Sri Lanka Institute of Information Technology

"SeamSense" – Real-time Quality Monitoring System for the Apparel Industry

R24-066

Samaraweera G.P.M.D.

IT21016066

Data Science Specialization

Table of Contents

Table of Figures

Project Component and Current Status

Component: Fusion Modeling for Worker-Centric Defect Analysis: Integrating Traditional and Time-Series Approaches

The project has made significant strides in developing a fusion model that integrates traditional time-series analysis with worker-centric demographic data. The time series model effectively forecasts future defect rates by analyzing historical data, while the demographic model enhances accuracy by identifying correlations between worker characteristics and defect occurrences. These models have been successfully combined into a robust fusion model, trained and validated for reliability. Additionally, a continuous feedback loop has been established to refine the model, ensuring it adapts to the dynamic conditions in the manufacturing environment.

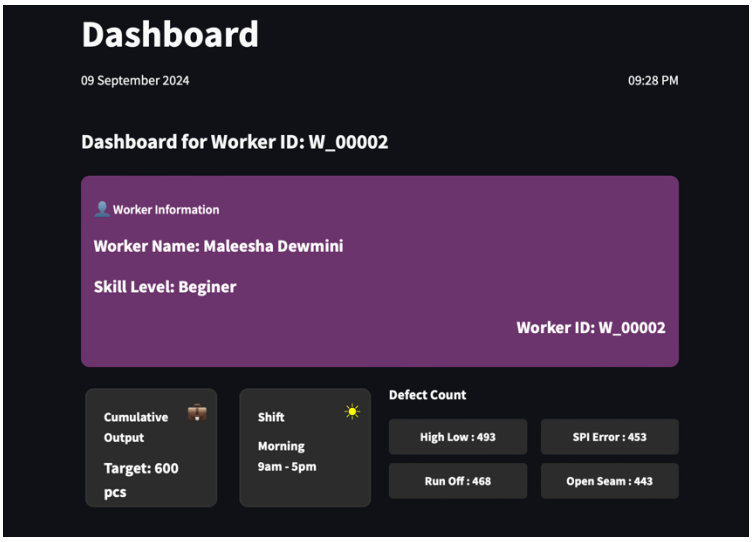
Progress

The system mainly composed of 2 objectives:

1. **Time Series Model Development: Successfully constructed a time series model using ARIMA, which analyzes historical defect data to forecast future defect rates based on identified temporal patterns and trends.**

```python
def train_arima_model(data, order=(3, 0, 3), steps=5):
    """
    Train an ARIMA model for a given time series data.
    """
    try:
        model = ARIMA(data, order=order)
        model_fit = model.fit()
        forecast = model_fit.forecast(steps=steps)
        logging.info(f"ARIMA model trained successfully with order {order}")
        return model_fit, forecast
    except Exception as e:
        logging.error(f"Error in training ARIMA model: {e}")
        return None, None
```

Figure 1 - Each worker time-series-analysis .py


Dashboard
09 September 2024
09:28 PM
Dashboard for Worker ID: W_00002
Worker Information
Worker Name: Maleesha Dewmini
Skill Level: Beginer
Worker ID: W_00002
Cumulative Output
Target: 600 pcs
Shift
Morning
9am - 5pm
Defect Count
High Low : 493
SPI Error : 453
Run Off : 468
Open Seam : 443

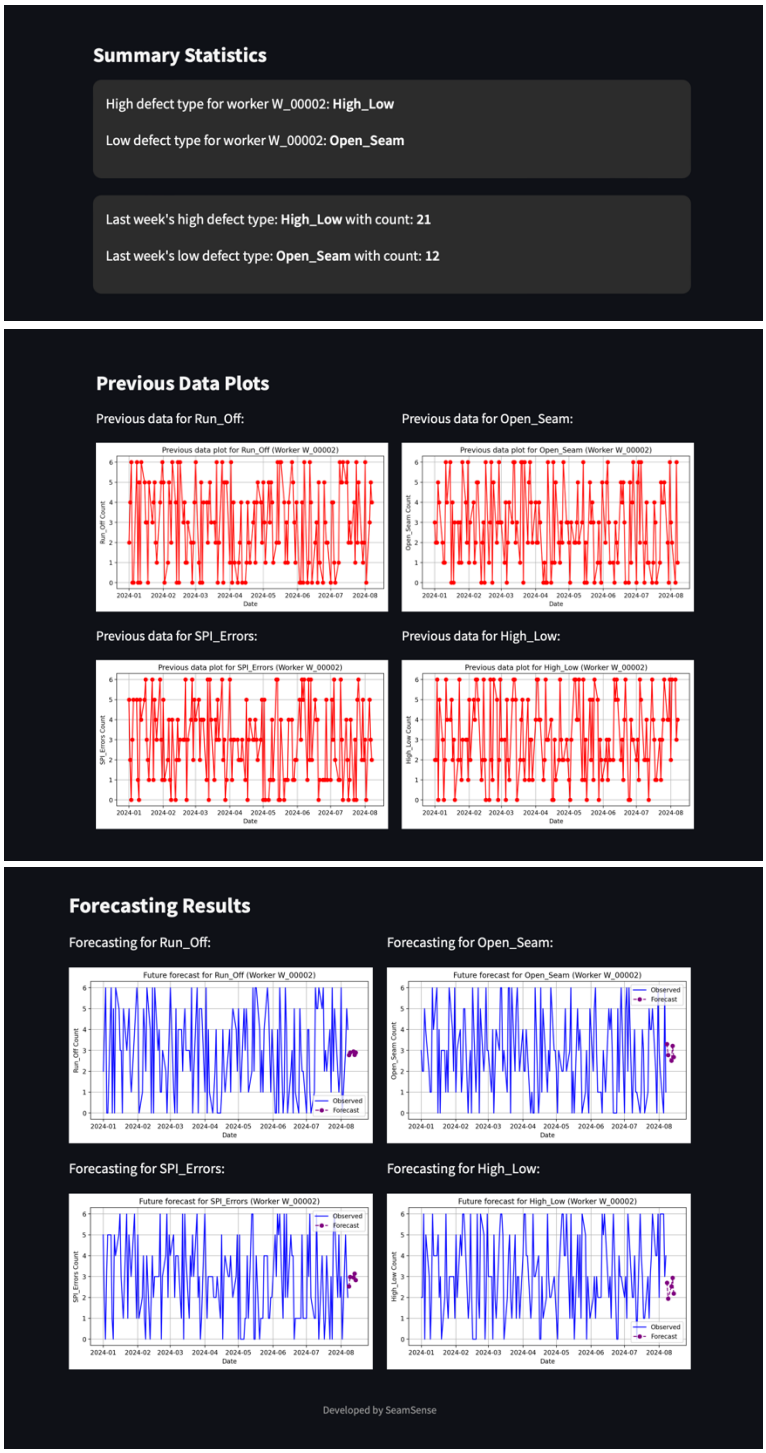


Figure 2 - UI for each worker forecasting

```python
def train_arima_model(data, order=(1, 1, 1), steps=5):
    """Train an ARIMA model and forecast future data."""
    model = ARIMA(data, order=order)
    model_fit = model.fit()
    forecast = model_fit.forecast(steps=steps)
    return model_fit, forecast
```

Figure 3 - all workers time-series-analysis .py

2. **Demographic Model Construction: Developed a demographic model that integrates worker characteristics and demographic data to identify correlations with defect occurrences, enhancing the overall prediction accuracy.**

```python
# Separate features and target variables
X = combined_data.drop(columns=['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low', 'defect_count', 'count', 'Worker_ID', 'Date'])
y = combined_data[['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low']]

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Define models and hyperparameter grids
models = {
    'RandomForest': {
        'model': MultiOutputRegressor(RandomForestRegressor(random_state=42)),
        'param_dist': {
            'regressor__estimator__n_estimators': randint(50, 200),
            'regressor__estimator__max_depth': randint(5, 20),
            'regressor__estimator__min_samples_split': randint(2, 10),
            'regressor__estimator__min_samples_leaf': randint(1, 10),
        }
    },
    'GradientBoosting': {
        'model': MultiOutputRegressor(GradientBoostingRegressor(random_state=42)),
        'param_dist': {
            'regressor__estimator__n_estimators': randint(50, 200),
            'regressor__estimator__learning_rate': uniform(0.01, 0.1),
            'regressor__estimator__max_depth': randint(3, 15),
            'regressor__estimator__min_samples_split': randint(2, 10),
            'regressor__estimator__min_samples_leaf': randint(1, 10),
        }
    },
    'SupportVector': {
        'model': MultiOutputRegressor(SVR()),
        'param_dist': {
            'regressor__estimator__C': uniform(0.1, 10),
            'regressor__estimator__epsilon': uniform(0.01, 0.1),
            'regressor__estimator__kernel': ['linear', 'poly', 'rbf'],
        }
    },
```

```
    'LinearRegression': {
        'model': MultiOutputRegressor(LinearRegression()),
        'param_dist': {}
    },
    'RidgeRegression': {
        'model': MultiOutputRegressor(Ridge(random_state=42)),
        'param_dist': {
            'regressor__estimator__alpha': uniform(0.1, 10)
        }
    },
    'LassoRegression': {
        'model': MultiOutputRegressor(Lasso(random_state=42)),
        'param_dist': {
            'regressor__estimator__alpha': uniform(0.1, 10)
        }
    }
}
```

Figure 4 - Demographic model .py

```
  demographic_data['Joining_Date'] = pd.to_datetime(demographic_data['Joining_Date'])
Model: RandomForest
  Best Score (MSE): 4.11106174832035
  Best Parameters: {'regressor__estimator__max_depth': 6, 'regressor__estimator__min_sampl
----------------------------------------------------------------
Model: GradientBoosting
  Best Score (MSE): 4.223969180805452
  Best Parameters: {'regressor__estimator__learning_rate': 0.025601864044243652, 'regresso
----------------------------------------------------------------
Model: SupportVector
  Best Score (MSE): 4.022989988760407
  Best Parameters: {'regressor__estimator__C': 2.2233911067827616, 'regressor__estimator__
----------------------------------------------------------------
Model: LinearRegression
  Best Score (MSE): 4.029561429271488
  Best Parameters: N/A
----------------------------------------------------------------
Model: RidgeRegression
  Best Score (MSE): 4.024396989425847
  Best Parameters: {'regressor__estimator__alpha': 9.60714306409916}
----------------------------------------------------------------
Model: LassoRegression
  Best Score (MSE): 4.014036666248644
  Best Parameters: {'regressor__estimator__alpha': 3.845401188473625}
----------------------------------------------------------------
Best model: LassoRegression with Mean CV MSE = 4.014036666248644
...
  - Mean Squared Error: 4.162211765146858
  - R² Score: -0.005724057334019372
  - Mean Absolute Error: 1.7601799904164834
----------------------------------------------------------------
```

