**Senior Developer Plan**

**1. Understand and Prepare Data**

* **Objective:** Load and merge the datasets into a single unified dataset for analysis.
* **Steps:**
  1. Identify the data schema and clean missing or inconsistent values.
  2. Perform exploratory data analysis (EDA) to summarize the distribution of variables like gender, age, BMI, smoking status, geographic location, and number of dependents.
  3. Engineer necessary features:
     + Categorize BMI into health status (Underweight, Normal, Overweight, Obese).
     + Encode categorical variables like gender and smoker status.
  4. Split the dataset into analysis data (to answer questions) and model data (to train the prediction model).

**2. Analytical Insights**

Perform statistical and visual analyses for the following:

1. Gender constraints for extending policies.
2. Average amount spent on each policy.
3. Differences in policy requirements by geographic location.
4. Impact of the number of dependents on claim amounts.
5. Correlation between BMI and insurance claims.
6. The effect of smoking status on claims.
7. Age as a factor in claims.
8. Feasibility of health-based (BMI-related) discounts.

**3. Machine Learning Pipeline**

* **Model Objective:** Predict the amount spent for each policy cover.
* **Steps:**
  1. **Feature Selection:** Choose predictive features (e.g., age, BMI, gender, smoker status, dependents, location).
  2. **Model Selection:** Start with a few models like:
     + Linear Regression (for simplicity).
     + Decision Tree or Random Forest (for non-linear relationships).
     + Gradient Boosting (XGBoost, LightGBM, or CatBoost).
  3. **Data Split:** Train-test split (e.g., 80%-20%) and create a validation set using cross-validation.
  4. **Hyperparameter Tuning:** Use Grid Search or Randomized Search for hyperparameter optimization.
  5. **Evaluation Metrics:** Use metrics like RMSE, MAE, and R² to assess model performance.

**4. Final Deliverables**

* Analytical report answering the provided questions.
* Well-documented Python code implementing:
  + EDA and insights.
  + Machine learning pipeline.
* A detailed guide for selecting models and interpreting results.

**Guide to Model Selection**

1. **Start Simple:** Use linear models for a baseline when relationships are mostly linear.
2. **Understand Data:** If non-linear relationships or interactions exist, try tree-based models like Random Forest or Gradient Boosting.
3. **Handle Complexities:** Gradient Boosting models (e.g., XGBoost, LightGBM) are powerful for complex datasets.
4. **Experimentation:** Use cross-validation to test different models systematically.
5. **Metrics-Based:** Choose the model with the best performance on evaluation metrics like RMSE, MAE, and R².