

TEMPLATE DI PROJECT WORK = 3 CFU
min 12 pagine - max 20 pagine
**da compilare e caricare in formato pdf*

Cognome e Nome:	Ayala Shanti
Numero di Matricola:	0312201311
Corso di Studio: ◇ L-5 Filosofia ed Etica ◇ L-22 Scienze Motorie X L-31 Informatica per le Aziende Digitali	Barrare la casella riferita al proprio corso di studio
Tema n:	1
Titolo del tema:	La digitalizzazione dell'impresa
Traccia del PW n:	1.6
Titolo della traccia:	Sviluppo di una dashboard in Python per l'analisi delle prestazioni aziendali nel settore primario
Titolo dell'elaborato:	Ottimizzazione dei Processi operativi nel settore primario tramite Data Engineering e Data Visualization con API Python
PARTE PRIMA – DESCRIZIONE DEL PROCESSO	
Utilizzo delle conoscenze e abilità derivate dal percorso di studio:	Durante la redazione dell'elaborato, ho fatto ampio uso delle conoscenze tecniche e delle abilità di sviluppo del codice acquisite durante il corso di Programmazione, fondamentali per l'implementazione del progetto. Ho applicato concetti di Statistica per garantire la generazione di dati coerenti e ben strutturati, essenziali per l'affidabilità del sistema. La gestione efficiente del database è stata possibile grazie alle competenze apprese in Basi di dati, assicurando una corretta organizzazione e manipolazione delle informazioni. Infine, gli insegnamenti di Ingegneria del software sono stati cruciali per garantire al progetto modularità, separazione delle responsabilità, manutenibilità e scalabilità, oltre a fornire un'appropriata documentazione.
Fasi di lavoro e relativi tempi di implementazione per la predisposizione dell'elaborato:	Le attività svolte durante la redazione dell'elaborato si sono articolate nelle seguenti fasi: <ol style="list-style-type: none"> 1. Valutazione iniziale dei requisiti e progettazione (1 settimana): analisi dei requisiti necessari e definizione dell'architettura progettuale. 2. Definizione del contesto operativo (1 settimana): individuazione dell'ambiente e delle condizioni in cui il progetto si sarebbe sviluppato. 3. Definizione degli obiettivi e partizione del carico di lavoro (2 settimane): suddivisione del progetto in moduli e distribuzione delle responsabilità. 4. Ricerca di dati coerenti al contesto operativo (1 settimana): raccolta di informazioni rilevanti per il progetto. 5. Analisi e scrematura dei dati (2 settimane): valutazione e selezione dei dati per garantirne la coerenza e l'utilizzabilità. 6. Creazione del database, elaborazione e popolazione delle tabelle (1 mese e mezzo): implementazione del database e inserimento dei dati. 7. Elaborazione della dashboard interattiva (1 mese): sviluppo delle funzionalità interattive e visualizzazione dei dati tramite grafici.