Figure 5 - Select best model - Lasso Regression

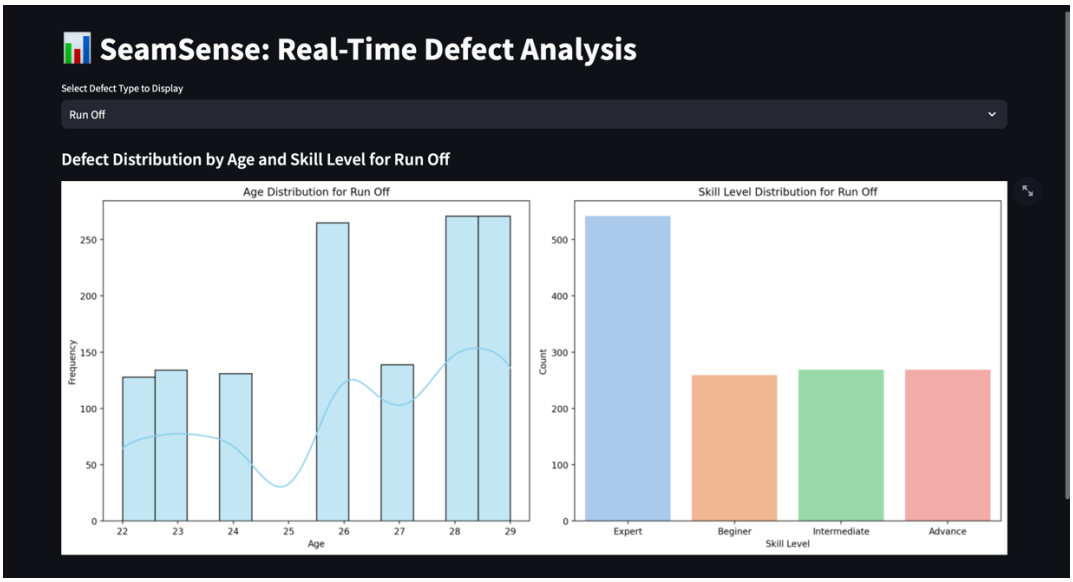


Figure 6 - Age and Skill level analysis for workers

3. **Fusion Model Integration: Combined the time series and demographic models into a unified fusion model. This model leverages both historical trends and worker-centric data for more robust defect predictions.**

```
# Prepare the fusion data
defect_types = ['Run_Off', 'Open_Seam', 'SPI_Errors', 'High_Low']
combined_forecasts = prepare_fusion_data(time_series_forecasts, traditional_model, combined_data, defect_types)

# Extract target values
y_actual = combined_data[defect_types]

# Align combined forecasts with y_actual
combined_forecasts = combined_forecasts.loc[y_actual.index]

# Train the fusion model using stacking with a meta-model
X_train, X_test, y_train, y_test = train_test_split(combined_forecasts, y_actual, test_size=0.2, random_state=42)

# Define base models and meta-model
base_models = [
    ('gb', MultiOutputRegressor(GradientBoostingRegressor(random_state=42))),
    ('rf', MultiOutputRegressor(RandomForestRegressor(random_state=42)))
]

# The meta-model should also be wrapped in MultiOutputRegressor to handle multi-output
meta_model = MultiOutputRegressor(GradientBoostingRegressor(random_state=42))

# Train the base models
base_model_predictions_train = np.zeros((X_train.shape[0], len(defect_types) * len(base_models)))
base_model_predictions_test = np.zeros((X_test.shape[0], len(defect_types) * len(base_models)))
for i, (name, model) in enumerate(base_models):
    model.fit(X_train, y_train)
    base_model_predictions_train[:, i*len(defect_types):(i+1)*len(defect_types)] = model.predict(X_train)
    base_model_predictions_test[:, i*len(defect_types):(i+1)*len(defect_types)] = model.predict(X_test)

# Combine the base model predictions with the original features
X_train_combined = np.hstack([base_model_predictions_train, X_train])
X_test_combined = np.hstack([base_model_predictions_test, X_test])

# Train the meta-model on the combined features
meta_model.fit(X_train_combined, y_train)
```

Figure 7 - Fusion Model using Stacking technique .py

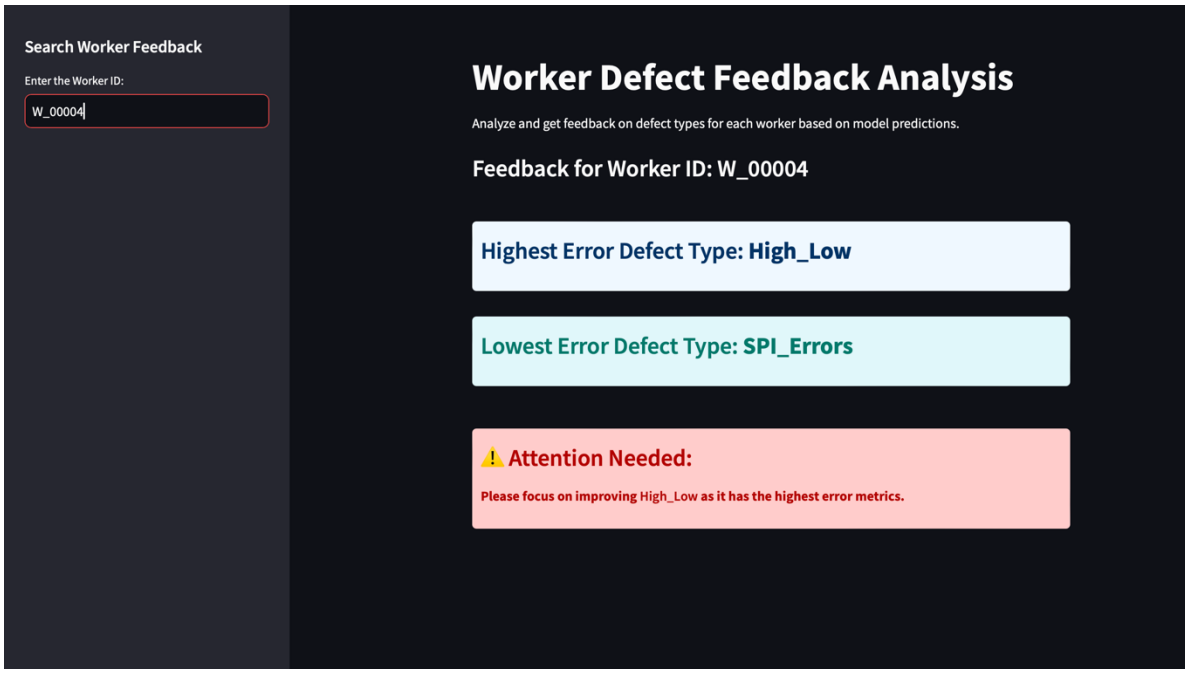


Figure 8 - UI for Fusion model output

Team Communication

The team opted for Microsoft Teams as its primary communication platform, creating a dedicated Team that included all four group members. They also utilized Zoom to stay in touch with supervisors, providing updates and receiving feedback on the project's progress. Regular team calls were scheduled for discussions, sharing knowledge, and planning.

To maintain constant communication with the supervisors, the team also used WhatsApp as an additional tool. This allowed for quick updates and coordination between the supervisor and co-supervisor, ensuring that everyone stayed informed and on the same page throughout the project.

