Introduction

- Hello my name is Santosh Karche. I'm from Maharashtra. I have Completed my Bachelor of Engineering from Shivajiuniversity Kolhapur.
- Currently I am working With **Garware Technical** as a python backend developer. Main work I am doing in my organization related to backend technology. I have good Knowledge of python programming, Django and Flask Framework and MySQL Database.
  .
- Currently I am working on Supply chain management system which comes under **oil and gas domain.**
- This project is entitled as oil and gas station warehouse and distribution management system. This is
- web-based application developed in Python and Django Framework with MySQL Database.
- The goal of this project provides an online and automated platform for Oil and Gas Station Inventory management to manage their fuel inventory and to increase sales within a short period of time. Therefore, an extension of the focus is on customer satisfaction. The system stores records of fuel stock it manages and notify the capacity of storage tanks at warehouse.
-
- While working on this project as a python backend developer, we have developed this software by using python, Django framework, Django rest framework for backend and MySQL database.
- While working on this project we used various python modules like datetime, app scheduler, math, pickle, JSON, also, we have **use Authentication and authorization mechanism, Django models**, Django serilaizers, Djangoforms, Django URLs and performing CURD operations by using Django ORM queries.
- To design and Develop dashboards we used the Python concepts such as List Comprehensions, decorators, generators, lambda, map, reduce and OOPs concepts
- I am also familiar with popular development tools like GIT, Jenkins Docker and AWS services.
-
- I was a part of this software where we have **to create rest Api** which is going to meet the requirement ofclient.
- In this project we have implemented python API query to fetch data form database and update the tables and support existing application
- also working on agile tool which is Jira it involves the day-to-day activities like Scrum meeting/Scrum call, sprint meeting.

ROLES AND RESPONSIBILITIES:

- === So as per the PCR Request to add the oil and gas service in application.
- === and developing reports and revenue for every open top container included in client list.
- === Reports for every container is created on Daily, Monthly and Yearly Basis and also use the multiple components and performing the mathematical formula for calculation.
- === I was a part of this software where I am responsible to **create rest Api** which is going to meet the requirement of client.
- Responsible for writing optimized code, debugging and resolving bugs with troubleshooting.
- Create and write result reports in different formats like txt, xls, and JSON.
- Implemented modules for registration of the products, placing the order and tracking.
- Managed, developed and designed a dashboard control panel using basic and advance Python

- PyCharm IDE for developing the code and performing the unit test.
- concepts like List Comprehensions, decorators, generators, OOPS Concept.
- Used Exposure of working in front-end technologies to develop and improve user interface and integrate the API's.
- Responsible for managing code repository using GITHUB.
- I also involved in logical enhancement related to type of web application, should follow the taxes rules, margin, transit charges applied each-type of fuel type based on government rules of client country.
-

Roles and responsibiliies:

=== I have attended Daily stand-up meetings (Scrum), Estimation meetings and peer review meetings to analyze requirements for each story card in a sprint.
=== I have Used JIRA for Task and defect tracking.
=== I have Created database schemas for newly generated data.
=== I have created Python APIs to Query the database and update the tables.
=== I have Sorted the extracted data in MySQL database for further querying and analysis.
=== I have Worked on the creation of REST APIs with Django Rest framework.
=== Keeping track of the changes in the versioning using GIT.
=== Creating an API for data classification with logs.
=== Coordinating with the team to meet the company requirements.
=== Created code to provide a flexible environment to send the reports as per requirement.

Summary On project:

- Currently I am working on Supply chain management system which comes under OIL AND GAS domain.
- Now I am working with **Cabot oil & Gas crop** which is American independent natural gas producer with headquarters in Houston, Texas.
- This includes handling the procurement of raw materials, specifically the Active Pharmaceutical Ingredients (**APIs**) used in the **production of medicines**
- By integrating this vendor management system, we aim to optimize the manufacturing workflow, ensuring a seamless supply chain for the pharmaceutical industry.
- To manage purchase orders and invoices, making it easier to handle the financial aspects of the vendor management system.
- Inventory management features to keep track of available stock, stock levels, and automatically update stockquantities based on order fulfilment.

In Technical point of view

- This application requires the users to log in with their registered user credentials in order to gain access to the features and functionalities of the system.
- I am working on this project as python developer, we have developed this software by using python, Django framework, and MySQL database for backend and for frontend we have used HTML, CSS and Bootstrap.
- This application requires the users to log in with their registered user credentials in order to gain access to the features and functionalities of the system.
- After successful login in the system, the user can see Inventory Summary which consist of fuel type and its stock.
- we offered various dashboards such as executive dashboard for the management leads that offered key performance indicators and business forecasts for decision making (graphical representation). we used mat plot library used for graphical representation.
- User Dashboard is provided to facilitate real-time view of the stock positions
- in various tanks with respect to available stock for sales and blocked stock (sold but not delivered yet) in all the stock categories and for stock in transit.
- After successful login in the system, the user can see Inventory Summary
- User Dashboard is provided to facilitate real-time view of the stock positions in various tanks with respect to available.

- we are building a RESTful API for order management; you can use Django Rest Framework to handle order-related CRUD operations and track orders through API endpoints.
- We have to create custom database models in Django to represent orders and their statuses. By defining appropriate fields and methods, you can track orders' status changes and update the database accordingly.
- **Some API**
- 1)ZOLPAN D 2) Anti-OD 3) Advance-625 4) Ethixim- O 5) Ace-SP 6) Emlox-DC 7) Emiaktion-500

We used third party application

1) Map Box : It's used to track the current locations of vehicle
2) **Twilio**: Twilio is a cloud communications platform that provides a set of APIs. It's used to voice, messaging and video.
3) **Stripe**: Stripe is a popular payment processing api and manage transactions.
4) **SendGrid:** which is used to **email delivery** and marketing platform.
5) **Rest app Scheduler**: We scheduled a Program for Sending notification of daily status report

What challenges did you face during the project and how did you overcome them?