	<p>8. Revisioni finali (10 giorni): verifica complessiva del progetto e correzione di eventuali incongruenze.</p> <p>Difficoltà incontrate:</p> <p>Durante la popolazione delle tabelle, la grande quantità di dati generati ha rallentato il processo. La soluzione è stata limitare la generazione dei dati a un arco temporale specifico e ridurre le generazioni a cascata.</p> <p>La coerenza tra i dati generati ha presentato problemi. Questo è stato risolto applicando vincoli (constraint) coerenti tra le varie tabelle del database.</p>
Risorse e strumenti impiegati:	<p>Le risorse e gli strumenti utilizzati per la redazione dell'elaborato sono stati selezionati per garantire coerenza con il contesto del progetto e attendibilità dei dati. Tutti i dati raccolti sono stati organizzati nella cartella "dati_preliminari".</p> <p>Banche dati utilizzate:</p> <ul style="list-style-type: none"> - ISMEA (Istituto di Servizi per il Mercato Agricolo Alimentare) - ISTAT (Istituto Nazionale di Statistica) - Registrazioni meteorologiche di Grosseto, Toscana <p>Queste fonti sono state scelte per la loro affidabilità e per la coerenza geografica con la zona in cui opera l'azienda analizzata. Le risorse sono state reperite attraverso piattaforme ufficiali e database pubblici, ma si sono incontrate alcune difficoltà di accesso ai dati, poiché alcune informazioni non erano accessibili senza autorizzazioni o erano a pagamento. Questi ostacoli sono stati superati cercando soluzioni alternative o richiedendo l'accesso formale ai dati.</p> <p>Strumenti utilizzati:</p> <p>Il progetto ha richiesto l'utilizzo di diversi strumenti software, elencati nel file requirements.txt, tra cui:</p> <ul style="list-style-type: none"> - Dash: per la creazione della dashboard interattiva. - Numpy, Pandas: per la gestione e l'elaborazione dei dati. - Plotly: per la visualizzazione dei dati attraverso grafici. - SQLite3, CSV, Psycpg2: per la gestione del database e dei file. - Random, String, Faker: per la generazione di dati sintetici. <p>Come modello teorico è stato applicato il modello di **Business Intelligence (BI)**, che ha permesso di trasformare i dati grezzi in informazioni utili per decisioni strategiche. Questo approccio ha facilitato l'analisi e la rappresentazione visiva dei dati attraverso strumenti di BI.</p>
PARTE SECONDA – PREDISPOSIZIONE DELL'ELABORATO	
Obiettivi dell'elaborato/progetto/artefatto:	<p>L'elaborato rispecchia gli obiettivi richiesti dalla traccia del Project Work, sviluppando l'intero progetto in linea con quanto indicato.</p> <p>1) È stato integrato l'uso di diverse fonti di dati per alimentare un dashboard interattivo, rispondendo all'esigenza di:</p> <ul style="list-style-type: none"> - Integrare e utilizzare dati provenienti da fonti diverse. - Elaborare e sintetizzare le informazioni in un formato facilmente interpretabile dai decisori aziendali.

	<p>2) È stata creata un'interfaccia utente efficace e reattiva, capace di visualizzare i dati con la possibilità di interazioni dinamiche da parte dell'utente.</p> <p>3) Sono stati applicati principi di visualizzazione dei dati e di user experience, per garantire che il dashboard risultasse accessibile e comprensibile anche per utenti non tecnici. La scelta preliminare dei dati e la gestione delle informazioni generate sono state accuratamente pianificate. Inoltre, la dashboard è stata suddivisa in sezioni per migliorare la comprensione e facilitare la consultazione.</p>
Contestualizzazione:	<p>L'elaborato è stato sviluppato nel contesto teorico-applicativo di un'azienda agricola situata in Toscana, nella provincia di Grosseto. La scelta del territorio è stata motivata dalle caratteristiche del suolo grossetano, noto per la sua fertilità e adatto alla coltivazione di diverse colture, tra cui frutta, ortaggi, cereali, olio d'oliva e vigneti. La presenza di numerose aziende agricole operanti nella zona conferma la versatilità del terreno e ne rende ideale l'analisi e lo sviluppo di soluzioni tecnologiche legate alla gestione e visualizzazione dei dati aziendali.</p>
Descrizione dei principali aspetti progettuali:	<p>Introduzione</p> <p>L'azienda agricola in questione opera nel cuore della Maremma toscana, nella provincia di Grosseto. Inserita nel settore primario, si occupa di un'ampia gamma di coltivazioni che spaziano da cereali a frutta, ortaggi e legumi. Il presente report ha lo scopo di analizzare il contesto in cui opera l'azienda, i suoi processi produttivi, e delineare la struttura organizzativa e le strategie di vendita, fornendo una panoramica completa di come questa realtà agricola gestisca le proprie operazioni.</p> <p>Contesto Aziendale</p> <p>L'azienda è situata nella zona rurale di Grosseto, una delle aree più rinomate per la produzione agricola di qualità, grazie al clima favorevole e alla fertilità dei terreni. Operando nel settore agricolo, la missione dell'azienda è valorizzare le risorse locali, mantenendo un legame stretto con le tradizioni ma, allo stesso tempo, implementando tecnologie innovative per ottimizzare la produzione e la distribuzione dei suoi prodotti.</p> <p>Le Colture</p> <p>L'azienda agricola coltiva una vasta gamma di prodotti che riflettono la diversità delle coltivazioni tipiche della Toscana:</p> <ul style="list-style-type: none"> • Cereali: Grano, orzo, avena e farro rappresentano la base della produzione agricola dell'azienda. Questi cereali vengono utilizzati sia per la produzione di farine, destinate al consumo locale e alle vendite online, sia come mangime per il bestiame. • Frutta: L'azienda produce una selezione variegata di frutta, tra cui albicocche, pesche, ciliegie, mele e fichi. Questi frutti vengono raccolti stagionalmente e venduti sia freschi che trasformati in confetture e succhi. • Ortaggi: Pomodori, zucchine, cipolle, asparagi e carciofi sono alcuni degli ortaggi di punta dell'azienda, apprezzati per la loro qualità e per essere coltivati secondo pratiche sostenibili e rispettose dell'ambiente. • Legumi: Fagioli, ceci e lenticchie rappresentano un'importante produzione per l'azienda, garantendo una varietà nella rotazione delle colture e contribuendo a mantenere la fertilità del terreno. • Uva e olivo: Queste due colture sono di particolare rilevanza per la regione. L'azienda produce uva per vino e olio extravergine di