Teams Channel

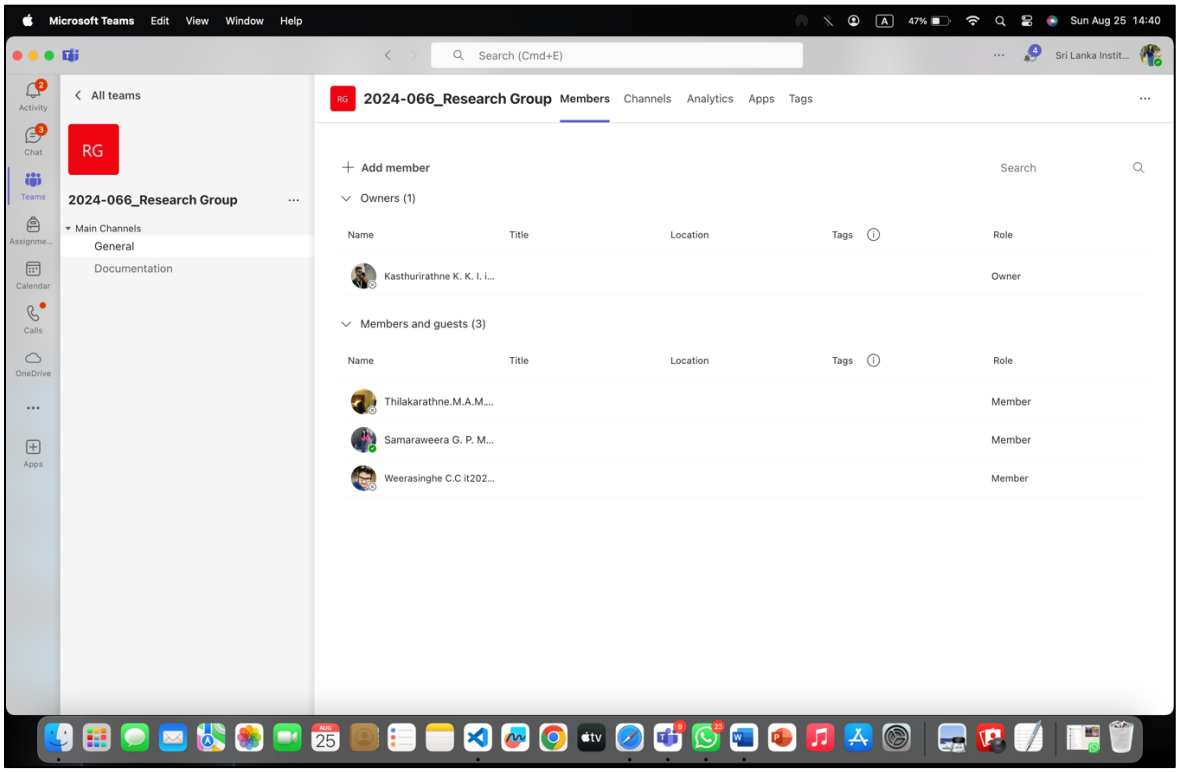


Figure 9 - Teams Channel Creation

Teams Calls with Research Team

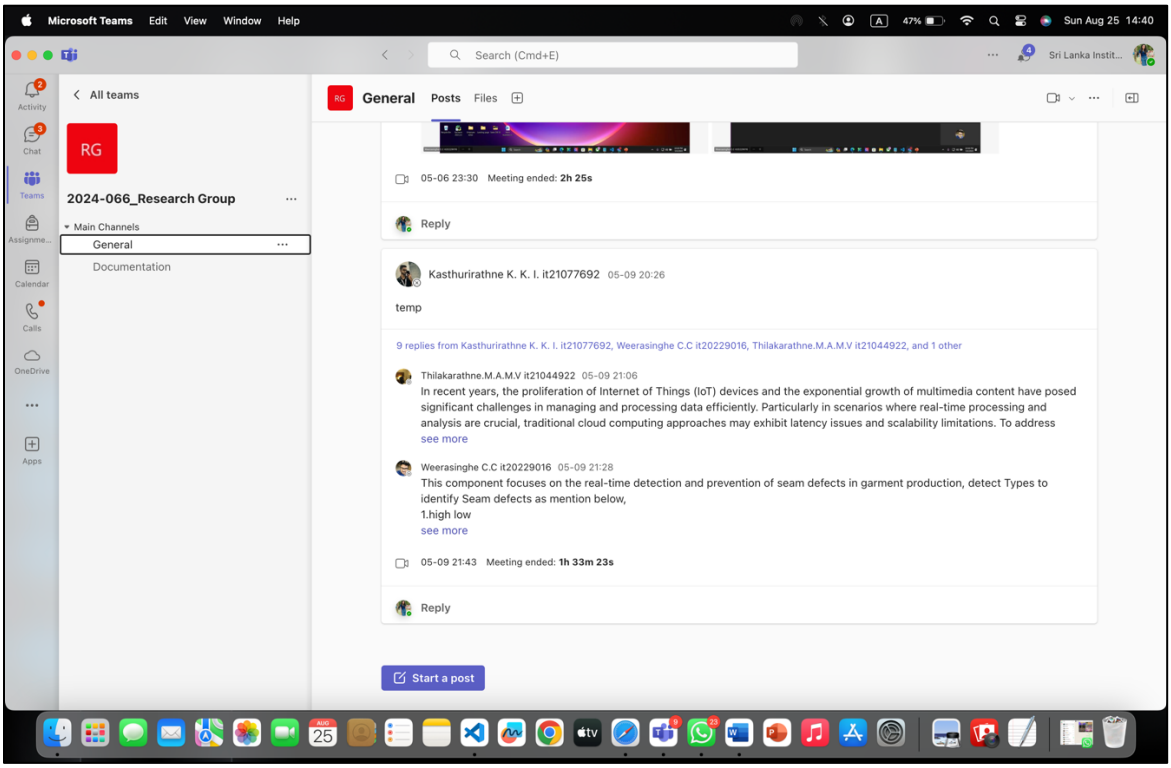


Figure 10 - Overview of Teams Calls and Communication

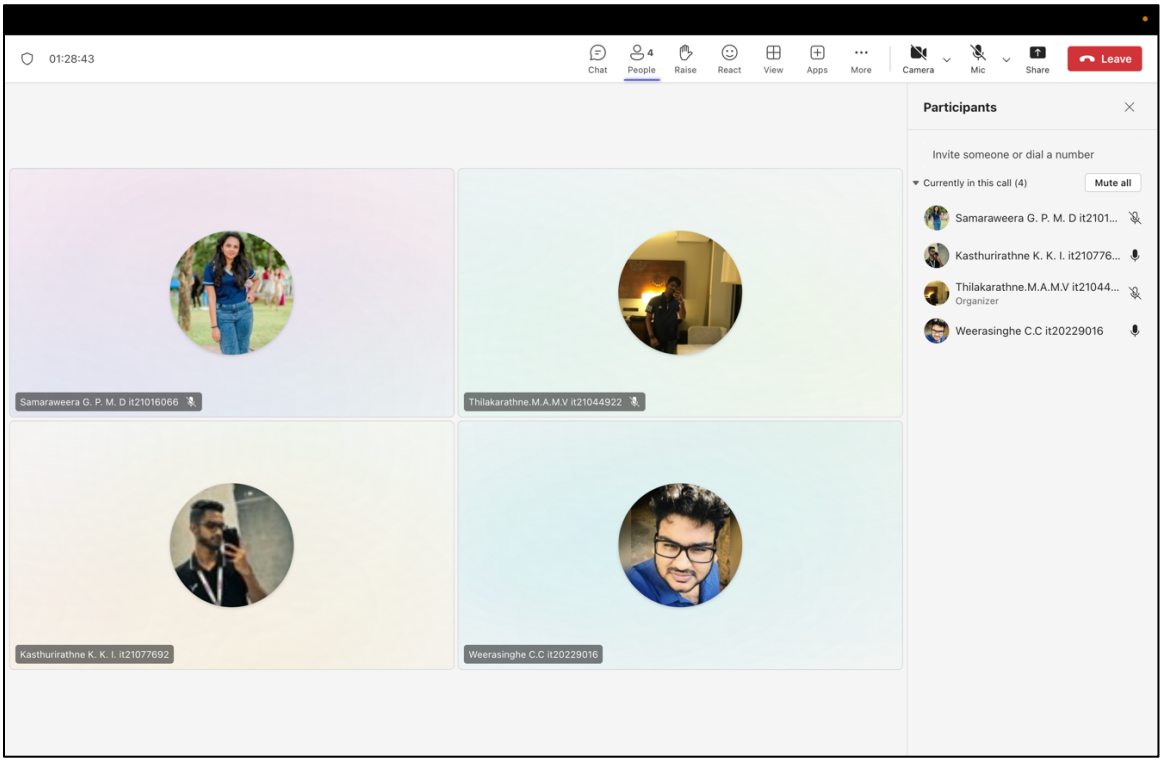


Figure 11 - Team Member Calls (Example 01)

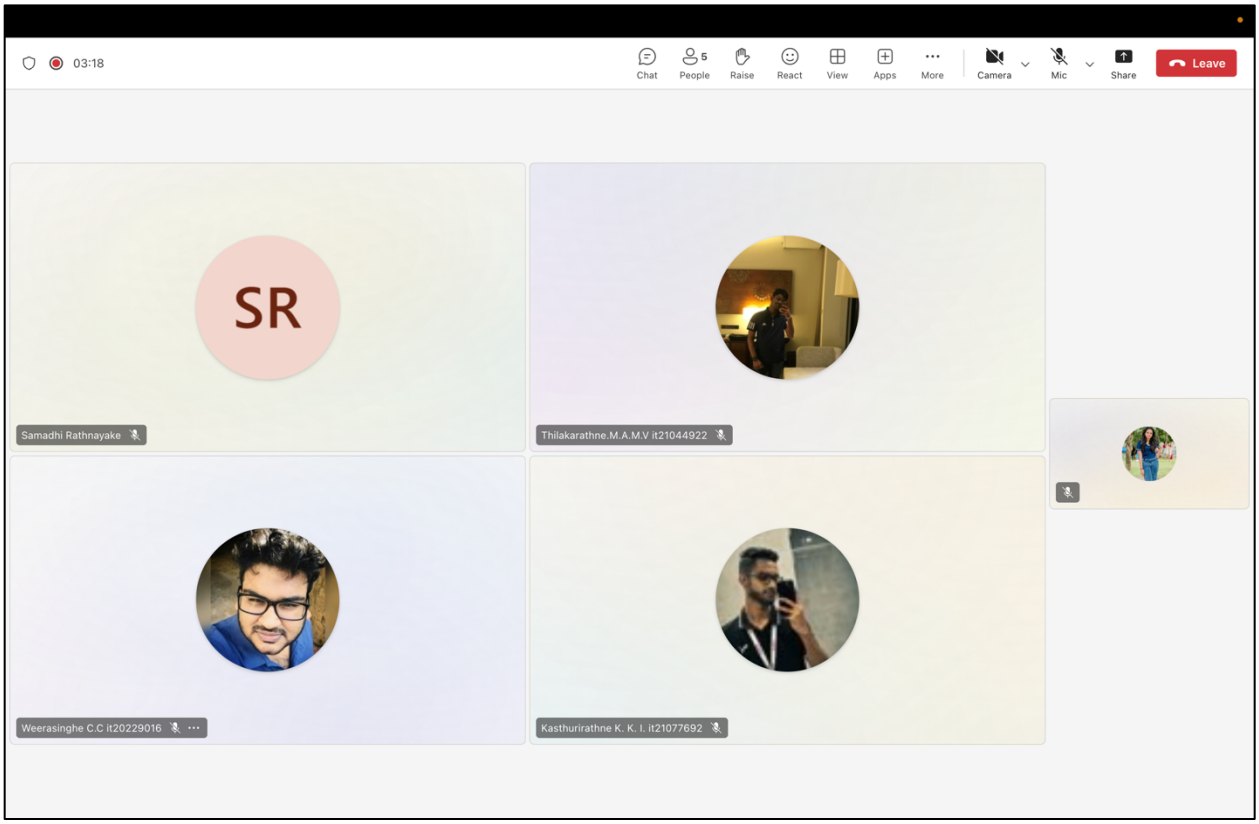


Figure 12 - Team Member Calls (Example 02)

Online Calls with Supervisors

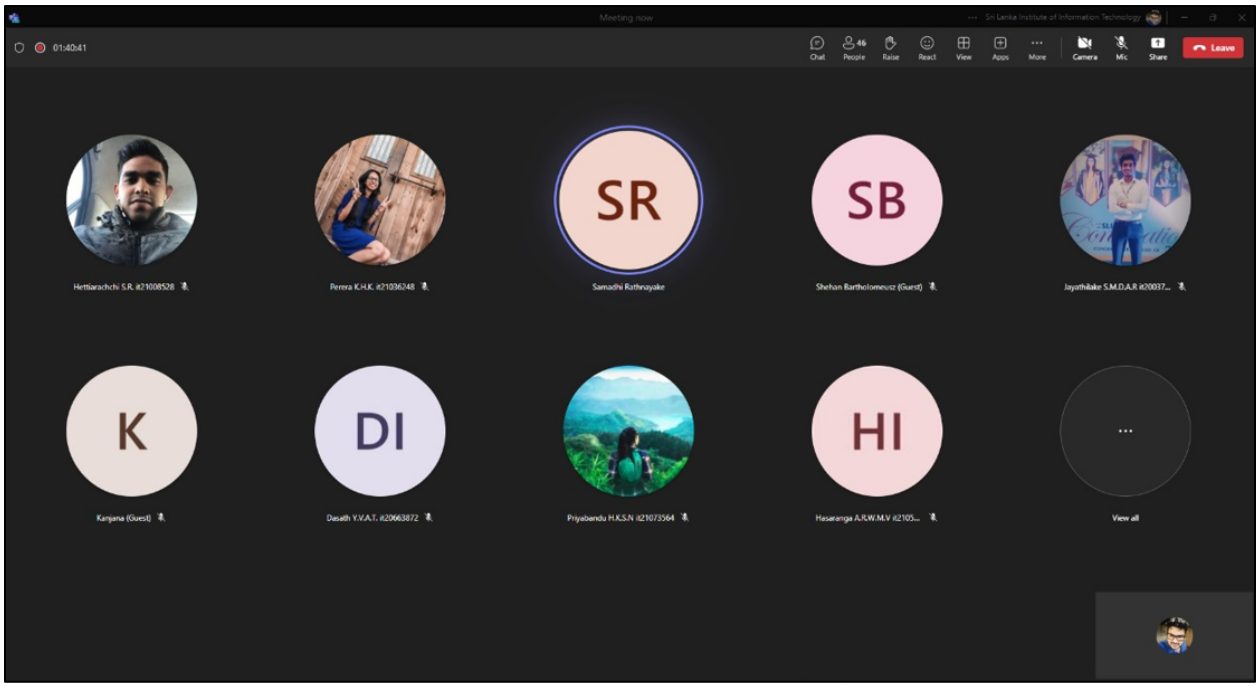


Figure 13 - Common Meeting with Supervisor's RP Groups

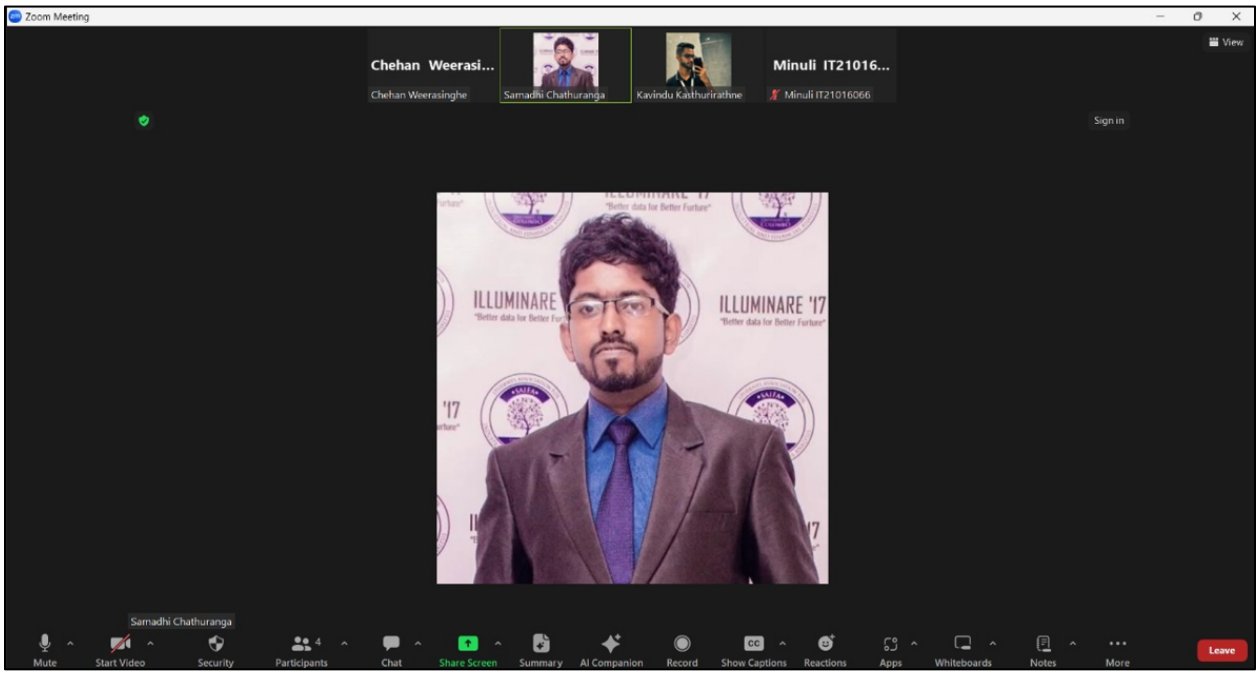


Figure 14 - ZOOM Meeting with the Supervisor (Example 01)

