Week 1: Setup & Data Preparation

- **Issue 1.1: Development Environment & Repository Setup**
  Set up Git repositories, configure CI/CD (if needed), and establish communication channels.
- **Issue 1.2: Dataset Loading & Parsing**
  Load the GarmentCode dataset and parse `train_valid_test_data_split.json` using the `default_body` specification.
- **Issue 1.3: Data Preprocessing & Augmentation**
  Implement normalization, resizing, and other augmentations for the pattern images.
- **Issue 1.4: Pattern Type Mapping**
  Develop a script to create a mapping from images to pattern types using design parameters and segmentation.
- **Issue 1.5: Data Split Validation**
  Verify that the train, validation, and test sets are correctly split as per the provided JSON file.

Week 2: Model Component Development (Parallel)

- **Issue 2.1: Feature Extraction Module with ResNet50**
  Integrate a pre-trained ResNet50 for robust feature extraction and set up fine-tuning.
- **Issue 2.2: Dimension Estimation Module using EfficientNet-B0**
  Adapt EfficientNet-B0 for predicting dimensions (transition from 2048 features to 2 outputs: width and height).
- **Issue 2.3: LSTM-Based Classification Module**
  Develop an LSTM network to capture sequential/contextual cues and modify the final layer to output the number of classes.
- **Issue 2.4: U-Net Module for Corner Detection and Segmentation**
  Build and validate the U-Net architecture for accurate segmentation and corner detection.
- **Issue 2.5: Unit Testing for Individual Modules**
  Write unit tests for each module to ensure they perform as expected before integration.

Week 3: Integration & Training Pipeline Setup

- **Issue 3.1: Pipeline Integration**
  Integrate the modules (normalization, feature extraction, classification, segmentation, and dimension estimation) into one cohesive pipeline.

- **Issue 3.2: Loss Functions & Optimizer Setup**
  Define and implement the loss functions for classification, regression (dimensions), and segmentation, and set up the optimizer.
- **Issue 3.3: Training Loop Implementation**
  Develop the training loop that processes each epoch and batch, computes the total loss, backpropagates, and updates weights.
- **Issue 3.4: Logging & Metric Monitoring**
  Integrate logging to track loss, accuracy, and other key metrics during training.
- **Issue 3.5: Model Checkpointing**
  Implement checkpointing to save the best model based on validation performance.

Week 4: Model Training & Validation

- **Issue 4.1: Run Initial Training Sessions**
  Execute the first training runs and monitor model performance on the training set.
- **Issue 4.2: Hyperparameter Tuning**
  Adjust learning rates, batch sizes, and other hyperparameters based on initial results.
- **Issue 4.3: Validation & Best Model Selection**
  Validate the model on the validation set at the end of each epoch and select the best-performing checkpoint.
- **Issue 4.4: Debugging Training Issues**
  Identify and fix issues like vanishing gradients, overfitting, or data pipeline bottlenecks.
- **Issue 4.5: Documentation of Training Process**
  Document the training configuration, hyperparameters, and results for future reference.

Week 5: Inference Pipeline & HRL Integration

- **Issue 5.1: Inference Pipeline Development**
  Build the inference pipeline to load the saved model and process user input images.
- **Issue 5.2: Preprocessing for User Input**
  Implement the same normalization and feature extraction steps for user-provided pattern images.
- **Issue 5.3: Feature & Corner Extraction for Patterns**
  Integrate the modules to extract features, detect corners, and estimate dimensions from user images.
- **Issue 5.4: Cloth Image Processing Pipeline**
  Develop a similar preprocessing pipeline for the user's cloth image (extract edges, contours, dimensions, and area).
- **Issue 5.5: HRL-Based Placement Module Integration**
  Begin integrating the extracted features into the HRL module for optimal pattern placement.

Week 6: Final Integration, Testing & Deployment

- **Issue 6.1: End-to-End Pipeline Testing**
  Conduct comprehensive testing of the entire pipeline from user input through HRL placement.
- **Issue 6.2: Integration Debugging & Bug Fixes**
  Resolve any issues found during integration testing and optimize performance.
- **Issue 6.3: Final Validation with Real Data**
  Collaborate with team member <@321432498959679488> to run final tests on real hardware.
- **Issue 6.4: Prepare Documentation & Deployment Guidelines**
  Document the system architecture, setup instructions, and user guidelines for deployment.
- **Issue 6.5: Post-Mortem & Sprint Review**
  Conduct a final review meeting to discuss lessons learned and plan for the next stage (i.e., fitting the pattern inside the cloth).