oliva, due eccellenze che rappresentano la tradizione agricola toscana.

Processo Produttivo

Il processo produttivo dell'azienda è suddiviso in diverse fasi, ognuna delle quali è attentamente pianificata e ottimizzata per garantire la qualità e la sostenibilità delle coltivazioni:

1. **Preparazione del terreno:** Ogni ciclo di produzione inizia con la lavorazione del terreno. Questo processo include l'aratura, la concimazione e la semina, adattati a seconda delle caratteristiche delle colture stagionali.
2. **Coltivazione:** Durante la fase di crescita delle piante, l'azienda implementa pratiche di agricoltura sostenibile, come la rotazione delle colture e l'irrigazione mirata, per ridurre l'uso di risorse naturali e preservare la biodiversità del terreno.
3. **Raccolta:** La raccolta avviene seguendo rigorosi controlli di qualità, assicurando che i prodotti siano colti nel momento giusto di maturazione per mantenere intatte tutte le proprietà nutritive e organolettiche.
4. **Trasformazione:** Alcuni prodotti, come la frutta e gli ortaggi, vengono trasformati in prodotti confezionati (conservate, succhi, salse) all'interno degli impianti aziendali, che sono dotati di attrezzature moderne e conformi agli standard di sicurezza alimentare.
5. **Distribuzione e Vendita:** L'azienda gestisce le vendite sia attraverso un sistema di negozi fisici, situati nelle principali città della Toscana, sia tramite un canale di e-commerce, che consente di raggiungere un pubblico più vasto. Le vendite online includono la spedizione in tutta Italia e in alcuni paesi europei.

Divisione del Lavoro

L'organizzazione interna dell'azienda è strutturata in divisioni specializzate per garantire un'elevata efficienza operativa. Ogni divisione è responsabile di una fase specifica del ciclo produttivo o della gestione aziendale:

- **Divisione agronomica:** Responsabile della gestione delle colture, dalla preparazione del terreno alla raccolta.
- **Divisione trasformazione:** Si occupa della lavorazione e confezionamento dei prodotti.
- **Divisione logistica:** Gestisce la distribuzione dei prodotti ai vari punti vendita e le spedizioni per gli ordini online.
- **Divisione commerciale e marketing:** Si occupa delle relazioni con i clienti, del marketing digitale e della promozione sui canali social e piattaforme di e-commerce.

Conclusione

L'azienda agricola di Grosseto si configura come una realtà solida e ben organizzata, capace di unire tradizione e innovazione per valorizzare i prodotti tipici della Toscana. Grazie a una gestione accurata delle colture e a una strategia di vendita diversificata, l'azienda si posiziona in modo competitivo nel mercato agricolo, continuando a crescere e a soddisfare le esigenze dei suoi clienti, sia locali che internazionali.

Introduzione per la sezione delle tabelle

In questa sezione verranno presentate le tabelle di dati generate per la gestione dell'azienda agricola. Ogni tabella rappresenta un componente chiave del sistema di gestione del database, che permette di organizzare e tracciare le informazioni relative ai prodotti agricoli, al personale, alle vendite e ai processi operativi dell'azienda. Le tabelle, concepite per un utilizzo efficiente, supportano l'azienda nel monitoraggio delle operazioni quotidiane e nell'analisi delle performance produttive.

Di seguito vengono descritte le strutture delle principali tabelle del database, accompagnate da una breve spiegazione del loro utilizzo e del significato di ogni campo.

Tabella 1: Products

Descrizione: Questa tabella contiene l'elenco dei prodotti venduti dall'azienda agricola, con informazioni dettagliate su ciascun prodotto, incluse categoria, prezzo e quantità in magazzino.