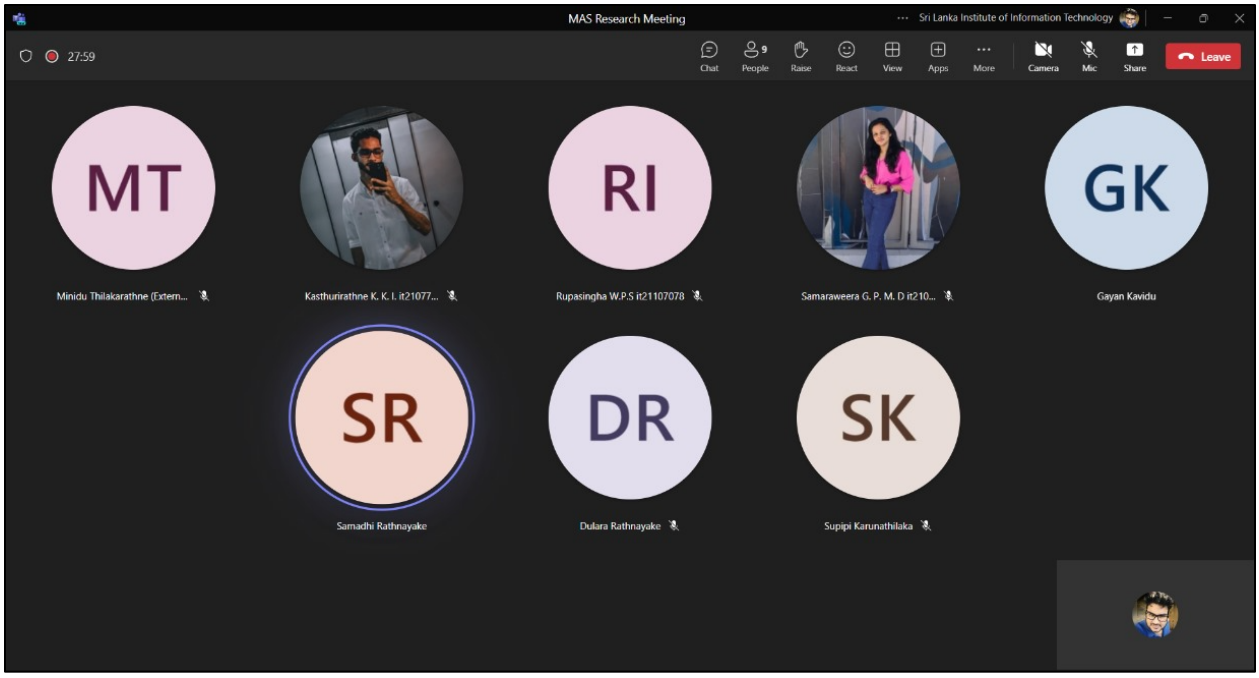


Figure 15 - Teams Call with Supervisor and External Supervisor (MAS)

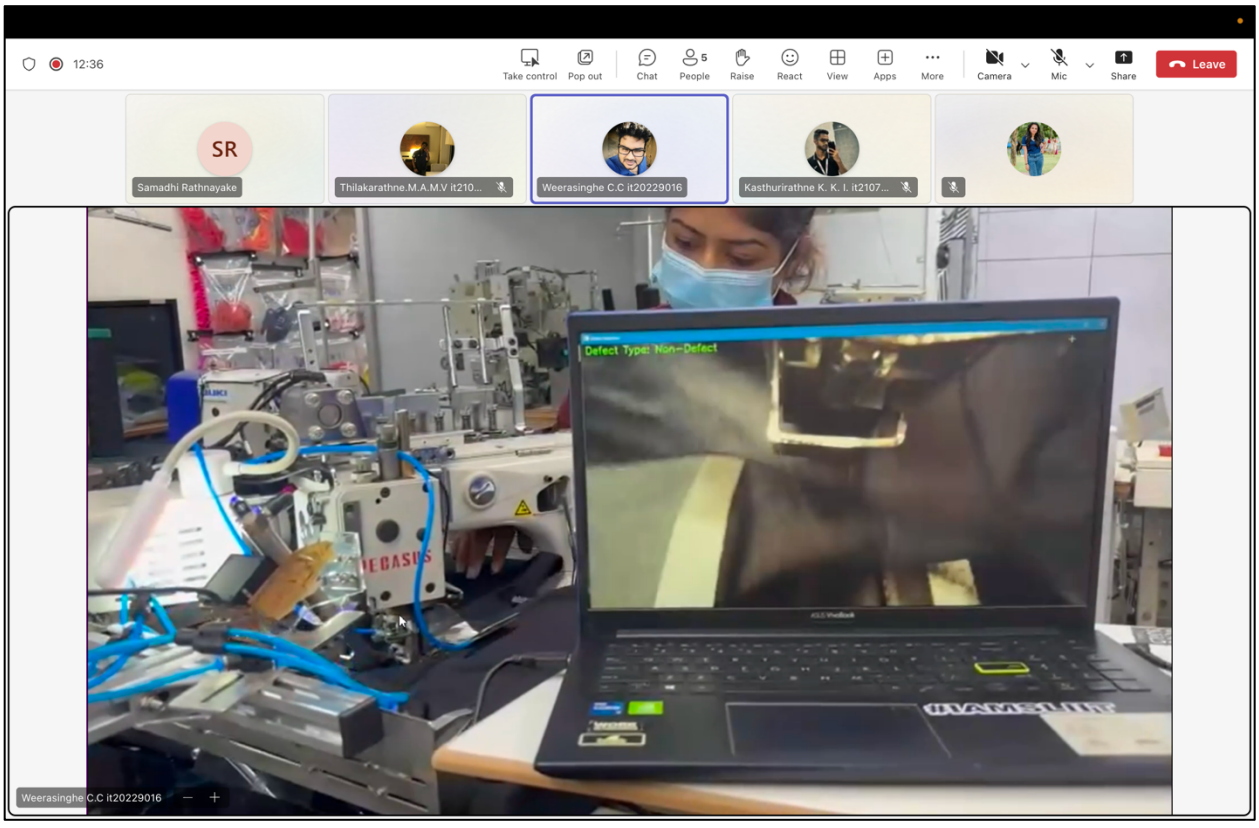


Figure 16 - Teams Meeting with the Supervisor (Example 02)

Approval Request for Research Paper Submission

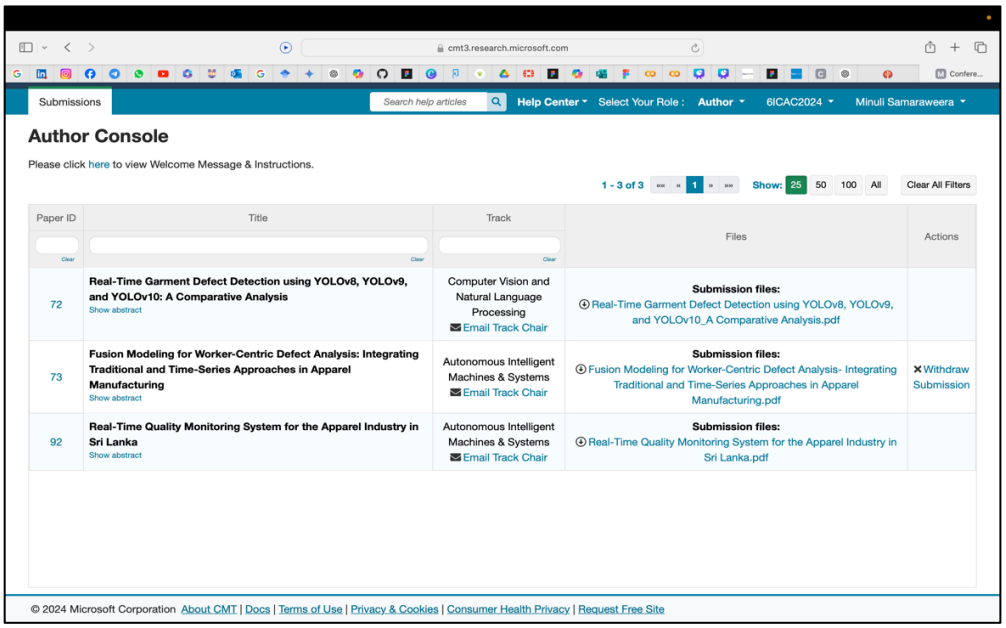


Figure 17 - Submit ICAC our research papers

Physical Meetings with the Supervisors and MAS Linea Aqua

Figure 18 - Physical Meeting with Supervisor and Co-Supervisor

Figure 19 - Field Visit to MAS Linea Aqua

WhatsApp Group Creation

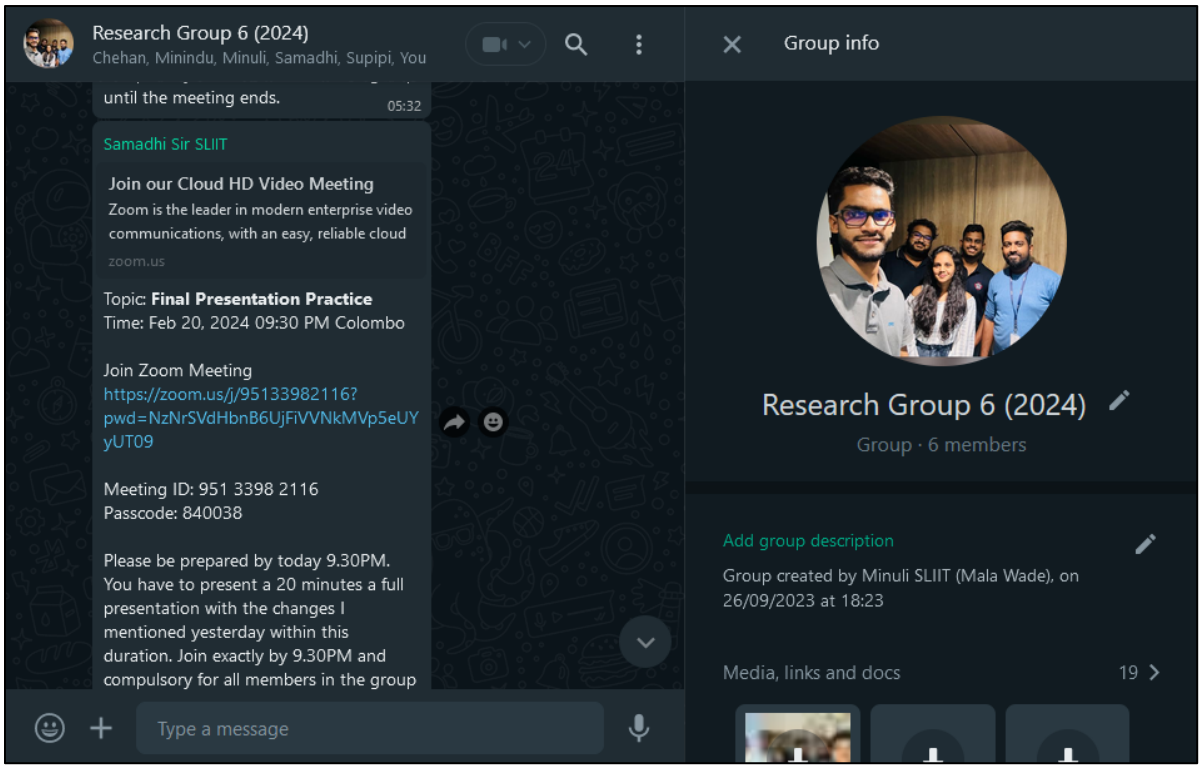


Figure 20 - WhatsApp Group Creation with Supervisor & Co-Supervisor

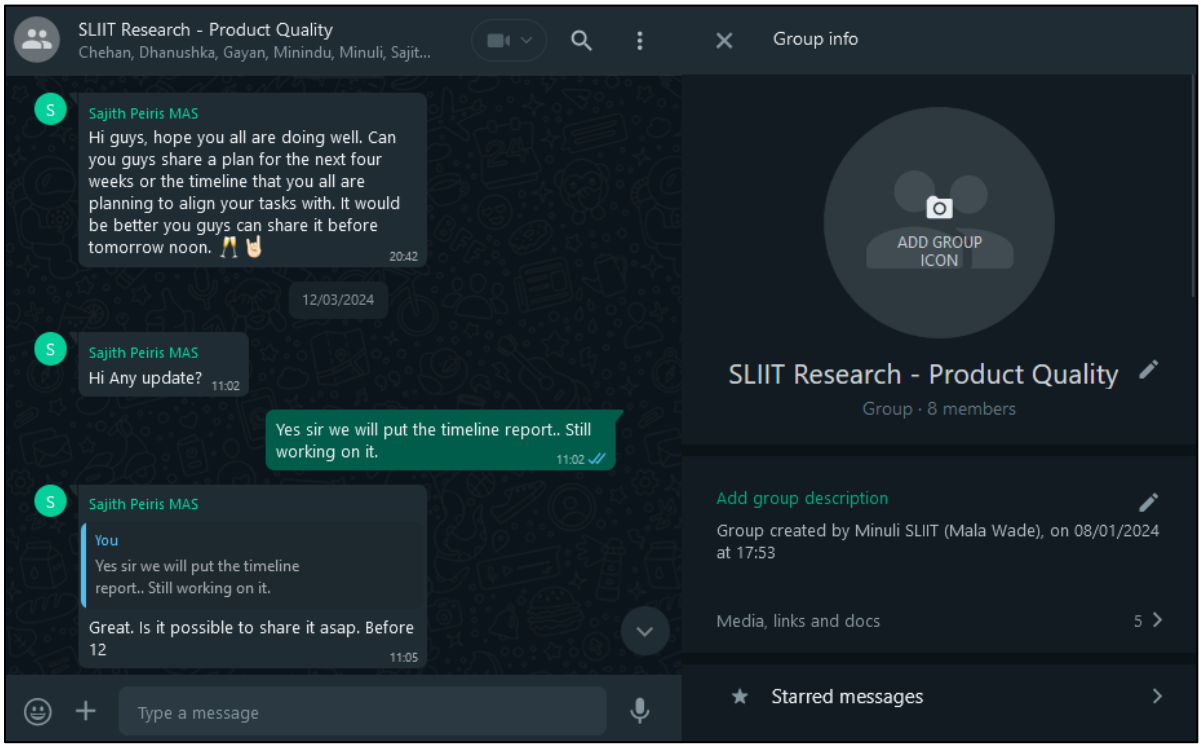


Figure 21 - WhatsApp Group Creation with External Supervisor (MAS)

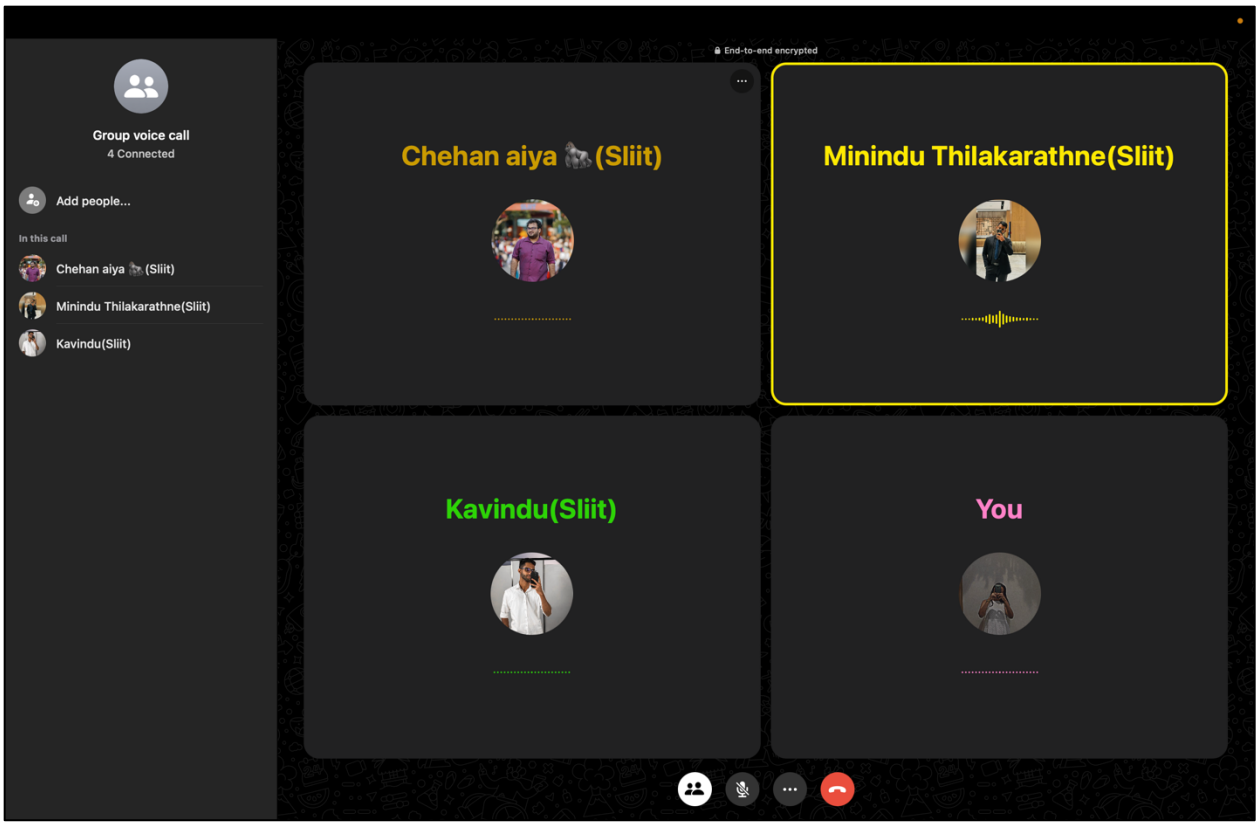


Figure 22 - WhatsApp Group Call with Group Members

Project Timeline

A Gantt chart is a visual tool used in project management to show the timeline of a project. It displays the start and finish dates of the various elements of a project, such as tasks, milestones, and phases, as well as their dependencies.

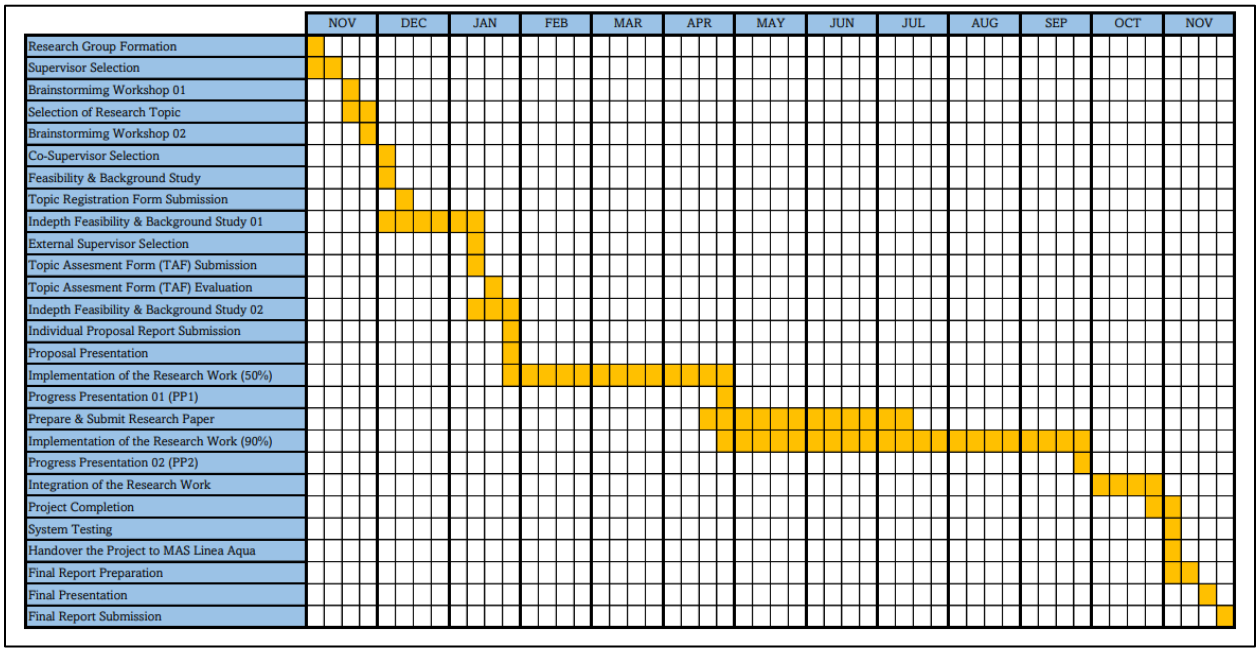


Figure 23 - Gannt Chart (Project Timeline)

Work Break-Down

A Work Breakdown Structure (WBS) is a hierarchical decomposition of a project into smaller, more manageable components. It organizes the project into specific deliverables and work packages, making it easier to plan, execute, and control the project.

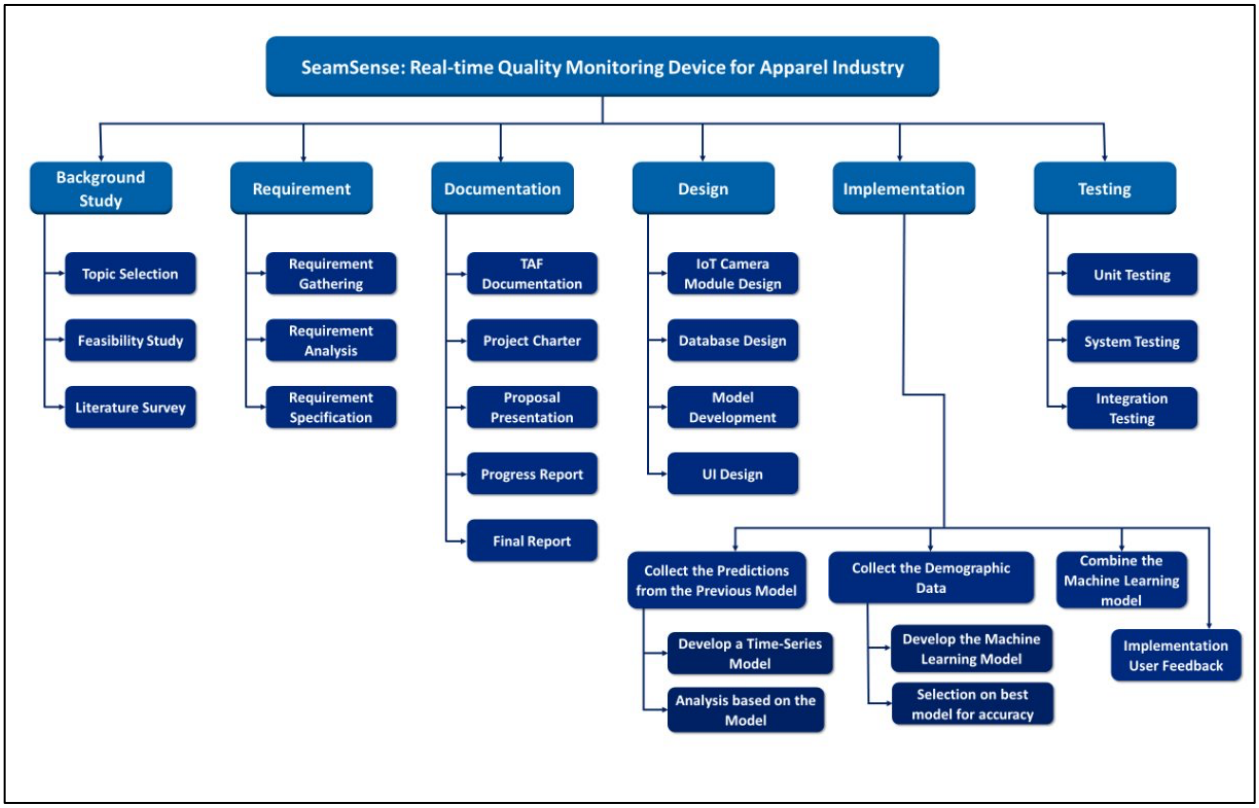


Figure 24 - Work Break-Down Structure

Version Controlling

The team implemented GitHub as their version control system, creating a repository that included all four group members. Each member was responsible for committing their code changes and progress to the repository.