- One of the main challenges we faced was integrating the "**Django-ordering API"** for order tracking.
- We are able to overcome this challenge by working closely with our team and researching different solutions until we found the right one.

Introduction

- We had to ensure that the API was implemented correctly and that it worked seamlessly with the rest of the system.
- To overcome this challenge, we collaborated closely with the frontend developer to ensure that the API was integrated properly.

Client requirements

1. Vendor Communication and Collaboration:

Requirement:

• The client requires a platform for seamless communication and collaboration with vendors, including sharing documents, requesting quotes, and resolving issues.

Solution:

• Create a communication portal or messaging system that facilitates real-time communication between the client and vendors.

• Enable document sharing and version control to ensure all parties have access to the latest information.

• Implement a ticketing or issue resolution system to address and track vendor-related issues or inquiries.

2. Compliance and Risk Management:

Requirement:

• The client needs a system to ensure vendor compliance with legal, regulatory, and internal policies, as well as to manage vendor risks.

Solution:

• Develop a compliance management module that tracks vendor certifications, insurance coverage, and adherence to legal and regulatory requirements.

• Implement risk assessment and mitigation processes to identify and manage vendor-related risks.

• Provide reporting and alerts for non-compliance or high-risk vendors.

Q. Where we can used decorator in your project?

In your oil and gas warehouse management system project, decorators can be used to add additional functionality to your code without changing the original function's code. Some ways you might use decorators in your project include:

1. **Authentication**: You can use a decorator to check if a user is authenticated before allowing them to access certain parts of the system.

2. **Logging**: You can use a decorator to log information about each function call, such as the function name, input parameters, and output.

3. **Caching**: You can use a decorator to cache the results of expensive or time-consuming functions, so that subsequent calls with the same parameters can be returned from cache without having to re-run the functions.

4. **Error handling**: You can use a decorator to catch and handle exceptions thrown by a function, or to add custom error messages or logging.

Q. Where we can used Generators in your project?

- Generators in Python are a useful tool for handling large datasets and processing them efficiently. In an oil and gas warehouse management system project, you might use a generator to process a large dataset of oil and gas inventory records.
- For example, let's say you have a CSV file containing information about all the items in your vendor, including their names, mobile number, and locations. You could use a generator to read the CSV file line by line, and process each one record at a time, rather than loading the entire dataset into memory at once.

Q. Where we used multithreading in your project?

Multithreading can be used in various scenarios to improve the performance and efficiency of the application. Such as **Data Processing, File Uploads/Downloads, Email Notifications**, Concurrent Requests, Background Tasks

Q. Where we can used pandas in your project?

1) **Data Analysis and Reporting**: Pandas provides efficient data structures and functions for data analysis, manipulation in a supply chain management system, you can use Pandas to perform various analytical tasks, such as generating reports, calculating statistics, aggregating data, and extracting insights from large data sets.

2) **Data Import and Cleaning**: Supply chain management systems often deal with data from multiple sources, such as suppliers, inventory systems, and logistics providers. Pandas can be used to import data from different file formats (CSV, Excel, etc.), clean and preprocess the data (removing duplicates, handling missing values, standardizing formats), and transform it into a suitable format for further processing or storage.

3) **Data Integration**: In a supply chain management system, you may need to integrate data from different sources or merge datasets to gain a holistic view of the supply chain. Pandas provides tools for joining, merging, and concatenating datasets based on common attributes or keys. This can help you combine information from different systems or sources into a unified representation.

Introduction

Work On Agile methodology:

Currently We work in Agile methodology. I'm working on web development for user stories assigned to me during the sprints, working on python, Django framework and for database using MySQL, after completion of specific the user story and its code review, I send pull request for the code merge and then after successful testing by testers, deployment is Done.

Daily Routine:

- Currently We work on Agile methodology, we were Assigned with user Stories, We Work in Sprint And task are Assigned by Jira tool.
- Each Sprint is of two weeks and two Sprints are one release Cycle.
- Then We Check for tickets raised by team lead and need to acknowledge.
- After that working start according to tickets raised.
- Then we are Attending daily status meeting called Scrum Meeting conducted by scrum master every day for 15 minutes.
- In that meeting we are discussed. If there are any requirement missing, or developer is not able to do something, or tester is not able to complete something the scrum master will take of it.
- Every team member should say the status, like what are the task they have completed yesterday,
- what is the task they have planned today and are there any implements in doing work.

Introduction

Query used in project

1. Querying All Vendors- vendors = Vendor.objects.all ()
2. Querying a Single Vendor by ID- vendor = Vendor.objects.get(id=vendor_id)
3. Filtering Vendors based on Conditions- vendors = Vendor. Objects. filter(category='desired_category')
4. Ordering Vendors- vendors = Vendor. Objects. order_by('name')
5. Count the number of vendors- vendor_count = Vendor.objects. count ()
6. Prefetching related products for all vendors- vendors = Vendor.objects. prefetch related('products')
7. Updating Vendor Information- vendor = Vendor.objects.get(id=vendor_id)

   vendor.name = 'New Name'

   vendor. save ()
8. Deleting a Vendor- vendor = Vendor.objects.get(id=vendor_id)

   vendor. delete ()