Overview of Planned Project Stages:

1. **Data Collection and Storage:** Collect and store a snapshot of the last five years of data from the Israeli Meteorological Service (IMS), using the 10-minute observations database and aggregating the data to turn it into hourly recordings to allow short-range forecasting. An API key allowing access to this data has already been obtained.

2. **Data Quality Assessment/Exploratory Data Analaysis:** Write a comprehensive data report that describes data quality/integrity like missing data to sensor errors, etc.

3. **Data Cleaning and Feature Engineering:** Implement data cleaning pipelines to prepare the dataset for feature engineering and model training, afterwards implement feature engineering pipelines (with a strong emphasis on no data leakage) for strong features from meteorological papers or relevant books.

4. **Statistical Testing:** Perform feature importance analysis and statistical tests to understand the correlation and gauge importance of each feature.

5. **Cross Validation:** Cross validate base machine learning models like XGBoost, Random Forsts and RNN LSTM on the dataset without engineered features and with on the exact weather stations in order to measure the impact of the engineered features on model performance and generalization.

6. **Final Forecasting Method (Regression Kriging):** Using the best performing ML model and feature set identified in step 5, implement the full Regression Kriging workflow with walk-forward validation.

7. **Fine-Tuning Models:** Finetune the hyperparamteres of the entire Regression Kriging pipeline and the ML model to maximize final performance.

Optional Extensions (subject to project time constraints):

8. **Backend API:** Build a backend API for the model using FastAPI or Flask, and deploy the trained model artifact to an AWS environment. The API will serve predictions from this model.

9. **Online Learning:** Deploy the model with online mini-batch learning. Its performance will be evaluated using RMSLE between the model's forecast and the IMS forecast each time it is trained on a new batch of data. If the RMSLE exceeds a certain threshold, the backend will trigger a retraining routine that includes the initial five-year snapshot and the new accumulated data, to prevent catastrophic forgetting and adapt to data drift (Trigger-Based Retraining).

10. **Frontend Application:** Develop a mobile front-end weather application in React Native that will communicate directly with the model's backend API and display the model's forecasts alongside the IMS forecasts. The application will include an interactive map using Leaflet with a grid over the Galilee and Nazareth landscape area, with clickable blocks, so that clicking on each block will display a small window comparing the model's forecast to the IMS forecast.