To ensure transparency and collaboration, all code changes were incrementally committed and pushed to individual branches. These branches were later merged into the main branch during weekly meetings. This approach allowed for effective version control and streamlined collaboration among team members.

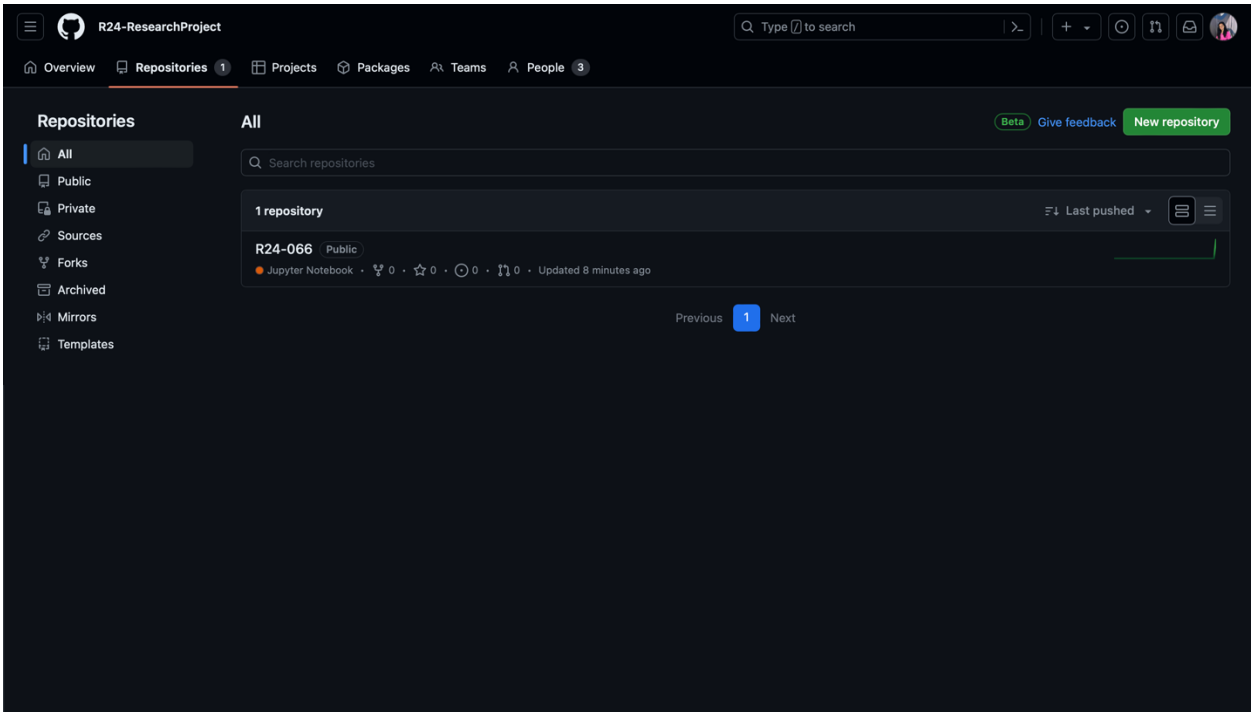


Figure 25 - GitHub Repository

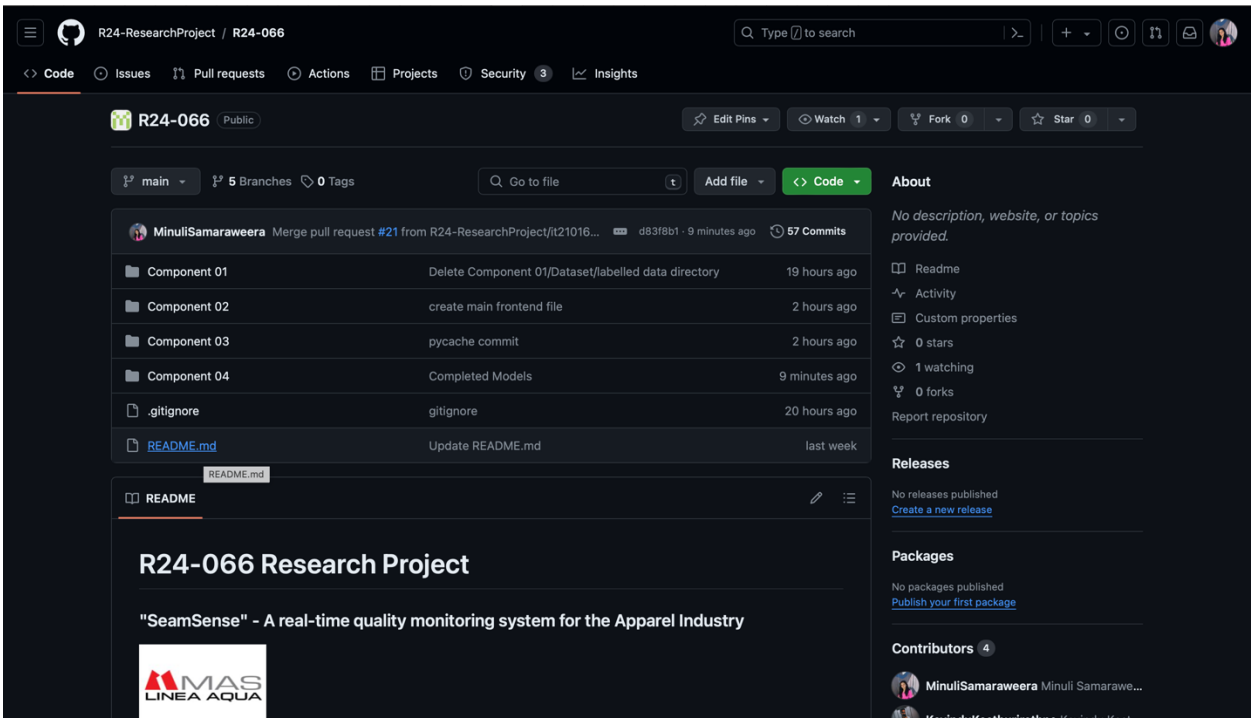
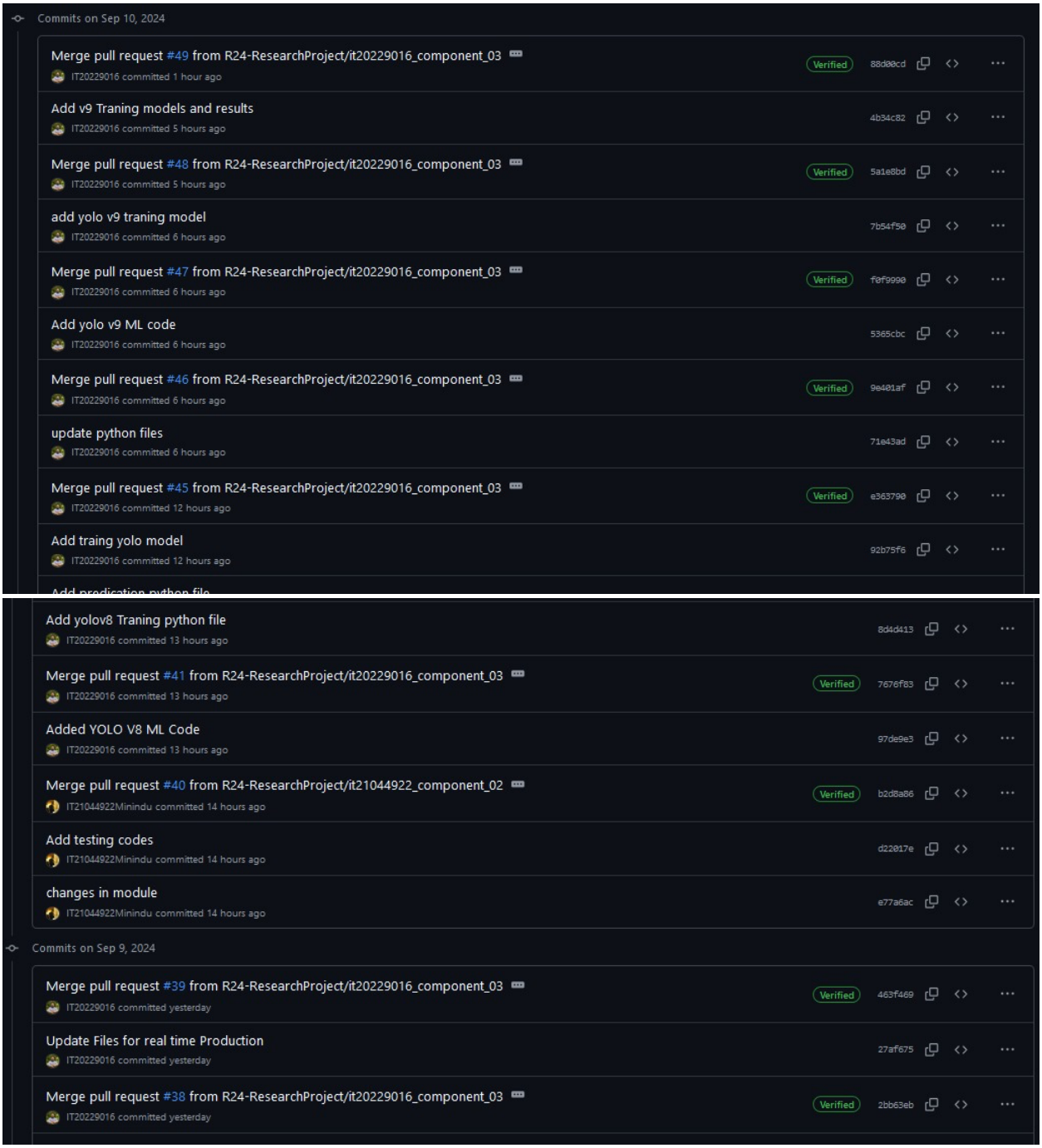
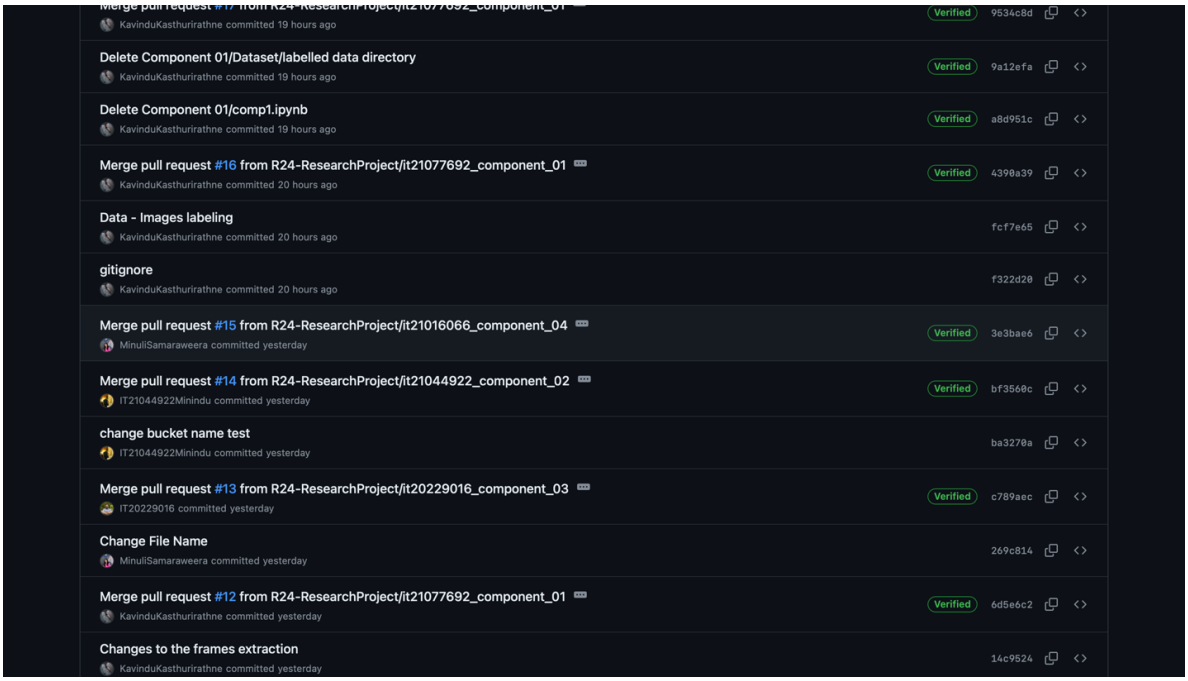
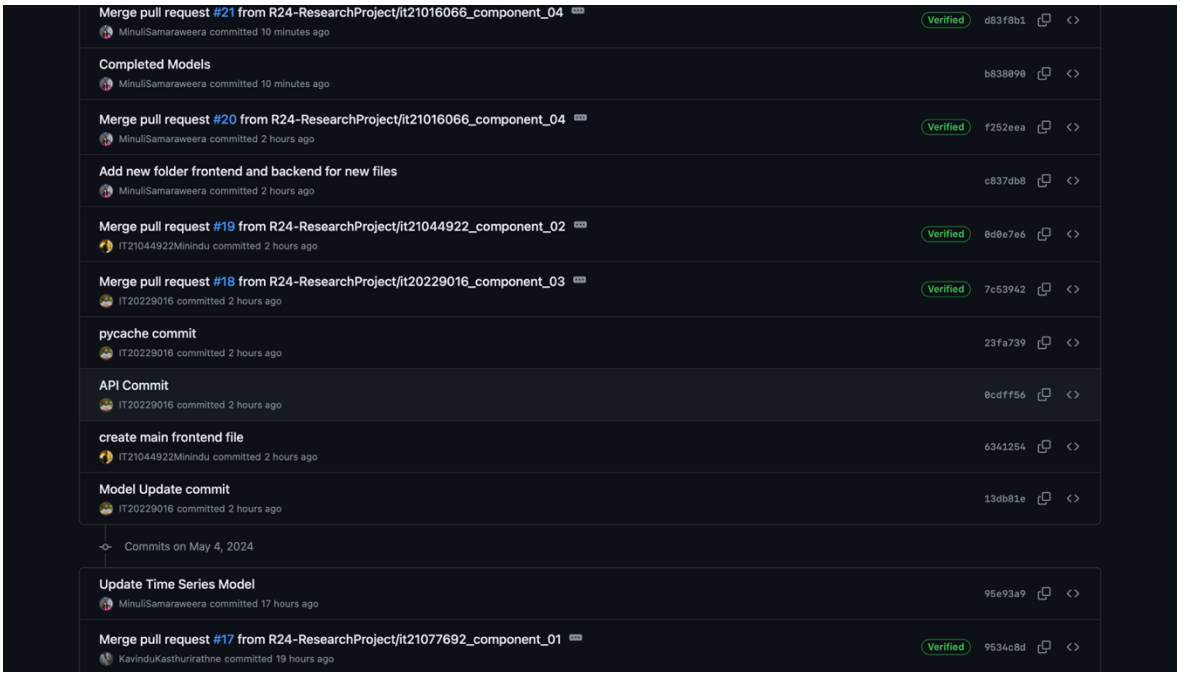


