Here's a detailed project explanation that you can get a knowledge on my project

**Sales Data Entry Web Application - Project Report**

## Motivation Behind This Project

After graduating two years ago, I faced challenges in securing a job despite my interest in **data analytics and software development**. To bridge the gap between theoretical knowledge and real-world applications, I decided to build this **Sales Data Entry** project.

This project serves as a **learning milestone** to strengthen my skills in **Flask, SQL, Power BI, and web development** while also demonstrating my ability to develop functional data-driven applications. My goal is to showcase this project as a **portfolio piece** to potential employers and contribute to real-world business solutions.

**1. Project Overview**

The **Sales Data Entry Web Application** is a simple yet effective tool for recording and managing sales transactions. This project is built using **Flask (Python), MySQL, HTML, CSS**. Users can enter sales details such as **date, product name, quantity, and price**, which are stored in a structured database.

**Key Highlights**

* **User-friendly Interface**: A clean and simple form to enter sales data.
* **Database Connectivity**: Sales records are stored securely in MySQL.
* **Flask-powered Backend**: Handles form submissions and database interactions.
* **Scalability**: Can be extended with features like authentication, dashboards, and data analytics.

**2. Technologies Used**

The project leverages the following technologies:

| **Technology** | **Purpose** |
| --- | --- |
| **Flask** (Python) | Backend framework to handle form submissions and database connections |
| **MySQL** | Database to store sales records |
| **HTML, CSS** | Frontend design for the user interface |
| **VS Code** | IDE for development |
| **Git & GitHub** | Version control and project hosting |

**3. Project Structure**

The project is structured as follows:

/sales-data-entry

/templates

index.html --> Frontend form for data entry

/static

style.css --> (Optional) For styling

app.py --> Main Flask application

requirements.txt --> Python dependencies

README.md --> Project documentation

config.py --> (Optional) Database settings

**4. Features**

**Frontend (HTML, CSS)**

* **User-friendly Sales Form**: Allows users to input sales details easily.

**Backend (Flask - Python)**

* **Handles Form Submission**: The Flask application receives and processes the sales data.
* **Database Storage**: Data is securely stored in a MySQL database.
* **Real-time Data Entry**: Users can submit data, which is immediately added to the database.

**5. Database Schema**

The **MySQL database** consists of a single table named **sales**, with the following structure:

|  |
| --- |
| Sql  CREATE DATABASE sales\_db; |
| USE sales\_db; |
|  |
| CREATE TABLE sales ( |
| id INT AUTO\_INCREMENT PRIMARY KEY, |
| date DATE NOT NULL, |
| product\_name VARCHAR(255) NOT NULL, |
| quantity INT NOT NULL, |
| price DECIMAL(10,2) NOT NULL)); |

**Each record stores:**

* **Date** of the sale
* **Product name**
* **Quantity** sold
* **Price** of the product

**6. Installation & Setup Guide**

**Prerequisites**

Ensure you have the following installed:  
✅ **Python (3.7 or higher)**  
✅ **MySQL Server**  
✅ **VS Code (or any code editor)**

**Steps to Run the Project**

**Step 1: Clone the Repository**

git clone https://github.com/yourusername/sales-data-entry.git

cd sales-data-entry

**Step 2: Set Up a Virtual Environment (Optional, but Recommended)**

python -m venv venv

source venv/bin/activate # (For macOS/Linux)

venv\Scripts\activate # (For Windows)

**Step 3: Install Dependencies**

pip install -r requirements.txt

**Step 4: Configure the Database**

1. Open MySQL and execute the database schema commands.
2. Ensure Flask is connected to the database in app.py.

**Step 5: Run the Flask Application**

python app.py

**Step 6: Open the Web Application**

Go to http://127.0.0.1:5000/ in your browser.

**7. Code Explanation**

**index.html (Frontend)**

* A simple HTML form that collects sales data.

**app.py (Backend)**

* **Handles HTTP Requests**: Receives form data and processes it.
* **Connects to MySQL**: Stores submitted data securely.
* **Runs a Flask Server**: Hosts the application locally.

**requirements.txt**

Contains a list of required Python packages, including Flask and MySQL libraries.

**8. Future Enhancements**

🚀 **Dashboard Integration**: Use **Power BI** to visualize sales trends.  
🔒 **User Authentication**: Secure access with login credentials.  
📊 **Data Export**: Export sales records to **Excel or CSV** for further analysis.

**9. Author**

👨‍💻 **Naveen Kumar Narasapuram**  
📌 **Aspiring Data Analyst** | Seeking Opportunities in Power BI, SQL, and Python

**10. License (MIT License)**

This project is licensed under the **MIT License**, meaning it can be freely used, modified, and distributed.

MIT License

Copyright (c) 2024 Naveen Kumar Narasapuram

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software and associated documentation files (the "Software"), to deal

in the Software without restriction, including without limitation the rights

to use, copy, modify, merge, publish, distribute, sublicense, and/or sell

copies of the Software, and to permit persons to whom the Software is

furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all

copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER

LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

**1. How to Upload to GitHub**

1. **Initialize a Git repository**

git init

git add .

git commit -m "Initial commit"

1. **Add the remote repository**

git remote add origin https://github.com/yourusername/sales-data-entry.git

git branch -M main

1. **Push to GitHub**

git push -u origin main

**Conclusion**

This **Sales Data Entry Web Application** provides a foundation for businesses to manage sales data effectively. With a simple interface, structured database, and powerful backend, it can be further enhanced into a full-fledged sales management system.