Campo	Tipo di dato	Descrizione
ProductID	SERIAL PRIMARY KEY	Identificativo unico del pr
ProductName	VARCHAR(50)	Nome del prodotto
Category	VARCHAR(50)	Categoria del prodotto
Price	DECIMAL(10, 2)	Prezzo del prodotto
StockQuantity	INT	Quantità disponibile in m

Tabella 2: PriceHistory

Descrizione: Memorizza lo storico dei prezzi di ciascun prodotto, tracciando la variazione di prezzo in periodi differenti e indicando se il prodotto è stato dismesso.

Campo	Tipo di dato	Descrizione
PriceHistoryID	SERIAL PRIMARY KEY	Identificativo unico per lo sto
ProductID	INT	Identificativo del prodotto
ProductName	VARCHAR(50)	Nome del prodotto
StartDate	DATE	Data di inizio validità del pre
EndDate	DATE	Data di fine validità del prez
Price	DECIMAL(10, 2)	Prezzo del prodotto in quel

	Discontinued	BOOLEAN	Indica se il prodotto è stato
	<hr/>		
	Tabella 3: CustomerFeedback		
	Descrizione: Raccoglie i feedback dei clienti, incluse valutazioni e commenti su prodotti specifici, fornendo preziose informazioni per migliorare l'offerta.		
	Campo	Tipo di dato	Descrizione
	FeedbackID	SERIAL PRIMARY KEY	Identificativo unico del fe
	Date	DATE	Data in cui è stato dato il
	Rating	INT	Valutazione (da 1 a 5)
	Comment	TEXT	Commento del cliente
	BestProductID	INT	Identificativo del prodotto
	CostumerName	VARCHAR(50)	Nome del cliente
	<hr/>		
	Tabella 4: Employees		
	Descrizione: Contiene le informazioni relative ai dipendenti dell'azienda, incluse le informazioni di contatto e i dettagli lavorativi.		
	Campo	Tipo di dato	Descrizione
	EmployeeID	SERIAL PRIMARY KEY	Identificativo unico del
	FirstName	VARCHAR(50)	Nome del dipendente
	LastName	VARCHAR(50)	Cognome del dipender
	Email	VARCHAR(100)	Indirizzo email
	Phone	VARCHAR(20)	Numero di telefono
	City	VARCHAR(50)	Città di residenza
	State	VARCHAR(50)	Stato di residenza
	ZipCode	VARCHAR(10)	Codice postale
	Country	VARCHAR(50)	Paese
	DateOfBirth	DATE	Data di nascita
	Gender	VARCHAR(10)	Genere

	HireDate	DATE	Data di assunzione
	JobTitle	VARCHAR(100)	Posizione lavorativa
	Salary	DECIMAL(10, 2)	Stipendio
	EmploymentStatus	VARCHAR(50)	Stato occupazionale
	EmergencyContact	VARCHAR(255)	Contatto di emergenza
	Tabella 5: WeatherConditions		
	Descrizione: Questa tabella registra le condizioni meteorologiche giornaliere, fondamentali per la gestione delle coltivazioni.		
	Campo	Tipo di dato	Descrizione
	WeatherID	SERIAL PRIMARY KEY	Identificativo unico del record

	Date	DATE	Data della registrazione meteorologica
	Temperature	DECIMAL(5, 2)	Temperatura media giornaliera
	Humidity	DECIMAL(5, 2)	Umidità media giornaliera
	Precipitation	DECIMAL(5, 2)	Precipitazioni giornaliere
	WindSpeed	DECIMAL(5, 2)	Velocità del vento
	SolarRadiation	DECIMAL(5, 2)	Radiazione solare giornaliera
	SoilMoisture	DECIMAL(5, 2)	Umidità del suolo
	WeatherDescription	VARCHAR(100)	Descrizione del tempo
	Tabella 6: Transactions		

	Descrizione: Questa tabella contiene le transazioni di vendita, tracciando ogni operazione effettuata dai clienti.		
	Campo	Tipo di dato	Descrizione
	TransactionID	SERIAL PRIMARY KEY	Identificativo unico della transazione
	Date	DATE	Data della transazione
	TotalAmount	DECIMAL(10, 2)	Importo totale

