Muhammad Aliyan

PortFolio: https://aliyan16.github.io/
GitHub: https://github.com/aliyan16
Email: aliyanm12376@gmail.com
LinkedIn: linkedin.com/in/muhammad-aliyan-b0525b26b/
Phone: +923359354084

EDUCATION

National University of Science and Technology (NUST- CEME) Rawalpindi
Bachelor of Engineering in Computer Engineering

EXPERIENCE

Freelance **(Fiverr)** June 2023 – Present

MERN Full stack developer Contract, **3Dim** (China) Aug-2025

MERN Full Stack Developer Intern, **Zapply** (US) June 2025 - Aug 2025

AI/ML Intern, **CARE** Mar 2025-May 2025

Director **Web** & IT wing, Comppec Jan 2025 – May 2025

Deputy Director Admin & HR, Comppec Jan 2024-May 2024

PROJECTS

Full-Stack Freelancing Platform | React.js, TypeScript, Redux, Node.js, Express, MongoDB, Azure Blob Storage, JWT

- Designed and developed a Fiverr-inspired freelancing marketplace enabling users to register as buyers or sellers, create and manage gigs, browse available services, and place orders.
- Implemented secure authentication and authorization using JWT tokens and Redux state management for persistent sessions.
- Integrated Azure Blob Storage for efficient and scalable storage of gig images, profile pictures, and media assets.
- Built dynamic, responsive frontend components with React.js, TypeScript, Tailwind CSS, and Redux Toolkit, including gig listings, seller dashboards, order tracking, and profile management.
- Developed RESTful API endpoints in Express.js for CRUD operations on gigs, users, and orders, supporting file uploads with Multer and integration with Azure cloud services.
- Designed MongoDB schemas to optimize relationships between users, gigs, and orders, enabling fast querying and filtering.
- Added gig analytics (impressions, clicks, orders, and cancellations) and seller dashboard features for business performance tracking.
- Deployed the backend on Azure App Service and the frontend on Azure Static Web Apps, ensuring a fully cloud-hosted solution.

Full-Stack Social Media Application using MERN Stack | React.js, Node.js, Express, MongoDB, Multer, bcrypt

- Designed and developed a Facebook-inspired social media platform enabling users to create accounts, share posts (text, images, videos), view feeds, and interact with content through likes and comments.
- Implemented secure user authentication with bcrypt password hashing and JWT-based session management, ensuring data privacy and protection against unauthorized access.
- Developed a robust media handling system using Multer and MongoDB GridFS for efficient storage and retrieval of user-generated content (profile pictures, cover photos, posts, and stories).
- Created dynamic frontend components with React.js including News Feed, Reels, User Profiles, and Story features with responsive design for optimal cross-device experience.

- Implemented RESTful API endpoints with Express.js and Node.js for CRUD operations, supporting real-time updates to user content and interactions.
- Designed MongoDB schemas for optimal data relationships between users, posts, comments, and media, enabling efficient querying and population of nested data.
- Optimized performance through client-side state management, server-side pagination, and efficient media streaming techniques for large files.
- Added advanced features including profile customization, timeline organization, and comprehensive error handling for both client and server-side operations.

Full-Stack Real-Time Chat Application using MERN Stack | React.js, Node.js, Express, MongoDB, Socket.io, Multer

- Built with React.js, Node.js, Express, MongoDB, and Socket.io for instant messaging.
- Features: User auth (bcrypt), real-time messaging (text/media), chat history, online/offline status.
- Socket.io for live updates, message sync, and status tracking.
- React.js frontend with dynamic UI: Chat Interface, User List, Message Threads, and Profile Management.
- RESTful APIs (Express/Node.js) for user management, messaging, and chat operations.
- MongoDB for optimized data storage (users, messages, media) with efficient querying.
- Performance optimizations: State management, GridFS for media, and error handling.

AI-Powered Document Assistant "MERNBot" | Full-Stack Application using MERN Stack + OpenAI API

- Developed a MERN stack application integrating OpenAI GPT-3.5 for document summarization and AI-powered Q&A through a real-time chatbot interface.
- Enabled users to upload or paste documents and receive concise AI-generated summaries, with secure JWT-based authentication and role-based access control.
- Built RESTful API routes for text preprocessing, prompt engineering, and chatbot response handling, ensuring efficient and accurate interactions with OpenAI.
- Created a responsive React.js frontend with chat UI, loading states, and typing indicators for enhanced user experience and dynamic document interaction.
- Implemented MongoDB data models to store users, document history, and chat logs, supporting persistent and context-aware AI conversations.

