

Project Portfolio - Muhammad Humayun Abdullah

hamayuna47@gmail.com

Botnist – AI-Powered Web Intelligence Chatbot

Tech Stack: MERN Stack, Python, Node.js, LangChain, LLaMA 2, ChromaDB, MPNet Embeddings

Summary: Developed a multi-tenant AI system that transforms websites into intelligent chatbot APIs. The system scrapes data, classifies it using NLP, stores it in vector databases, and uses Retrieval-Augmented Generation to answer user queries.

Key Features:

- Custom scraper for both static and dynamic websites
- NLP pipeline for data cleaning and classification
- Multi-tenant API with isolated vector stores per client
- LLM-powered chatbots using LangChain and LLaMA 2
- Admin panel to manage users, feedback, and analytics

PDF Chatbot using LLaMA2 and RAG

Tech Stack: Python, LangChain, LLaMA2, HuggingFace Transformers, ChromaDB, Streamlit

Summary: Developed an interactive chatbot capable of answering questions from uploaded PDFs using Retrieval-Augmented Generation (RAG) architecture with LLaMA2. Implemented chunking, vector store indexing, and conversational context.

Key Features:

- Efficient PDF parsing and embedding
- Seamless conversational memory integration
- Fast retrieval using Chroma vector DB

Web Data Extraction Bot

Tech Stack: Python, BeautifulSoup, Requests, Selenium

Summary: Created customizable web scraping scripts to collect structured data from multiple dynamic and static websites.

Key Features:

- Works with JavaScript-rendered pages
- Supports CSV/JSON export
- Anti-bot evasion techniques included

Custom Neural Network from Scratch

Tech Stack: Python, NumPy

Summary: Built a complete feedforward neural network with backpropagation for binary/multiclass classification problems.

Key Features:

- Custom weight initialization and activation functions
- Manual gradient computation
- Training loop with loss minimization

Timetable Generator using Genetic Algorithms

Tech Stack: Python

Summary: Solved a university scheduling problem using evolutionary principles to optimize time-slot allocation.

Key Features:

- Conflict detection and minimization
- Fitness scoring and elitism
- Dynamic mutation/crossover rate

Glaucoma Detection from Medical Images

Tech Stack: Python, TensorFlow, CNN

Summary: Designed and trained a deep learning model for early-stage glaucoma detection from retinal images with over 95% accuracy.

Key Features:

- Data augmentation pipeline
- ROC curve and confusion matrix evaluation
- Integration with medical image DICOM standards

CycleGAN for Face to Sketch Translation

Tech Stack: PyTorch, GANs

Summary: Implemented a CycleGAN to translate real face photos into sketch representations without paired data.

Key Features:

- Unsupervised learning on CelebA dataset
- Generator-discriminator architecture
- High-quality visual results

Skin Disease Detection App

Tech Stack: Python, TensorFlow, Streamlit

Summary: Built a diagnostic CNN-based app to classify various skin diseases from user-uploaded images.

Key Features:

- Multi-class classification model
- Mobile-friendly UI
- Real-time prediction and feedback

Student Portal using Windows Forms

Tech Stack: C#, Windows Forms, MSSQL

Summary: Created a student management system supporting profile creation, attendance tracking, and result management.

Key Features:

- Authentication and role-based access
- CRUD functionality
- Modern UI and validation

Proof-of-Work Blockchain Simulation

Tech Stack: Python

Summary: Developed a conceptual blockchain with mining logic using proof-of-work algorithms and cryptographic hashing.

Key Features:

- Block generation and validation
- Simple P2P node simulation
- SHA256 hash-based consensus

Hotel Reservation System

Tech Stack: Java, JavaFX, MySQL

Summary: Desktop-based hotel reservation system with room management, bookings, and payments.

Key Features:

- GUI with calendar booking
- Admin dashboard
- Persistent database storage

Rush Hour Puzzle Game (Multithreaded)

Tech Stack: C++, SFML, Threads

Summary: Recreated the Rush Hour sliding block game with multithreaded support for simultaneous AI opponent.

Key Features:

- Pathfinding with A*
- Game logic separated in threads
- Visual game simulation

Brick Breaker Game (Assembly)

Tech Stack: NASM (x86 Assembly)

Summary: Designed and developed the classic Brick Breaker game using low-level assembly for educational purposes.

Key Features:

- Pixel-level graphics manipulation
- Keyboard input handling
- Collision detection

OlaDoc HMS Clone

Tech Stack: C++, OOP, File Handling

Summary: Created a clone of the OlaDoc health management system for appointment booking and patient records.

Key Features:

- Search by doctor, date, or specialty
- Persistent file storage
- Console-based interface with modular code