Progress Report 1

Intelligent Business Analytics System

- for Maximizing Revenue and Efficiency

Student Name: Seungyeol Chae

Student Number: 300362271

Work Logs

Date	Hours	Description of Work
January 25, 2025	3	Started the project on GitHub with the first commit. Created a structured repository directory system based on my project proposal.
January 27, 2025	6	Designed the overall system architecture by defining backend services, database schema, and API endpoints. Set up FastAPI and Flask for backend logic and PostgreSQL for structured data storage and processing. Tested uploading test excel file feature and checked it is successfully working.
January 30, 2025	4	Worked on preprocessing the uploaded test data to show the desired results. Identified and removed incorrect pull requests to migrate the frontend to React.js as frontend for better and flexible UI. Worked on the project set up and build locally for easier debugging and development. Ensured Python virtual environment was properly structured.
February 1, 2025	6	Fixed incorrect pull requests again, updated several files, and stabilized the project structure. Implemented core backend functionality using FastAPI and Flask for API services. Successfully created charts and tables as I desired.
February 3, 2025	5	Refactored API endpoints and modified data handling to align with the new frontend structure. Started making new features to show results like showing best/worst item based on reviews based on NLTK analysis
February 5, 2025	6	Implemented and debugged the **NLTK sentiment analysis feature** to make the result more accurate to show the percentage of positive/negative reviews. Optimized data visualization, refining charts to better present insights.
February 8, 2025	3	Updated the README.md file with improved content and a more structured layout to enhance clarity and usability for users.

Description of Work Done

This progress period involved critical improvements in backend stability, frontend flexibility, and analytical capabilities within the Intelligent Business Analytics System.

1. Repository Setup & GitHub Configuration
    - Structured the project repository with clear directory separation for API, services, and utilities.
    - Implemented CI/CD workflows using GitHub Actions for automated testing and deployment.
    - Set up a robust .gitignore file to prevent unnecessary files from being committed.
2. Backend Development & API Services
    - Developed core API routes using FastAPI and Flask, ensuring seamless integration between different services.
    - Designed PostgreSQL database schema with optimized indexes for faster query execution.
    - Integrated JWT authentication and API key-based access control.
    - Implemented structured logging with Python's logging module for better debugging.
3. Frontend Migration to React.js
    - Refactored frontend architecture to React.js for improved flexibility and maintainability.
    - Utilized Redux for centralized state management, improving application performance.
    - Enhanced user interface with modern UI components and better data-fetching logic for real-time analytics.
4. Text Analytics Implementation (NLTK)
    - Developed sentiment analysis module using NLTK to analyze customer feedback.
    - Optimized NLP pipelines by removing stop words and lemmatizing text for better sentiment classification.
    - Ensured accurate visual representation of sentiment distribution through data visualization techniques.
5. Data Visualization Enhancements
    - Implemented Chart.js and Plotly for interactive and customizable charts.
    - Improved dashboard responsiveness, enabling dynamic filtering and real-time data updates.

Repo Check-in of Implementation Completed

The following updates were pushed to the GitHub repository during this progress period:

1. Backend Services
    - api/ → FastAPI and Flask-based services for revenue, weather, and sentiment analysis.
    - models/ → PostgreSQL database models with SQLAlchemy ORM.
    - services/ → Core analytics modules for predictive forecasting and customer sentiment processing.
    - routes/ → Optimized API endpoints with better error handling and authentication.

2. Frontend Components
    - src/components/ → React.js UI components for displaying revenue trends and sentiment analysis.
    - src/pages/Dashboard.js → Updated dashboard with improved filtering options and responsive design.
    - src/utils/api.js → Standardized API request structure with improved error handling.

3. Data Processing Scripts
    - scripts/data_cleaning.py → Automated data validation and preprocessing script.
    - scripts/nlp_analysis.py → Sentiment classification and text processing using NLTK.
    - scripts/revenue_forecasting.py → Implementation of Scikit-learn for predictive analytics.

4. Other Improvements
    - Removed incorrect pull requests that caused project inconsistencies.
    - Improved documentation in README.md to include project setup instructions, API documentation, and contribution guidelines.

Next Steps & Improvements

Here are the main improvements planned for the next phase of the project:

1. Better Graphs and Charts

- Improve the chart design to make it more cohesive, clear and concise.
- Add different types of charts, e.g. scatter plots and heatmaps, to showcase data in a more useful way.
- Make charts more interactive so users can filter and adjust data easily.

2. Downloadable Reports

- Add an option to download reports as PDF, CSV, or Excel so users can save and share their results.
- Allow users to customize reports by choosing specific data they want to include.
- Set up automatic reports that summarize key insights weekly or monthly.

3. Better Prediction Models

- Improve forecasting revenue by making the prediction models more accurate.
- Try out different machine learning techniques to see which one works best.
- Use better ways to check if predictions are correct and adjust them as needed.

4. Live Data Integration

- Connect with live weather data to see how it affects sales and customer trends.
- Fetch real-time data so dashboards update automatically without needing a refresh.
- Bring in other useful data sources like market trends and customer reviews for better insights.