Figure 26 - GitHub Repository (Component 04 Branch)

GitHub Commits




KavinduKasthurirathne committed 19 hours ago
Verified 9534c8d
Delete Component 01/Dataset/labelled data directory
KavinduKasthurirathne committed 19 hours ago
Verified 9a12efa
Delete Component 01/comp1.ipynb
KavinduKasthurirathne committed 19 hours ago
Verified a8d951c
Merge pull request #16 from R24-ResearchProject/it21077692_component_01
KavinduKasthurirathne committed 20 hours ago
Verified 4390a39
Data - Images labeling
KavinduKasthurirathne committed 20 hours ago
fcf7e65
gitignore
KavinduKasthurirathne committed 20 hours ago
f322d20
Merge pull request #15 from R24-ResearchProject/it21016066_component_04
MinuliSamaraweera committed yesterday
Verified 3e3bae6
Merge pull request #14 from R24-ResearchProject/it21044922_component_02
IT21044922Minindu committed yesterday
Verified bf3560c
change bucket name test
IT21044922Minindu committed yesterday
ba3270a
Merge pull request #13 from R24-ResearchProject/it20229016_component_03
IT20229016 committed yesterday
Verified c789aec
Change File Name
MinuliSamaraweera committed yesterday
269c814
Merge pull request #12 from R24-ResearchProject/it21077692_component_01
KavinduKasthurirathne committed yesterday
Verified 6d5e6c2
Changes to the frames extraction
KavinduKasthurirathne committed yesterday
14c9524


Merge pull request #21 from R24-ResearchProject/it21016066_component_04
MinuliSamaraweera committed 10 minutes ago
Verified d83f8b1
Completed Models
MinuliSamaraweera committed 10 minutes ago
b838090
Merge pull request #20 from R24-ResearchProject/it21016066_component_04
MinuliSamaraweera committed 2 hours ago
Verified f252eea
Add new folder frontend and backend for new files
MinuliSamaraweera committed 2 hours ago
c837db8
Merge pull request #19 from R24-ResearchProject/it21044922_component_02
IT21044922Minindu committed 2 hours ago
Verified 0d0e7e6
Merge pull request #18 from R24-ResearchProject/it20229016_component_03
IT20229016 committed 2 hours ago
Verified 7c53942
pycache commit
IT20229016 committed 2 hours ago
23fa739
API Commit
IT20229016 committed 2 hours ago
0cdff56
create main frontend file
IT21044922Minindu committed 2 hours ago
6341254
Model Update commit
IT20229016 committed 2 hours ago
13db81e
Commits on May 4, 2024
Update Time Series Model
MinuliSamaraweera committed 17 hours ago
95e93a9
Merge pull request #17 from R24-ResearchProject/it21077692_component_01
KavinduKasthurirathne committed 19 hours ago
Verified 9534c8d

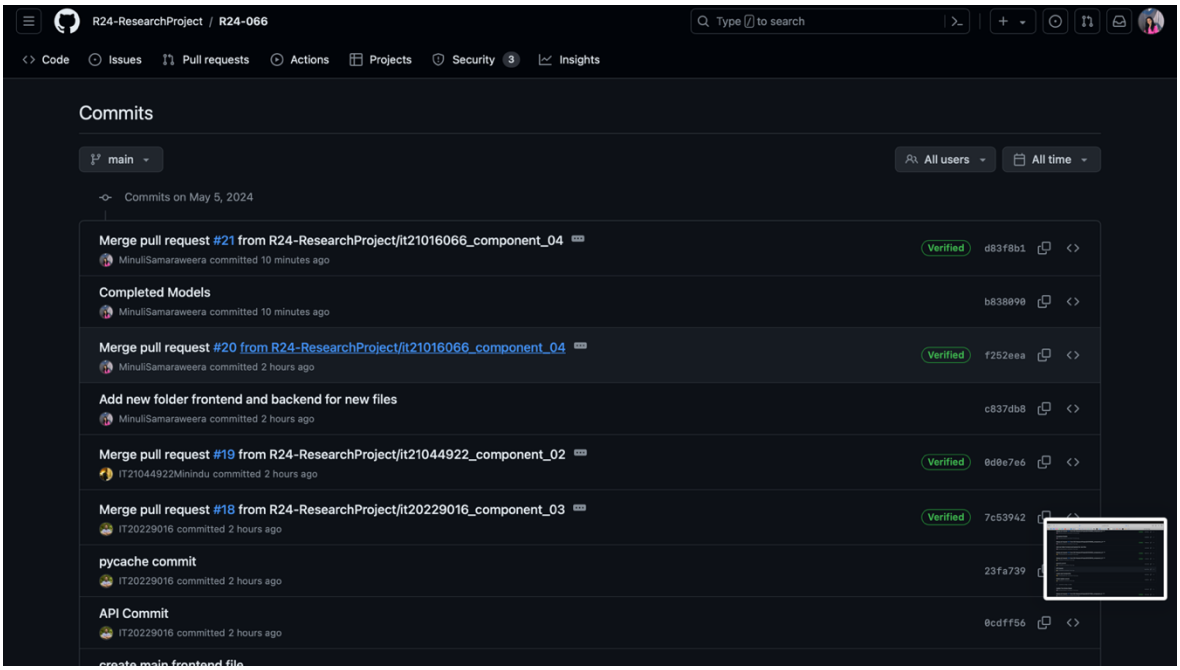
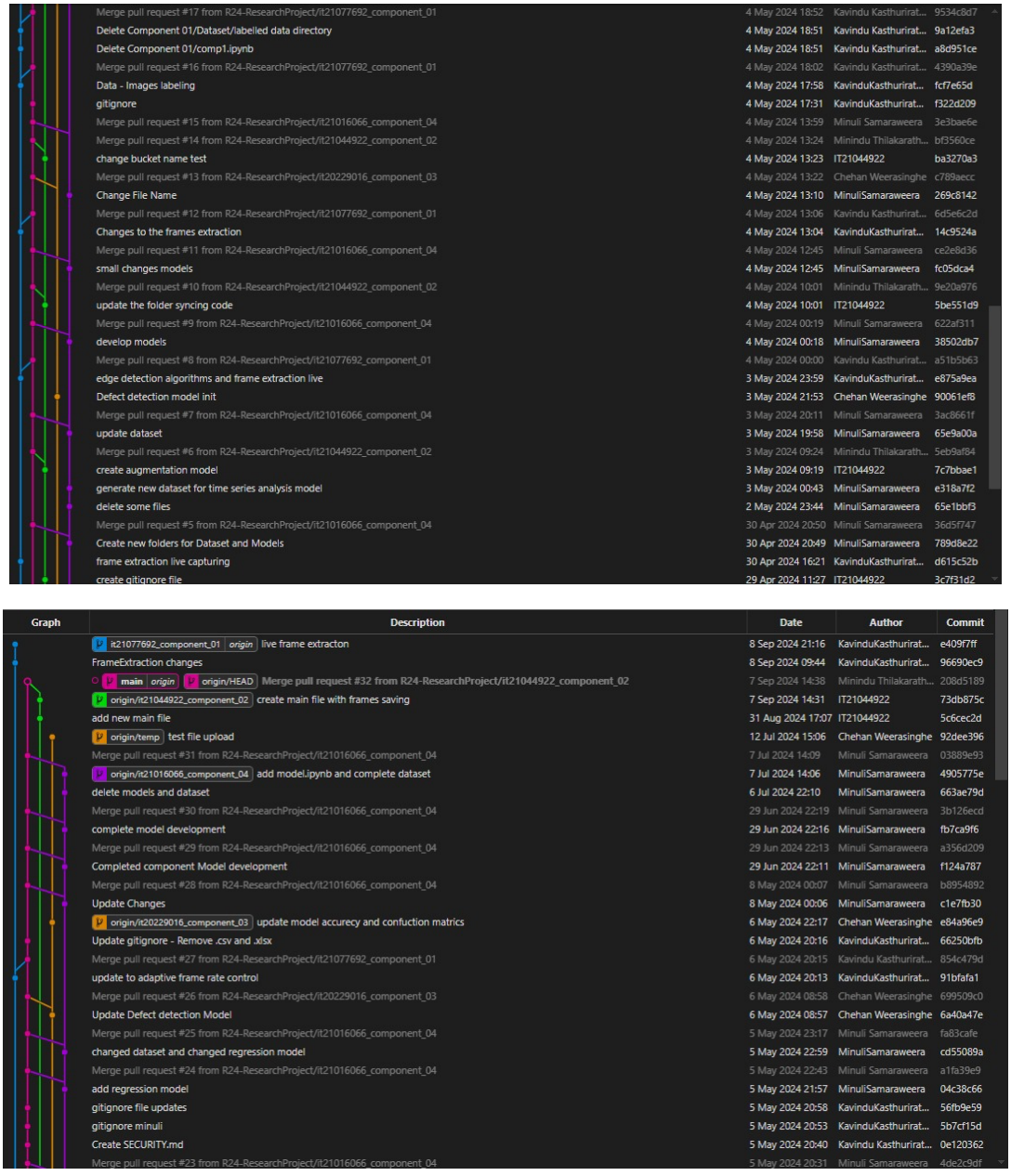


