REST API Specifications for Demand Forecasting System

Overview

This document outlines the specifications for the REST API that replaces the current Google Firestore implementation. The API will interact with a MariaDB database while maintaining the same data structure and functionality.

Database Structure

Main Table: `forecast_data`

Primary keys:

- `Year_Month` (DATE) - The month of the forecast
- `Product code` (VARCHAR) - The product identifier

Fields:

- `Product Group` (VARCHAR) - Product group identifier
- `Product description` (VARCHAR) - Description of the product
- `prod_class` (VARCHAR) - Product class identifier
- `Quantity` (DECIMAL) - Actual quantity
- `new_forecast` (DECIMAL) - New forecast value
- `old_forecast` (DECIMAL) - Previous forecast value
- `old_forecast_error` (VARCHAR) - Error message for old forecast
- `correction_percent` (DECIMAL) - Correction percentage
- `explanation` (TEXT) - Explanation for the forecast
- `new_forecast_manually_adjusted` (DECIMAL) - Manually adjusted forecast value
- `correction_timestamp` (TIMESTAMP) - Timestamp of the correction
- `forecast_corrector` (VARCHAR) - User who made the forecast correction
- `last_manual_correction_date` (TIMESTAMP) - Date of the last manual correction
- `id` (VARCHAR) - Unique identifier
- `created_at` (TIMESTAMP) - Record creation timestamp
- `updated_at` (TIMESTAMP) - Record last update timestamp

API Endpoints

1. Get Forecast Data

```
GET /api/forecast
```

Query Parameters:

- `Year_Month` (optional) - Filter by specific month (YYYY-MM-DD format)
- `Product code` (optional) - Filter by specific product
- `Product Group` (optional) - Filter by product group
- `start_date` (optional) - Start date for date range
- `end_date` (optional) - End date for date range

Response:


```
{
    "data": [
        {
            "Year_Month": "2024-03-01",
            "Product Group": "GROUP_A",
            "Product code": "PROD_001",
            "Product description": "Sample Product",
            "prod_class": "CLASS_A",
            "Quantity": 100.5,
            "new_forecast": 98.2,
            "old_forecast": 95.0,
            "old_forecast_error": null,
            "correction_percent": 3.2,
            "explanation": "Adjusted based on market trends",
            "new_forecast_manually_adjusted": 98.2,
            "correction_timestamp": "2024-03-15T10:00:00Z",
            "forecast_corrector": "user@example.com",
            "last_manual_correction_date": "2024-03-15T10:00:00Z",
            "id": "forecast_123",
            "created_at": "2024-03-15T10:00:00Z",
            "updated_at": "2024-03-15T10:00:00Z"
        }
    ]
}
```


2. Create/Update Forecast

```
POST /api/forecast
```

Request Body:


```
{
    "Year_Month": "2024-03-01",
    "Product Group": "GROUP_A",
    "Product code": "PROD_001",
    "Product description": "Sample Product",
    "prod_class": "CLASS_A",
    "Quantity": 100.5,
    "new_forecast": 98.2,
```

```
    "old_forecast": 95.0,
    "old_forecast_error": null,
    "correction_percent": 3.2,
    "explanation": "Adjusted based on market trends",
    "new_forecast_manually_adjusted": 98.2,
    "correction_timestamp": "2024-03-15T10:00:00Z",
    "forecast_corrector": "user@example.com",
    "last_manual_correction_date": "2024-03-15T10:00:00Z"
}
```

Response:

```
{
    "success": true,
    "message": "Forecast data saved successfully",
    "data": {
        "Year_Month": "2024-03-01",
        "Product Group": "GROUP_A",
        "Product code": "PROD_001",
        "Product description": "Sample Product",
        "prod_class": "CLASS_A",
        "Quantity": 100.5,
        "new_forecast": 98.2,
        "old_forecast": 95.0,
        "old_forecast_error": null,
        "correction_percent": 3.2,
        "explanation": "Adjusted based on market trends",
        "new_forecast_manually_adjusted": 98.2,
        "correction_timestamp": "2024-03-15T10:00:00Z",
        "forecast_corrector": "user@example.com",
        "last_manual_correction_date": "2024-03-15T10:00:00Z",
        "id": "forecast_123",
        "created_at": "2024-03-15T10:00:00Z",
        "updated_at": "2024-03-15T10:00:00Z"
    }
}
```

3. Delete Forecast

```
DELETE /api/forecast
```

Query Parameters:

- `Year_Month` (required) - Month to delete (YYYY-MM-DD format)
- `Product code` (required) - Product identifier to delete

Response:

```
{
    "success": true,
    "message": "Forecast data deleted successfully"
}
```

Error Handling

All endpoints will return appropriate HTTP status codes:

- 200: Success
- 400: Bad Request (invalid parameters)
- 404: Not Found
- 500: Internal Server Error

Error Response Format:

```
{
    "error": {
        "code": "ERROR_CODE",
        "message": "Human readable error message"
    }
}
```

Authentication

The API will use token-based authentication:

- All requests must include an `Authorization` header with a Bearer token
- Token format: `Authorization: Bearer <token>`

Rate Limiting

- Maximum 100 requests per minute per IP address
- Rate limit headers will be included in responses:
  - `X-RateLimit-Limit`
  - `X-RateLimit-Remaining`
  - `X-RateLimit-Reset`

Data Validation

- Year_Month must be in YYYY-MM-DD format
- Product code must be a non-empty string
- Numeric values must be valid decimal numbers
- All required fields must be present in POST requests

Notes

- The API maintains backward compatibility with the existing Firestore implementation
- All timestamps are in UTC
- The database uses the same schema as the current Firestore implementation
- Primary keys (Year_Month, Product code) ensure data uniqueness and efficient querying