Autonomous Vehicle System using Raspberry Pi and Classical Image Processing | Python, OpenCV, Raspberry Pi Camera, GPIO

- Designed and developed a self-driving car prototype using Raspberry Pi equipped with a camera module to simulate autonomous navigation in real-world scenarios.
- Implemented classical image processing algorithms (edge detection, contour analysis, color segmentation) using OpenCV to recognize obstacles, road boundaries, and directional cues in real-time.
- Developed decision-making logic enabling the vehicle to adjust its path dynamically based on detected objects and environmental observations.
- Utilized Raspberry Pi GPIO pins to control vehicle motion (e.g., motor speed and direction) based on visual input and obstacle detection.
- Created a lightweight and efficient pipeline to process camera input frames, extract meaningful features, and trigger navigation responses with minimal latency.
- Ensured robustness of the system by testing in various lighting and surface conditions and optimizing the image processing pipeline for performance and reliability.
- Achieved obstacle avoidance and basic lane-following capabilities without the use of deep learning, demonstrating the power of classical CV techniques on low-power hardware.

Mission Logging and Tracking System for Pakistan Army Pilots | Flutter, Dart, Django, Python, SQL, HTTP

- Developed a cross-platform mobile application using Flutter and Dart to streamline mission logging and tracking for Pakistan Army pilots and their units.
- Implemented secure user authentication for pilots and unit personnel, allowing pilots to log new missions and view their mission history.
- Designed a Django backend with RESTful APIs (using Django REST Framework) to handle mission data, user authentication, and real-time updates.

- Integrated SQL database for efficient storage and retrieval of mission details, including pilot information, mission locations, timestamps, and status.
- Enabled real-time mission tracking for units, allowing them to view recently added missions, apply filters (by pilot, location, or date), and assign follow-ups.
- Utilized HTTP requests for seamless communication between the Flutter frontend and Django backend, ensuring data consistency and reliability.
- Enhanced user experience with responsive UI/UX design, including mission summaries, search functionality, and notification alerts for new missions.
- Ensured data security with encrypted transmissions and role-based access control (RBAC) to restrict unauthorized access.

Anti-Sleep Detection System for Vehicle Drivers | Python, C++, Arduino, ESP32

- Designed and implemented an anti-sleep detection system for vehicle drivers using Arduino, ESP32, and various sensors.
- Developed a system to continuously monitor the driver's eyes using cameras and sensors, with an algorithm to detect drowsiness based on eye closure duration (3-5 seconds).
- Integrated an alarm mechanism to alert the driver if drowsiness was detected and programmed the system to automatically stop the vehicle if the driver failed to respond.
- Implemented signaling features to notify other drivers of potential hazards when the vehicle is stopped due to drowsiness.
- Created a mobile app for real-time monitoring of driver and vehicle status, and for remote control of vehicle functions.
- Utilized motors and other hardware components to control vehicle safety features and interface with the vehicle's existing control systems.

Audio classification using CNN | Python, TensorFlow, keras, Librosa, Numpy, Matplotlib

- Developed an audio classification model using deep learning to categorize sound signals into speech, street_music, and noise using the UrbanSound8K dataset.
- Extracted MFCC features (Mel-Frequency Cepstral Coefficients) from audio signals for feature representation and classification.
- Built and trained a deep neural network (DNN) using TensorFlow/Keras, achieving high classification accuracy.
- Visualized spectrograms and waveforms to analyze different sound categories using Librosa and Matplotlib.
- Processed and augmented audio data to improve model robustness and generalization.
- Evaluated model performance using accuracy metrics, confusion matrix analysis, and loss plots.
- Implemented data preprocessing techniques such as feature scaling, label encoding, and train-test splitting for optimized model training.
- Designed and trained the model using Adam optimizer and categorical cross-entropy loss function to handle multi-class classification effectively.

Real-Time Hand and Face Detection Using OpenCV and MediaPipe | Python, OpenCV, MediaPipe, NumPy

- Developed a real-time hand and face detection system using OpenCV and MediaPipe to detect and track hands and faces in live video streams.
- Implemented hand landmark detection to count the number of fingers raised, enabling gesture-based interaction.
- Utilized MediaPipe's Face Detection module to detect faces and draw bounding boxes around them in real-time.
- Processed and visualized hand landmarks and face bounding boxes using OpenCV for real-time video analysis.
- Enhanced the system by adding visual feedback, such as bounding boxes and finger count displays, for better user interaction.
- Leveraged NumPy for efficient array manipulations and calculations during image processing.

Skills

Technical Skills

Web Development:

- Full-stack development with the MERN Stack (MongoDB, Express.js, React.js, Node.js).
- Built RESTful APIs using Express.js & Node.js with MongoDB (including GridFS for media storage).
- Developed dynamic and responsive UIs with React.js, including state management and component-based architecture.
- Proficient in HTML, CSS, JavaScript (ES6+) for frontend development.
- Implemented user authentication (bcrypt hashing, JWT) and session management.
- Used Multer for file uploads (images, videos) and MongoDB GridFS for efficient media storage/streaming.

Machine Learning & Deep Learning:

- Proficient in building and deploying machine learning models using Scikit-Learn, TensorFlow, and Keras.
- Experienced in data preprocessing, feature engineering, and model evaluation techniques.
- Skilled in implementing supervised and unsupervised learning algorithms for classification, regression, and clustering tasks.
- Hands-on experience with neural networks, including CNNs for image and audio data, and transfer learning.

Data Analysis & