



Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

ECEN 404 Bi-Weekly Update #5

Team 57: Deep Learning for Hydroponic Soybean Growth

Team members: Samuel He, Mary Hughes
Sponsor: Sambandh Dahl, Krishna Gadepally

Project Summary

- **Problem Statement:**

Researchers take time to track the solution and day of growth of a hydroponically grown plant

- **Solution**

Deep learning model and user interface that tracks Day of Growth and outputs other growth data that may be useful to the user

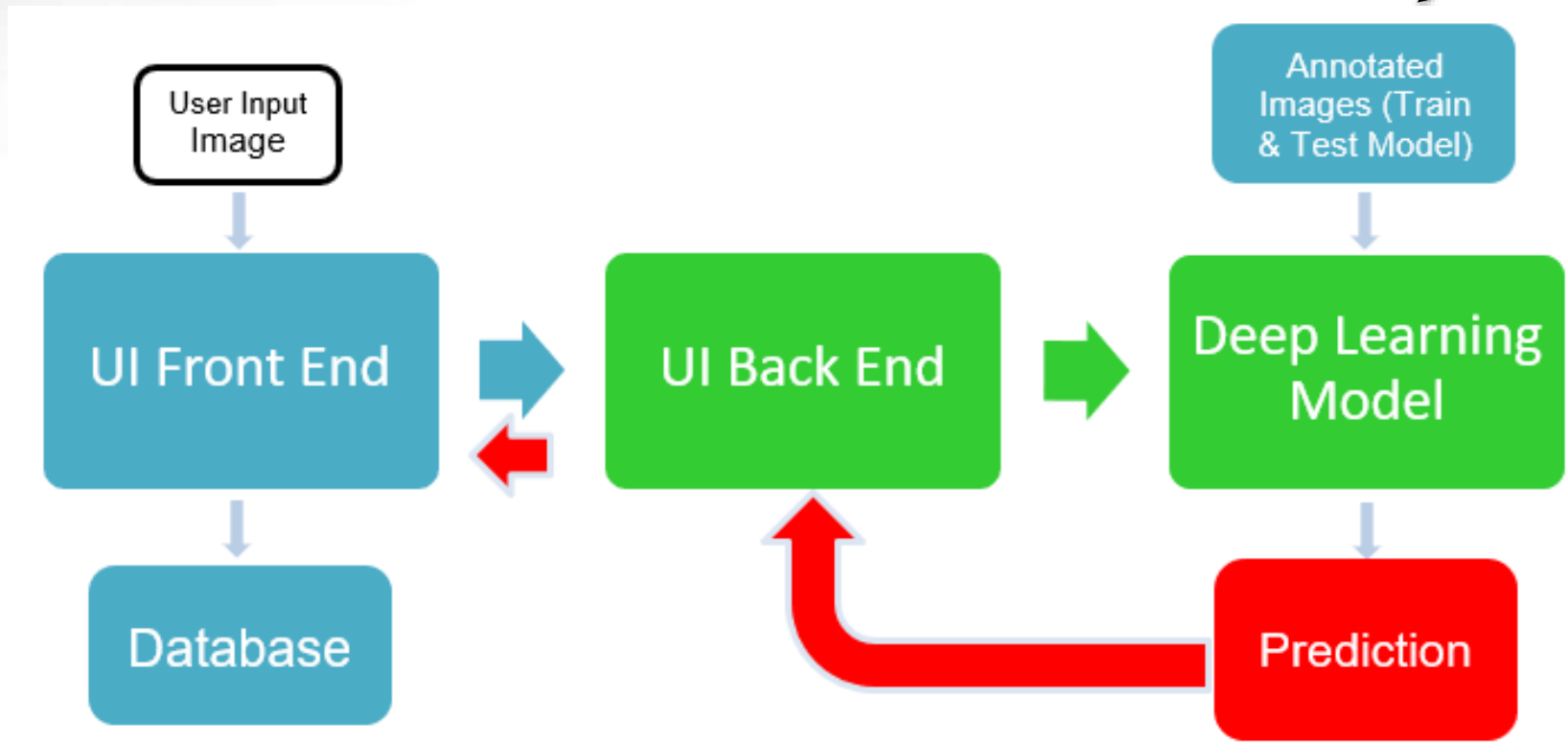


Image 1. Sample Data Image

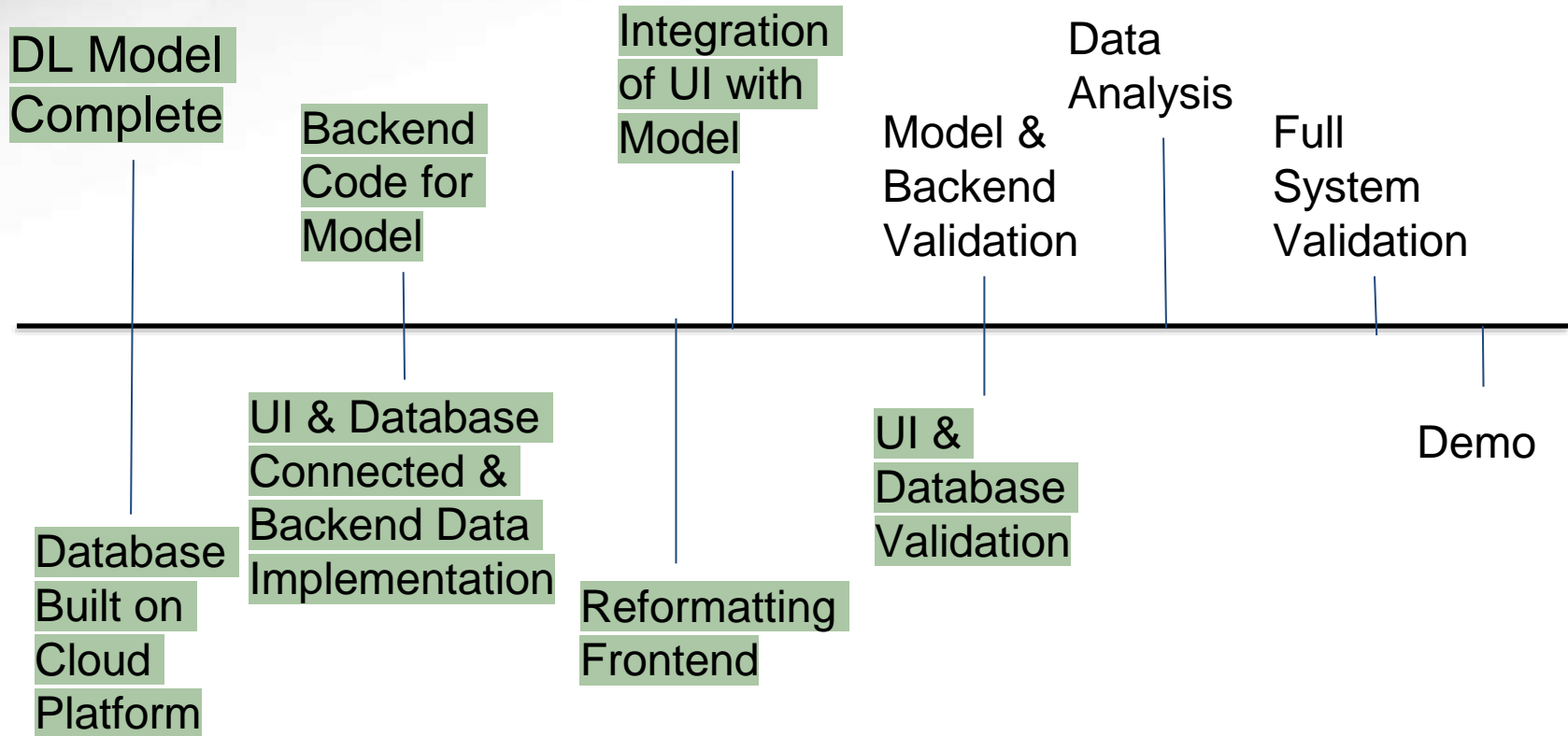
System Diagram



.JPG Data

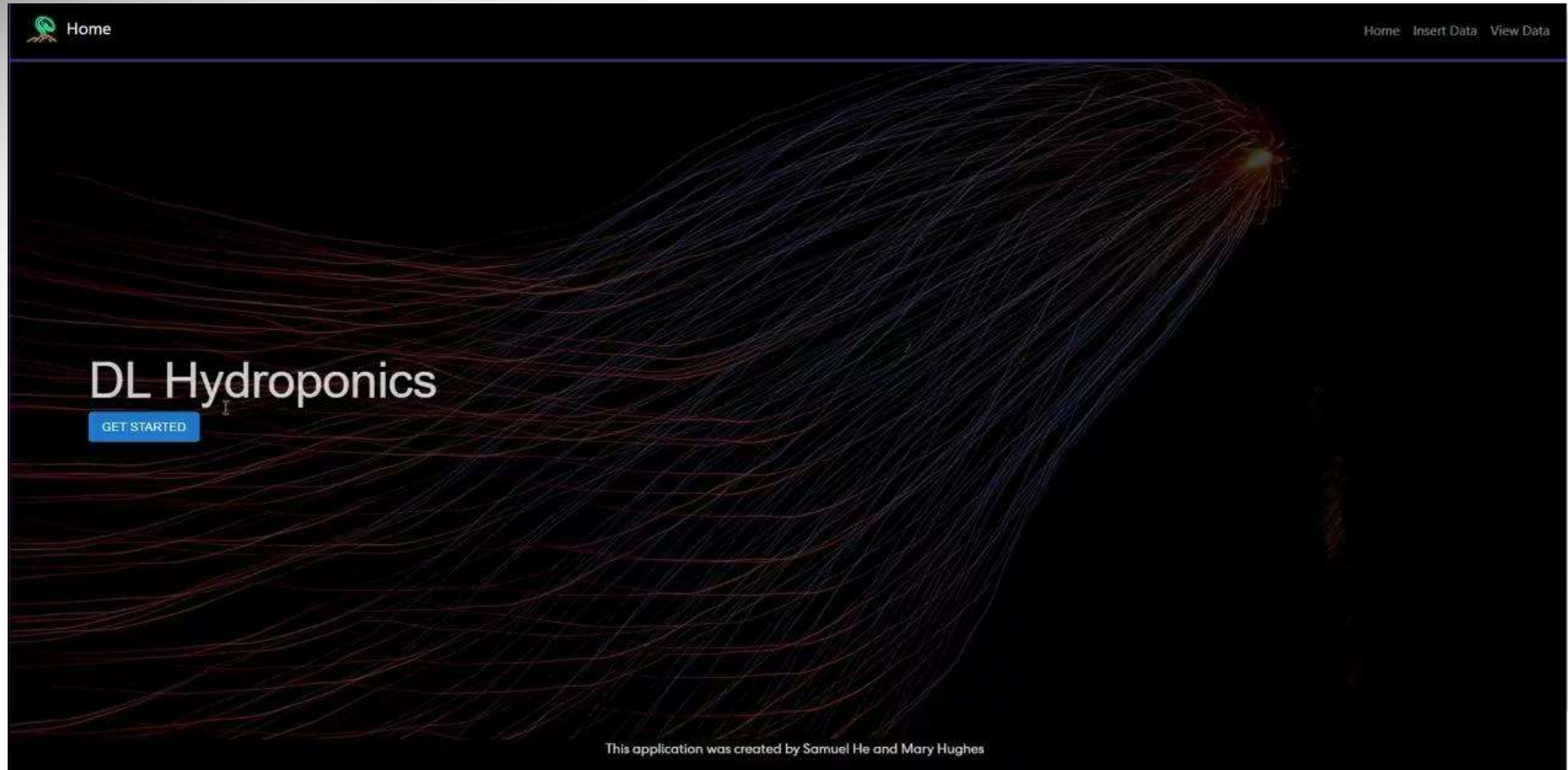


Project Timeline





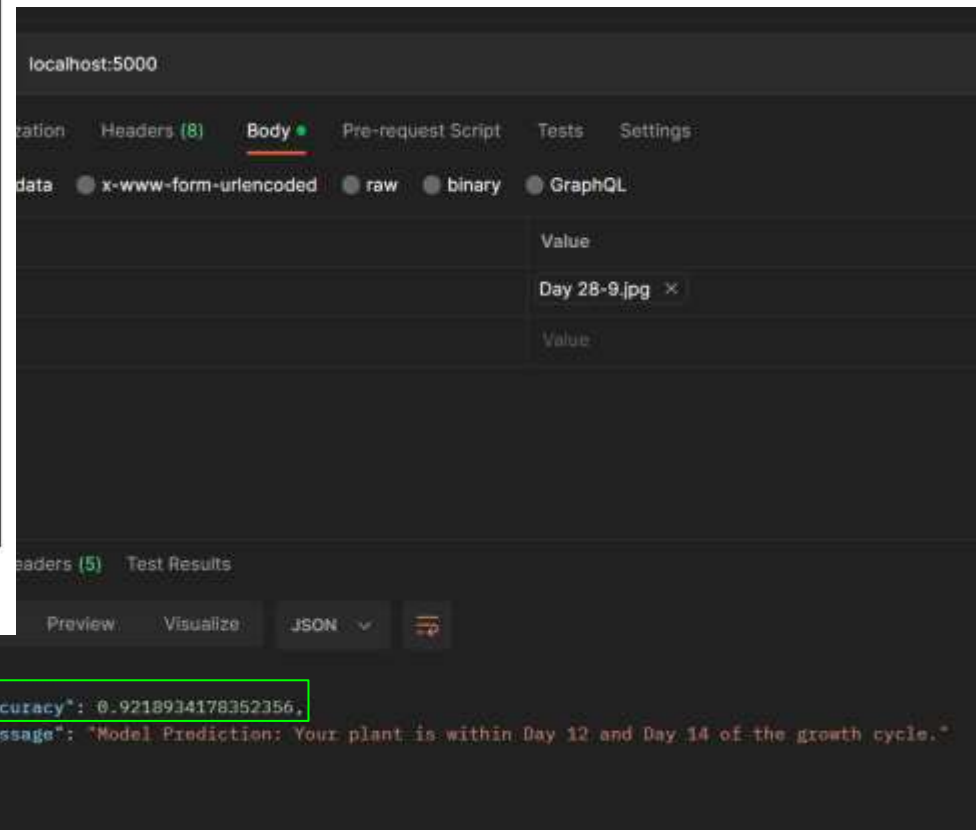
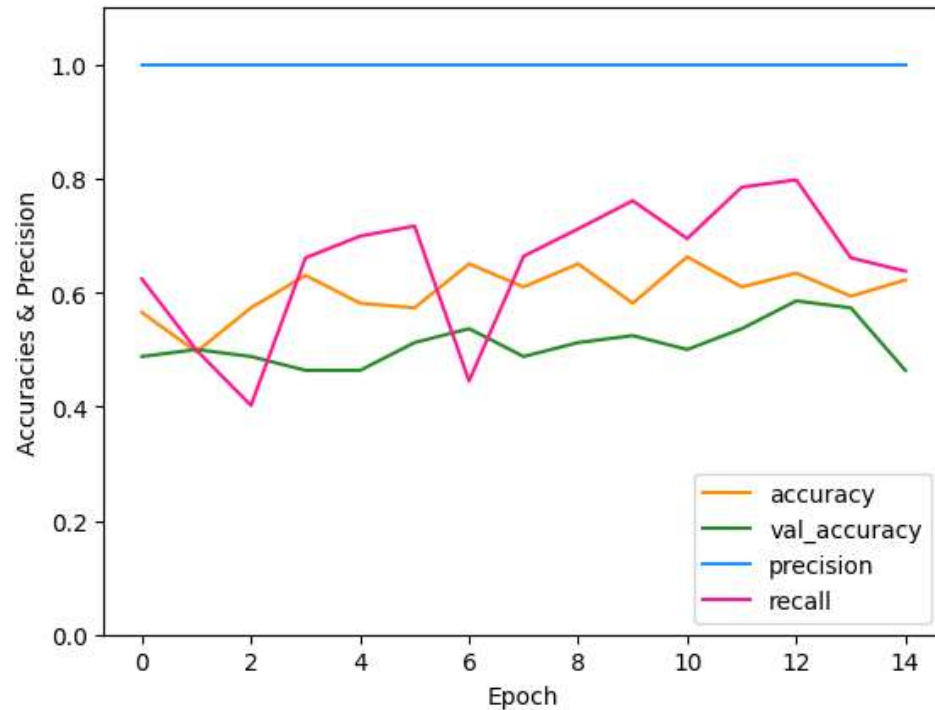
Full System Video



Subsystem: Deep Learning Model

| Accomplishments since 403 15 hours of effort | Ongoing progress/problems and plans until the next presentation |
|---|--|
| <ul style="list-style-type: none"> ● Implemented additional metrics to evaluate the model <ul style="list-style-type: none"> ○ Precision, Recall, F1 Score ○ included creating classes for each since they required custom metrics ● Rerun model training multiple times ● Changed output format to improve accuracy <ul style="list-style-type: none"> ○ Model favors a particular range of days regardless of input | <ul style="list-style-type: none"> ● More of the same, just running until we get the most ideal output ● Background data analysis implementation |

Subsystem: DL Model



"accuracy": 0.9218934178352356,

Subsystem: User Interface

| Accomplishments since last update 15 hours of effort | Ongoing progress/problems and plans until the next presentation |
|---|---|
| <p>React Front-end:</p> <ul style="list-style-type: none">- Fixed CSS Styling Grid layout issues- Updated Modal views <p>Still Validating UI/Full System - Inputting given Test data/Fixing Bugs</p> <ul style="list-style-type: none">- Made website more responsive to user errors (i.e. wrong file format, too much data, etc.) | <p>Finish Validating Full System with Updated Model</p> |



Execution Plan

| | 1/30/2023 | 2/6/2023 | 2/13/2023 | 2/20/2023 | 2/27/2023 | 3/6/2023 | 3/20/2023 | 3/27/2023 | 4/3/2023 | 4/10/2023 | 4/17/2023 | 4/26/2023 |
|-----------------------------------|-----------|----------|-----------|-----------|-----------|----------|-----------|-----------|----------|-----------|-----------|-----------|
| Create Database on Cloud Platform | | | | | | | | | | | | |
| Link Backend with Database | | | | | | | | | | | | |
| Display Database Info on UI | | | | | | | | | | | | |
| Deploy Frontend | | | | | | | | | | | | |
| Reformat Frontend | | | | | | | | | | | | |
| UI Autoscaling | | | | | | | | | | | | |
| Validate Frontend | | | | | | | | | | | | |
| Format Dataset for Use in Model | | | | | | | | | | | | |
| Finish DL Model | | | | | | | | | | | | |
| Deploy Model in S3 | | | | | | | | | | | | |
| Build AWS Lambda Function | | | | | | | | | | | | |
| Data Analysis in Backend | | | | | | | | | | | | |
| Debugging backend & model | | | | | | | | | | | | |
| Validate Model Accuracy | | | | | | | | | | | | |
| Validate Data Analysis | | | | | | | | | | | | |
| Integrate Frontend & Model | | | | | | | | | | | | |
| Validate Integrated System | | | | | | | | | | | | |
| Update Presentations | | | | | | | | | | | | |
| Final Demo | | | | | | | | | | | | |
| Engineering Project Showcase | | | | | | | | | | | | |

