PRIYANKA CARPENTER

priyankacarpenter30@gmail.com
+9111053655
https://www.linkedin.com/in/priyanka-carpenter/
https://github.com/Priya91110/

PROFILE SUMMARY

Skilled Python Developer with extensive experience in developing and maintaining dynamic web applications. Proficient in creating systems such as Invoice Management and web scraping with BeautifulSoup. Experienced in developing robust applications and working on data preprocessing and model. Developed a Flask-based chatbot using OpenAI's GPT-3.5-turbo and a tool extract images from video

SKILLS

Python	Pillow	Web Development and Frameworks	Databases	Deployment
pandas	Poetry	Django (MVT), DRF	MySQL	CPanel
Beautiful Soup	Opencv	FastAPI	PostgreSQL	AWS Instance
requests	pytest	API Development	SQLite 3	Docker
	JSON	Flask	sqlalchemy	GitHub

EDUCATION

Master of Computer Application	Scope Global Skill University	2024- pursuing
Bachelor of Computer Application	Rabindra Nath Tagore University	2017-2020 CGPA 7.9
CISM Course	IANT institude	2019-2020 Grade A
Higher and Secondary Schooling	Morning Star School	2014-2017 75% (Higher) and 79.99% (Secondary)

WORK EXPERIENCE

Technoque Software Solution **1 Nov 2022 - July 2023**

Jr. Python developer

Developed and maintained Python-based web applications using frameworks Django.

Fetching API data, Created dataframe of api data, Create csv file from processed dataframe and insert into database using pscycopg connector.
API developed using fastapi and django rest work.

Tech Simba Pvt. Ltd. **1 Sep 2023 - 12 Dec 2023**

Python Develeoper

Developed a Django web application for Invoice management

Incorporating models to create and delete invoices with auto-generated unique invoice numbers. The application included functionality for downloading invoices and generating detailed reports using the ReportLab library, as well as converting HTML to PDF with the PISA library. It managed CRUD (Create, Read, Update, Delete) operations for invoices, and featured forms for invoice submission and saving. The admin panel allowed for user management by superusers, including the creation and management of user accounts. Additionally, authentication was implemented to ensure secure access to the application.

Web Scraping for Upcoming Business Events

Scraped websites for crucial details about upcoming logistics business events, including email host names, organization names, locations, and event names.Employed BeautifulSoup to extract information from various websites providing event details. Saved extracted data into CSV files and used Pandas to handle missing values, reformat dates, and remove duplicate entries. Utilized SQLAlchemy to connect to a PostgreSQL database, where the cleaned and processed data was stored.

Ybi Foundation **5 july 2024 | 2 weeks**

Datascience and AIML intern

Machine Learning: Supervised Machine Learning, Linear Regression, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, Support Vector Machines (SVM), Naive Bayes
Model Evaluation: Confusion Matrix, Precision, Recall, F-score, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE)
Concepts: Overfitting, Underfitting, Generalized Model, Loss Function, Bias, Variance, Entropy, Information Gain, Gini Coefficient, Euclidean.
Regression Techniques: Ridge Regression, Lasso Regression
Probability and Classification: Conditional Probability, Understanding classification methods and applications

Linuxbean Solution LLP as Steve AI labs **8 April 2024 - present**

Python developer

Developed a Flask-based simple Chatbot

Created a conversational AI chatbot using Flask and OpenAI's GPT-3.5-turbo model for intelligent, context-aware responses.

Integrated OpenAI API: Utilized the OpenAI API to process and generate natural language responses and chatcompletion help to understatnd based on user input.

Ensured Security: Managed secure handling of API keys using environment variables to maintain application security.

Developed a Video-to-Image Extraction Tool

Created a FastAPI application for extracting the best images from videos. Used OpenCV to capture video and extract frames per second. Filtered out blurred images from the frames using cvtColor to convert colored images to grayscale. Employed Haar Cascade classifiers to detect faces in frames. Used DeepFace analyzer to detect emotions such as happy, surprised, and neutral. Implemented efficient file handling, including temporary file storage, to save the best images.utilized Docker for containerization and environment management.

PDF Document Processing and Text rettrival and question answering

Developed a Python application using PyPDF2 to extract and concatenate text from PDF documents.
Implemented LangChain for efficient text splitting and character-based segmentation.
Integrated OpenAIEmbeddings and FAISS for creating a robust document vector store, enabling fast and accurate similarity searches.
Designed a dynamic question-answering pipeline using LangChain's document and retrieval chains.
Developed a Python application to retrieve and process text data from a Wikipedia page using LangChain and WebBaseLoader.
Utilized OpenAIEmbeddings for semantic embeddings and FAISS for efficient storage and retrieval of text embeddings.
Integrated a ChatPromptTemplate to prompt the system for context-based question answering.

Background change by prompts

Integrated Stable Diffusion inpainting pipeline pretrained on the "stabilityai/stable-diffusion-2-inpainting" model to generate high-quality, realistic backgrounds that seamlessly blend with input images.
Designed an interactive interface using Gradio for intuitive user interaction, allowing users to input prompts and upload images to dynamically generate and view modified images with changed backgrounds.

Machine Test Site

Developed a platform for testing user-provided Python code with features for converting source code into executable form and providing a Python execution environment. Created APIs for running test cases and evaluating code correctness using Pytest, supporting error handling and feedback for logical mistakes in the user’s code. The project is run by Poetry, with all dependencies listed in pyproject.toml and commands added to run the FastAPI project.
Added an authentication API to create users and save them into a PostgreSQL database using SQLAlchemy for adding, committing, and saving. Created tests for user creation and added various question types for testing, including MCQ, text answers, and coding questions. Implemented CRUD operations for managing the question library.