Figure 27 - GitHub Commits

GitHub Graph of Commits



Graph	Description	Date	Author	Commit
	it21077692_component_01 origin live frame extracton	8 Sep 2024 21:16	KavinduKasthurirat...	e409f7ff
	FrameExtraction changes	8 Sep 2024 09:44	KavinduKasthurirat...	96690ec9
	main origin origin/HEAD **Merge pull request #32 from R24-ResearchProject/it21044922_component_02**	7 Sep 2024 14:38	Minindu Thilakarath...	208d5189
	origin/it21044922_component_02 create main file with frames saving	7 Sep 2024 14:31	IT21044922	73db875c
	add new main file	31 Aug 2024 17:07	IT21044922	5c6cec2d
	origin/temp test file upload	12 Jul 2024 15:06	Chehan Weerasinghe	92dee396
	Merge pull request #31 from R24-ResearchProject/it21016066_component_04	7 Jul 2024 14:09	Minuli Samaraweera	03889e93
	origin/it21016066_component_04 add model.ipynb and complete dataset	7 Jul 2024 14:06	MinuliSamaraweera	4905775e
	delete models and dataset	6 Jul 2024 22:10	MinuliSamaraweera	663ae79d
	Merge pull request #30 from R24-ResearchProject/it21016066_component_04	29 Jun 2024 22:19	Minuli Samaraweera	3b126ecd
	complete model development	29 Jun 2024 22:16	MinuliSamaraweera	fb7ca9f6
	Merge pull request #29 from R24-ResearchProject/it21016066_component_04	29 Jun 2024 22:13	Minuli Samaraweera	a356d209
	Completed component Model development	29 Jun 2024 22:11	MinuliSamaraweera	f124a787
	Merge pull request #28 from R24-ResearchProject/it21016066_component_04	8 May 2024 00:07	Minuli Samaraweera	b8954892
	Update Changes	8 May 2024 00:06	MinuliSamaraweera	c1e7fb30
	origin/it20229016_component_03 update model accurecy and confuction matrics	6 May 2024 22:17	Chehan Weerasinghe	e84a96e9
	Update gitignore - Remove .csv and .xlsx	6 May 2024 20:16	KavinduKasthurirat...	66250bfb
	Merge pull request #27 from R24-ResearchProject/it21077692_component_01	6 May 2024 20:15	Kavindu Kasthurirat...	854c479d
	update to adaptive frame rate control	6 May 2024 20:13	KavinduKasthurirat...	91bfafa1
	Merge pull request #26 from R24-ResearchProject/it20229016_component_03	6 May 2024 08:58	Chehan Weerasinghe	699509c0
	Update Defect detection Model	6 May 2024 08:57	Chehan Weerasinghe	6a40a47e
	Merge pull request #25 from R24-ResearchProject/it21016066_component_04	5 May 2024 23:17	Minuli Samaraweera	fa83cafe
	changed dataset and changed regression model	5 May 2024 22:59	MinuliSamaraweera	cd55089a
	Merge pull request #24 from R24-ResearchProject/it21016066_component_04	5 May 2024 22:43	Minuli Samaraweera	a1fa39e9
	add regression model	5 May 2024 21:57	MinuliSamaraweera	04c38c66
	gitignore file updates	5 May 2024 20:58	KavinduKasthurirat...	56fb9e59
	gitignore minuli	5 May 2024 20:53	KavinduKasthurirat...	5b7cf15d
	Create SECURITY.md	5 May 2024 20:40	Kavindu Kasthurirat...	0e120362
	Merge pull request #23 from R24-ResearchProject/it21016066_component_04	5 May 2024 20:31	Minuli Samaraweera	4de2c9df

Figure 28 - Git Graph

GitHub Contributors

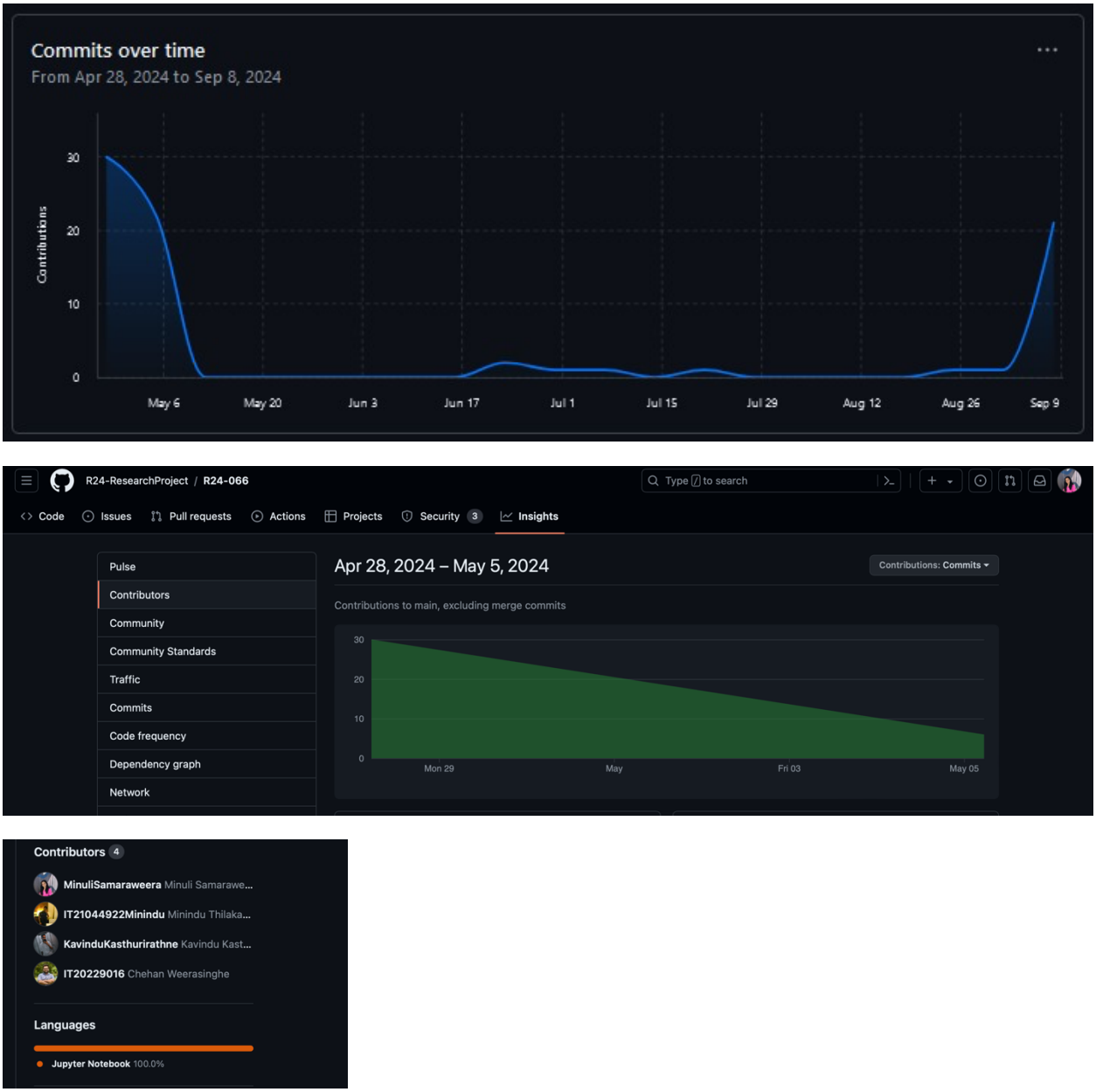


Figure 29 - GitHub Contributors

Contribution Charts

Figure 30 - Contribution Charts

Additional Project Status Information

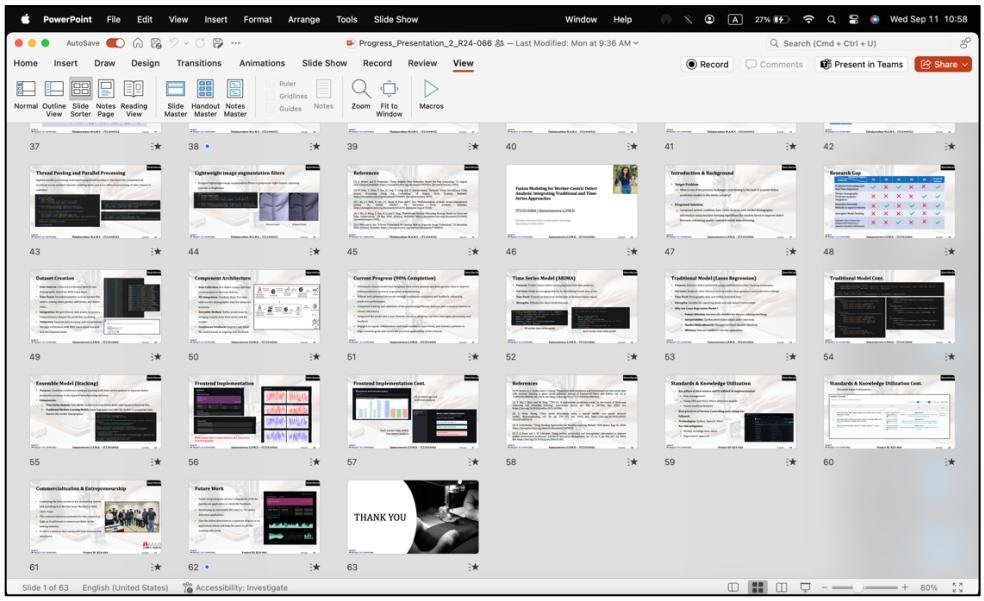


Figure 31 - Progress Presentation 2

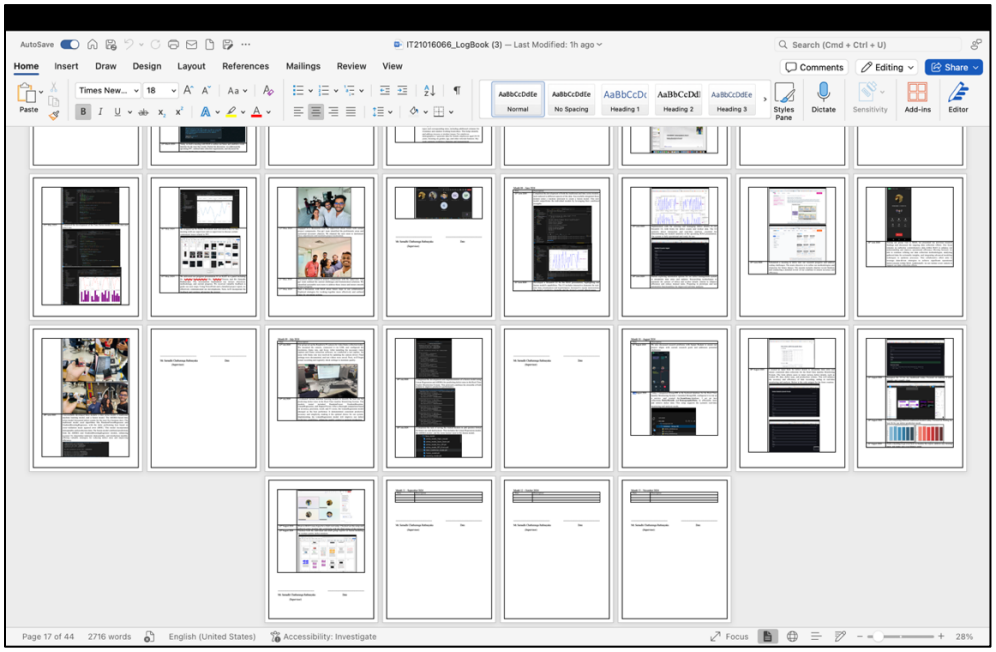


Figure 32 - Project Logbook