	PaymentMethod	VARCHAR(50)	Metodo di pagamento utilizzato

	TransactionStatus	VARCHAR(50)	Stato della transazione
	Tabella 7: Sales		
	Descrizione: Contiene le informazioni dettagliate sulle vendite effettuate, incluse quantità e canali di vendita.		
	Campo	Tipo di dato	Descrizione
	SaleID	SERIAL PRIMARY KEY	Identificativo unico della vendita
	Date	DATE	Data della vendita
	ProductID	INT	Identificativo del prodotto venduto
	TransactionID	INT	Identificativo della transazione
	Quantity	INT	Quantità venduta
	UnitPrice	DECIMAL(10, 2)	Prezzo per unità
	TotalPrice	DECIMAL(10, 2)	Prezzo totale della vendita
	Discount	DECIMAL(5, 2)	Sconto applicato
	SalesChannel	VARCHAR(50)	Canale di vendita (online, fisico)
	Tabella 8: WorkHours		
	Descrizione: Registra le ore di lavoro dei dipendenti, compresi straordinari e tipologia di lavoro svolto.		
	Campo	Tipo di dato	Descrizione
	WorkHourID	SERIAL PRIMARY KEY	Identificativo unico per le ore di lavoro
	EmployeeID	INT	Identificativo del dipendente
	Date	DATE	Data della prestazione lavorativa
	HoursWorked	DECIMAL(5, 2)	Ore lavorate
	OvertimeHours	DECIMAL(5, 2)	Ore di straordinario
	WorkType	VARCHAR(50)	Tipo di lavoro
	Tabella 9: Crops		

Descrizione: La tabella Crops contiene i dettagli relativi alle coltivazioni, incluse le informazioni sulle stagioni di impianto e raccolta, nonché le condizioni ottimali per la crescita.

Campo	Tipo di dato	Descrizione
CropID	SERIAL PRIMARY KEY	Identificativo unico della coltura
CropName	VARCHAR(100)	Nome comune della coltura
ScientificName	VARCHAR(100)	Nome scientifico della coltura
Description	TEXT	Descrizione della coltura
PlantingSeason	VARCHAR(50)	Stagione di impianto
HarvestSeason	VARCHAR(50)	Stagione di raccolta
GrowthDuration	INT	Durata della crescita in giorni
SoilType	VARCHAR(50)	Tipo di terreno richiesto
OptimalTemperatureMin	DECIMAL(5, 2)	Temperatura minima ottimale per la crescita
OptimalTemperatureMax	DECIMAL(5, 2)	Temperatura massima ottimale per la crescita
OptimalHumidityMin	DECIMAL(5, 2)	Umidità minima ottimale
OptimalHumidityMax	DECIMAL(5, 2)	Umidità massima ottimale
OptimalPrecipitationMin	DECIMAL(5, 2)	Precipitazione minima ottimale
OptimalPrecipitationMax	DECIMAL(5, 2)	Precipitazione massima ottimale
OptimalSunlightHours	DECIMAL(5, 2)	Ore di luce solare ottimale
FertilizersRequired	TEXT	Fertilizzanti necessari
PestsAndDiseases	TEXT	Parassiti e malattie che possono colpire la coltura
WateringNeeds	TEXT	Bisogni di irrigazione
Replant	BOOLEAN	Indica se la coltura deve essere ripiantata

Tabella 10: Plantings

Descrizione: Traccia le attività di impianto e raccolta delle coltivazioni, insieme alle informazioni sulla fertilizzazione e note operative.

Campo	Tipo di dato	Descrizione
PlantingID	SERIAL PRIMARY KEY	Identificativo unico dell'i
CropID	INT	Identificativo della coltiv
PlantingDate	DATE	Data di impianto
HarvestDate	DATE	Data prevista di raccolta
AreaDedicated	DECIMAL(10, 2)	Area dedicata all'implan
FertilizersUsed	TEXT	Fertilizzanti usati
Notes	TEXT	Note aggiuntive

Tabella 11: Costs

Descrizione: Memorizza i costi legati alla gestione aziendale, con particolare focus su risorse come acqua, energia e fertilizzanti.

Campo	Tipo di dato	Descrizione
CostsID	SERIAL PRIMARY KEY	Identificativo unico pe
Date	DATE	Data di registrazione
WaterCost	DECIMAL(10, 2)	Costo dell'acqua
EnergyCost	DECIMAL(10, 2)	Costo dell'energia
AvgFertilizerCost	DECIMAL(10, 2)	Costo medio dei ferti

Conclusione della Sezione delle Tabelle

Le tabelle descritte in questa sezione forniscono una solida infrastruttura per gestire i dati chiave dell'azienda agricola. Ogni tabella è progettata per tracciare informazioni specifiche, che vanno dai prodotti e coltivazioni, alle condizioni meteorologiche, fino al feedback dei clienti e alla gestione del personale. La loro implementazione consente una gestione più efficiente delle risorse, delle operazioni di campo e delle interazioni commerciali, contribuendo a una visione complessiva e dettagliata delle attività aziendali.

