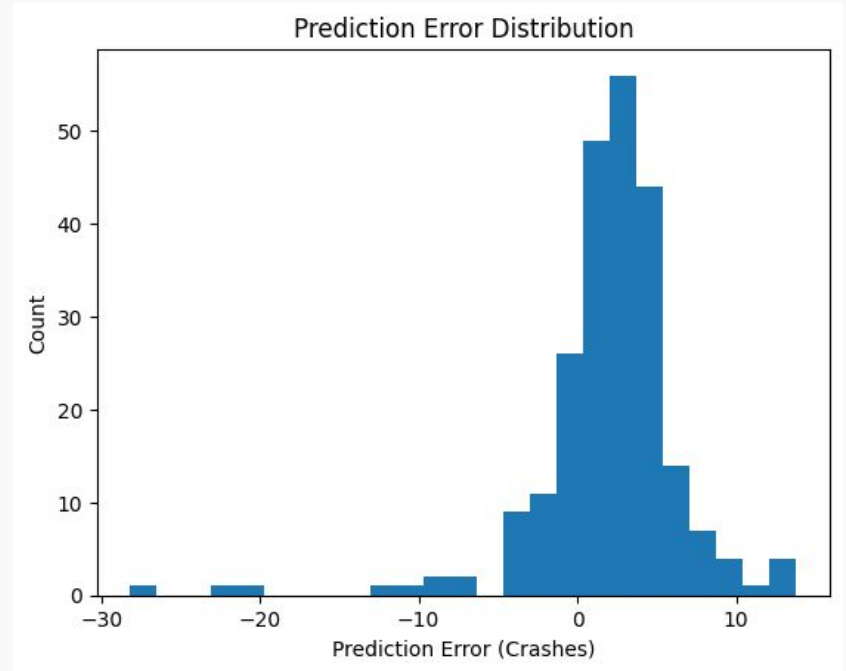


Version 3 Model

# ML Model Development: Performance



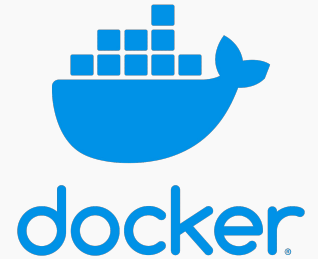
Version 2 Model



Version 3 Model

# Project Deployment Challenges

1. Bare-Metal Deployment: Compatibility Issue
  - a. Next.js requires GNU C Library (glibc) 2.18 and above
  - b. Current server only has 2.17
2. Deployment with Docker instead
  - a. Dockerfile & Compose file
  - b. Multi-stage image building
  - c. Volume binding for data persistence
  - d. Creates default users on initial startup
3. Docker Issues Encountered
  - a. Compatibility issues with Alpine Linux Image and Tensorflow
    - i. Use Debian-Slim Image instead
    - ii. Drawback: Larger image size



# Deliverables

1. Web application for model interaction and collection management
2. Machine learning model for crash rate prediction
3. Secure web application with user authentication

# Future plans for the project

1. More admin features
2. Accessibility features
3. Accept more model types in addition to .keras and .h5
4. ML model improvements
5. Bug fixes

# Lessons Learned

1. Team communication is key to ensure timely execution of tasks
2. Client work is about adaptability
3. Initial planning is crucial to an effective workflow



# Acknowledgements

Thank you, Dr. Mohamed Farag, for supporting the development of this project.



# References

URLs to follow for additional information:

- [1] [Project Canvas Page by Group 11](#)
- [2] [Jupyter Notebook starter code provided by client](#)
- [3] [Cross Validation in Machine Learning by GeeksforGeeks updated August 7, 2024](#)
- [4] [Data Normalization Machine Learning by GeeksforGeeks updated November 4, 2024](#)
- [5] [ReLU Activation Function in Deep Learning by GeeksforGeeks updated September 30, 2024](#)
- [6] [“What is a loss function?” by IBM, 12 July 2024](#)
- [7] [Epoch in Machine Learning by GeeksforGeeks updated November 2, 2024](#)

Thank You