Samuel He
Mary Hughes
Shared Goals

Complete
In Progress
Not Yet Started
Behind Schedule



Validation plan

| Paragraph # | Test Name | Success Criteria |
|-------------|---|--|
| 3.2.1.3 | UI Image Input | Users can upload up to 10 different images of any size with a 10 second maximum response time. |
| 3.2.1.3 | Webpage Autoscaling | Webpage autoscales properly to mobile and desktop screens |
| 3.2.1.3 | Webpage Interactivity | Webpage navigation interactions are functional |
| 3.2.1.3 | Database Outputs on Frontend | Webpage frontend has all database prediction information displayed properly |
| 3.2.1.3 | Input Delivery to Back End | Image is successfully being delivered to the backend from the front end of the UI in <1sec |
| 3.2.5.1.1 | Application Failure Detection | Internal testing properly identifies when the application fails to communicate with the deep learning model. |
| 3.2.5.1.1 | Application Failure Response | Webpage gives user a correct error message when incorrect image formats are uploaded |
| 3.2.5.1.1.1 | Model Failure Detection | Application correctly detects if the model has given a valid input to the UI. |
| 3.2.1.1 | Day of Growth Identification | The deep learning model is correctly identifying the day of growth of an input. |
| 3.2.1.2 | Nutrient Solution Detection | The deep learning model is correctly identifying the nutrient solution of an input. |
| 3.2.1.3 | UI Delivers Input to backend with API Calls | User Input images are successfully delivered to AWS using the APIs built in API Gateway. |
| 3.2.3.2.1 | UI Output Delivery | An output is being delivered to the UI in the correct format, including the prediction and the accuracy of prediction. |
| N/A | Full System Demo | The application and deep learning model process input as expected and deliver correct output to the UI. |
| 3.2.1.3 | UI Backend Communication with Model | The User Interface Back End API calls work as expected, and can return a prediction in a 3rd party testing platform. |
| 3.2.1.3 | UI Readability | UI design is clean and understandable, easy to use on multiple brightness levels |

Validation plan

| Methodology | Status | Responsible Engineers |
|---|-------------|-----------------------|
| Upload 20 different image sets to the User interface, each with varying sizes and quantities ranging from 1-20 photos and 1-10 MB and time the response times. | UNTESTED | Samuel He |
| Test the mobile view of the website on at least 10 different mobile views, using React Native Layout Tester. Compare results. | UNTESTED | Samuel He |
| Test button pressing functionalities of each button on navigation. | PASSED | Samuel He |
| Upload 50 images and monitor predictions for them both individually and altogether. Input images into model directly. Compare results. | PASSED | Samuel He |
| Monitor database to see if corresponding images and predictions are sent out and received. Send out time and retrieval time will be monitored by test cases in React. | IN PROGRESS | Samuel He |
| Restrict access from the application to the model. Attempt to upload an image to the model. | IN PROGRESS | Samuel He |
| Upload a set of 15 different files that are not .jpg or .jpeg. | PASSED | Samuel He |
| Create an invalid prediction response on the backend, and attempt to upload an image to the model. | PASSED | Samuel He |
| Create 264 test cases with corresponding images that cover all of the different categories. Compare results with pre-determined day of growth inputs. | IN PROGRESS | Mary Hughes |
| Create 66 test cases in Python with corresponding images that cover all of the different categories. Compare results with pre-determined nutrient solution inputs. | UNTESTED | Mary Hughes |
| Test the API using POSTMAN. Verify with VS Code terminal that the API was used. | PASSED | Mary Hughes |
| Upload 20 different images to POSTMAN, one from each day of the growth cycle represented in the dataset, and verify the output shown in the POSTMAN console is the correct format for the UI to receive and interpret properly. | PASSED | Mary Hughes |
| Upload a set of 20 images, and compare their individual predictions with the model to the UI output. | UNTESTED | Shared |
| First, validate the Frontend communication with the Backend. Secondly, send 30 images to the model using both Postman and the UI. Compare response results. | UNTESTED | Shared |
| Compare readability on at least 5 different monitor display/brightness settings. | UNTESTED | Shared |



Dwight Look College of

ENGINEERING
TEXAS A&M UNIVERSITY

Thanks & Gig 'Em!