Attraverso l'uso di queste strutture dati, l'azienda può prendere decisioni informate, ottimizzare i propri processi produttivi e migliorare continuamente la qualità dei prodotti offerti sul mercato.

Codice Python per la Dashboard

Introduzione

Per lo sviluppo della dashboard, è stato realizzato un progetto Python strutturato che segue il modello di un'applicazione web interattiva basata su Flask. L'obiettivo principale del progetto è fornire una piattaforma per l'analisi dei dati agricoli e meteorologici, la visualizzazione delle statistiche e la gestione delle informazioni aziendali. La struttura del progetto è organizzata in modo tale da separare chiaramente la logica di business, la gestione del database, la generazione di dati e la presentazione visiva, seguendo buone pratiche di sviluppo software.

Struttura del Progetto

Il codice della dashboard si articola in una serie di file e directory organizzati come segue:

```
track_1.6_dashboard/
├── code/
│   ├── main.py           # Main entry point
│   ├── models/           # Database models, not implemented
│   │   ├── user.py       # Example model for user data
│   │   └── data.py       # Model to manage your analytical data
│   ├── services/         # Business logic and data processing
│   │   ├── data_analysis.py # Statics generation service
│   │   ├── queries.py     # Queries SQL
│   │   ├── data_generation.py # Data generation service
│   │   └── static_generation.py # Static generation service
│   ├── static/           # Static files (CSS, JavaScript, images)
│   ├── assest/           # Templates for rendering
│   │   └── styles.css    # styles for index
│   ├── requirements.txt  # List of dependencies
│   └── populate_clean.py # Script to test db cleaning and db
├── population
│   ├── db_handler.py     # Database connection and operations
│   └── cleanup.py        # Cleanup logic on exit
├── dati_preliminari/     # Folder for preliminary data analysis
│   ├── ISMEA             # CSV folder from ISMEA
│   ├── ISTAT             # CSV folder from ISTAT
│   └── Toscana_meteo     # CSV folder from from local weather
├── farm_connection.session.sql # Sql for db generation
└── Traccia_dashboard.pdf  # Projectwork track
```

La struttura è organizzata in modo modulare:

- **Main entry point:** `main.py` e `app.py` contengono il codice necessario per l'avvio dell'applicazione.

- **Modelli:** All'interno della directory `models/`, troviamo `user.py` per la gestione dei dati utente e `data.py` per l'analisi dei dati aziendali. Questi modelli definiscono le strutture dati che vengono utilizzate dall'applicazione.
- **Servizi:** La directory `services/` racchiude la logica di business, con file come `data_analysis.py` per l'elaborazione dei dati e la generazione delle statistiche, e `queries.py` per l'esecuzione delle interrogazioni SQL. `data_generation.py` e `static_generation.py` forniscono servizi per la generazione di dati e contenuti statici, rispettivamente.
- **Assest Static Files:** Le cartelle `templates/` e `static/` contengono i file di stile e i template per la visualizzazione dell'interfaccia grafica, migliorando l'esperienza utente.
- **Preliminary Data:** La cartella `dati_preliminari/` contiene i dati provenienti da fonti come ISMEA, ISTAT e meteo locali, fondamentali per l'analisi iniziale che alimenta le dashboard.

Questa struttura modulare e ben organizzata facilita la manutenzione del codice, permette l'espandibilità dell'applicazione e consente di gestire con facilità dati complessi e processi di analisi.

Per consultare il codice Python sviluppato per la dashboard, le sezioni principali sono:

- **data_generation.py:** responsabile della generazione dei dati simulati utilizzati nel sistema.
- **static_generation.py:** incaricato della creazione di file CSV da analizzare e utilizzare nelle varie fasi del progetto.
- **main.py:** il file principale che gestisce l'interfaccia della dashboard e ne coordina le funzionalità operative.

Resoconto del Processo Seguito per lo Sviluppo del Codice Python

1. Comprensione delle Richieste

Il processo di sviluppo del codice Python per la dashboard è iniziato con una comprensione dettagliata delle richieste del progetto. L'obiettivo era creare uno strumento che permettesse l'analisi dei dati agricoli e la gestione delle risorse aziendali in modo efficiente e interattivo. Il primo passo è stato raccogliere le esigenze dell'azienda, comprendere i tipi di dati da trattare e identificare i requisiti specifici dell'applicazione.

