RealVision Team: Project Demand Analayzer Functional Architecture Report

The website is designed to serve as a project estimation tool for real estate companies. It helps assess whether a housing project will be demanded by the market based on materials, costs, location, and other related factors. The goal is to estimate potential demand and support better planning decisions.

Major Pages

1. Home Page:This is the first page users see. It introduces the purpose of the system and provides navigation to other sections. It briefly explains how the estimation works

2. Report Page:This page shows the results of the estimation. After a user enters material and project data, the system calculates and displays summaries, charts, and feasibility scores. It helps users easily understand the cost breakdown and potential buyers interest.

3. Materials & Calculation/Estimation Page:This is the main working page of the system. Here, users input details such as material type, quantity, and project location. The system uses this data to calculate estimated construction costs and predict project success

Major PHP Functions

1. Validation:Checks that all user inputs (like material prices or quantities) are correct, complete, and in the right format before processing.

2. Verification:Confirms that the data entered matches valid records in the database (for example, confirming that a material type exists and has a set price).

3. Display from Database:Retrieves and shows stored data such as materials, prices, and previous estimations on the web pages.

Functional Description of the Project Demand Analyzer (PDA): 1st Sprint

The Project Demand Analyzer (PDA) is designed as a streamlined, data-driven tool to address the challenge of inconsistent material costing and subjective market assessment within the Ghanaian real estate sector. The PDA's architecture is split into two essential components: a user-friendly, multi-page **Web Application** (front-end and PHP data layer) and an intelligent **Machine Learning Model** (Python back-end) that provides the core predictive insight.

The Web Application: User Interface and Data Management

The front-end is structured around a simple dashboard system, built on HTML, CSS, and Vanilla JavaScript, ensuring accessibility and rapid calculation.

The **Dashboard (`index.html`)** is the functional core. Its primary purpose is to enable the **Instant Cost Tally** (F-1). Users input quantities for key materials (like Cement and Blocks) via a simple form, and the main project summary panel calculates the **Total Project Value** by cross-referencing these inputs with current unit prices. Crucially, this page also houses the future **Demand Insights** section, which is where the outcome of the machine learning model will be displayed as the **Demand Feasibility Score (DFS)** after the "Assess Demand" button is clicked.

Supporting the Dashboard are essential administrative and reference pages:

- The **Materials DB (`materials.html`)** serves as a transparent reference point, displaying the **static prices** (e.g., GHS 85.00 for a 50kg bag of Cement) that are used for the costing calculations. This represents the `GHA_PRICES` data structure in the architecture.
- The **Recent Analyses (`recent.html`)** page provides persistent storage and historical context, allowing users (Junior Sales Estimators) to review past project assessments, total values, and their corresponding demand scores (e.g., Residential Build - Pokuase, 72% Demand Score).
- The **Settings (`settings.html`)** and authentication pages (`login.html`, `signup.html`) ensure platform security and allow users to manage basic preferences, such as preferred currency, providing a professional and maintainable application shell.

The Back-End Intelligence: The Machine Learning Model

The second critical component is the **predictive logic** executed by the Python-based machine learning algorithm (as demonstrated in the Google Colab notebook). The front-end cannot function as a "Demand Analyzer" without this intelligence.

The Python script employs **Linear Regression**—a simple but effective predictive model—to solve the problem of subjective demand assessment. The model is trained on simulated market data (Location, Property Type, Price) to predict a **Demand Multiplier (DM)**. This DM is the numerical representation of a project's market appeal.

The final interaction is a full-stack effort: the web application passes the calculated **Total Project Value** and the selected location/type to the PHP back-end, which then applies the *knowledge* generated by the Python model (the DM) to execute the final **Demand Feasibility Score (DFS) Generator** (F-3). This integration is what transforms the PDA from a mere calculator into a valuable business insight tool for our client, AccraBuild Streamline.

DESCRIPTION OF FRONT-END CHOICES

For our project so far, we have used a custom HTML and CSS frontend to create a clean, user-friendly interface without relying on heavy external frameworks. The layout and design are fully hand-coded in CSS, giving us complete control over the website’s appearance and responsiveness.

We chose to use Flexbox and CSS Grid for layout management instead of a pre-built CSS library, such as Bootstrap. This decision allows for lightweight styling, faster loading times, and a deeper understanding of responsive design principles. The colour scheme incorporates snow and dark grey to create a professional and visually appealing look.

For interactivity and form validation, we use vanilla JavaScript (no frameworks, such as jQuery, React, or Angular, so far). This ensures compatibility across all modern browsers while keeping the code simple and easy to maintain. A separate JavaScript file manages form validation, user interactions, and dynamic updates.

Overall, our frontend approach focuses on simplicity, performance, and maintainability, providing a solid foundation for potential integration with larger frameworks in the future, should the project scale.

DESCRIPTION OF TEAMMATES CONTRIBUTIONS

Nana Kwame: Handled updates on Architecture page and common deliverables

Hagen: Video demo on page interactivity and database architecture.

Edwin: Database architecture creation and database ERD

Kobby: Database architecture creation and ERD

Dalziel: Database Architecture creation

Retrospection

In the course of preparing our deliverables for the 2nd Sprint, we encountered a few problems in the creation of our database architecture as well as the entities to be included in the final database architecture and ERD. After further group deliberation, we ascertained that we would only need two entities, namely the materials entity(for the use of materials that users would select) and the housing_properties entity (for specific socioeconomic elements that would impact the demand for the housing property). After this, we quickly mobilised to put together our deliverables before making our final submission.

Moving forward, we would endeavor to make more structured checks and deliberations to ensure a common understanding about our project's necessary elements is shared across the group, and would thus be more timely in our workflow.