2. Valutazione dei Requisiti e Architettura del Progetto

Una volta compresi i requisiti, è stata condotta una valutazione primaria per identificare le risorse necessarie. I requisiti principali includevano la necessità di gestire dati relativi a colture, vendite, condizioni meteorologiche e feedback dei clienti. Da questa analisi è emersa una bozza architetturale modulare, con una separazione tra logica di business, gestione dei dati e presentazione visiva. Si è deciso di utilizzare un'architettura basata su Flask, con l'uso di moduli separati per la gestione del database, l'analisi dei dati e la generazione di contenuti statici.

3. Definizione degli Obiettivi e Divisione del Carico di Lavoro

Una volta stabilita l'architettura del progetto, gli obiettivi principali sono stati definiti con precisione: creazione di una dashboard per la

visualizzazione dei prodotti, il monitoraggio delle condizioni ambientali, l'analisi delle vendite e delle colture, e la gestione dei costi operativi. Il carico di lavoro è stato suddiviso in fasi distinte: la raccolta dei dati, la creazione del database, la generazione dei dati, e infine lo sviluppo dell'interfaccia utente.

4. Raccolta di Dati Coerenti e Scrematura

Per garantire un'analisi accurata e pertinente, è stato necessario raccogliere dati coerenti con il contesto operativo dell'azienda agricola. Sono stati utilizzati dataset provenienti da fonti affidabili, come ISMEA, ISTAT e dati meteorologici locali (archiviati nella cartella `dati_preliminari/`). Questi dati sono stati sottoposti a una scrematura che ha incluso visualizzazione, selezione e riorganizzazione per renderli utilizzabili nel contesto del progetto. Attraverso questa fase, sono stati eliminati i dati non pertinenti o incoerenti.

5. Scelta della Struttura del Database e Creazione

Dopo aver compreso la natura dei dati da trattare, è stato progettato un database relazionale per gestire le varie entità, come prodotti, colture, vendite, dipendenti e condizioni meteorologiche. La struttura del database è stata creata in modo che ogni tabella potesse mantenere integrità referenziale e permettere interrogazioni efficienti. Sono stati implementati vincoli e relazioni tra le tabelle per garantire la coerenza dei dati.

6. Popolazione del Database e Modularizzazione del Codice

Una volta creata la struttura del database, si è proceduto alla sua popolazione utilizzando script Python per generare dati realistici in modo modulare. Per questo, sono state utilizzate librerie come `csv` per la gestione di file, `random` e `string` per la generazione di valori casuali, `pandas` e `numpy` per l'elaborazione e manipolazione dei dati, `faker` (locale `'it_IT'`) per la simulazione di dati fittizi (ad esempio, nomi di dipendenti e clienti), `datetime` per gestire le date e `decimal` per valori monetari.

Queste librerie hanno permesso di generare dati coerenti tra loro, e strumenti statistici sono stati utilizzati per garantire che i dati fossero rappresentativi della realtà aziendale. Ad esempio, la generazione dei prezzi e delle quantità ha seguito distribuzioni plausibili, tenendo conto di eventuali errori casuali e garantendo l'accuratezza dei dati.

7. Sviluppo della Dashboard

La fase successiva ha riguardato lo sviluppo della dashboard vera e propria. Il codice Python è stato scritto per gestire le interazioni con il database, con file dedicati per le query SQL e il fetch dei dati.

L'implementazione della dashboard è stata suddivisa in sezioni per facilitare la gestione dei vari obiettivi. Le sezioni principali della dashboard includono:

- **Visualizzazione dei Prodotti:** permette di monitorare lo stock e i prezzi dei prodotti.
- **Monitoraggio Ambientale:** presenta dati sulle condizioni meteorologiche locali.
- **Analisi delle Vendite:** offre un'analisi dettagliata delle vendite e delle transazioni.

- **Analisi delle Colture:** visualizza dati sulle coltivazioni e le condizioni ottimali di crescita.
- **Analisi dei Costi:** fornisce una visione complessiva delle spese aziendali.

Le tecnologie utilizzate includono dash, plotly e pandas, mentre la visualizzazione è stata personalizzata con l'uso di fogli di stile CSS, migliorando l'estetica e la fruibilità della dashboard. La generazione dinamica dei grafici ha facilitato la presentazione visiva delle informazioni aziendali.

Conclusione del Resoconto

Lo sviluppo del codice Python ha seguito un approccio iterativo e modulare, garantendo flessibilità e adattabilità ai futuri requisiti. La gestione accurata dei dati e l'integrazione delle tecnologie di analisi hanno permesso la creazione di una dashboard funzionale, capace di offrire all'azienda una visione completa e dettagliata delle sue operazioni.

Eventuali Sviluppi Futuri

Il progetto attuale rappresenta una solida base per la gestione e l'analisi dei dati aziendali, ma ci sono molteplici aree che possono essere migliorate o ampliate in futuro. Ecco alcuni potenziali sviluppi futuri:

- **Integrazione di Funzioni di Machine Learning:** L'aggiunta di algoritmi di machine learning per prevedere le rese delle colture in base ai dati storici e alle condizioni ambientali potrebbe fornire all'azienda strumenti avanzati per ottimizzare la produzione. Ad esempio, tecniche di regressione e reti neurali potrebbero essere utilizzate per modellare e prevedere i cicli di crescita delle colture.
- **Monitoraggio Ambientale in Tempo Reale con IoT:** Un ulteriore passo potrebbe essere l'integrazione di sensori IoT per il monitoraggio in tempo reale di variabili ambientali come temperatura, umidità e qualità del suolo. Questi dati, raccolti direttamente sul campo, potrebbero essere analizzati in tempo reale per consentire una risposta rapida a cambiamenti ambientali critici.
- **Gestione delle Risorse Umane:** L'aggiunta di una sezione dedicata alla gestione delle risorse umane potrebbe migliorare la visione complessiva delle operazioni aziendali. Questa sezione potrebbe includere funzionalità come la pianificazione delle ore di lavoro, il monitoraggio delle prestazioni dei dipendenti e la gestione delle assunzioni e delle emergenze.
- **Espansione dell'Accesso Multi-Utente:** Un'ulteriore evoluzione potrebbe essere l'implementazione di un sistema avanzato di autenticazione e autorizzazione, che permetta a più utenti di accedere alla dashboard con diversi livelli di permessi. Questo sarebbe particolarmente utile per una gestione condivisa, con ruoli definiti per amministratori, tecnici agricoli e personale delle vendite.

Conclusione del Report

	<p>In conclusione, questo progetto ha permesso di affrontare con successo le sfide legate alla gestione e all'analisi dei dati in un contesto agricolo. L'applicazione sviluppata, attraverso la sua dashboard, consente all'azienda di visualizzare in modo intuitivo i dati aziendali e di prendere decisioni informate. L'adozione di un'architettura modulare ha reso il codice scalabile e facilmente mantenibile, assicurando la possibilità di espansione futura. Il report riflette il percorso metodologico e tecnico seguito, dimostrando come la tecnologia possa essere utilizzata per ottimizzare processi agricoli e commerciali.</p>
Campi di applicazione:	<p>L'elaborato progettuale trova applicazione principalmente nei settori primari, con un focus particolare sul settore agricolo. I vantaggi derivanti dalla sua applicazione includono un miglioramento nella comprensione e gestione delle attività aziendali, grazie all'integrazione di strumenti innovativi. Questo consente di ottimizzare le tecniche utilizzate, garantendo un miglioramento della produzione e della rendita aziendale. Inoltre, il progetto favorisce una visualizzazione più chiara e dettagliata dei costi e delle uscite, agevolando le decisioni strategiche e aumentando l'efficienza operativa.</p>
Valutazione dei risultati (potenzialità e criticità):	<p>L'elaborato offre ottime potenzialità in termini di riutilizzabilità e scalabilità, rendendolo facilmente adattabile a contesti diversi e a progetti futuri. Tuttavia, ci sono alcune limitazioni che non compromettono la validità complessiva del progetto. La parte di testabilità non è stata gestita in questa fase, ma può essere facilmente integrata in futuro. Inoltre, i dati attualmente utilizzati sono fittizi, e in un'applicazione reale la generazione automatica non sarebbe necessaria, poiché si utilizzerebbero dati reali. Infine, la gestione dell'autenticazione e degli utenti non è stata implementata, poiché il progetto lavora in locale, ma tale funzionalità potrebbe essere aggiunta in un contesto di utilizzo su larga scala o in rete.</p>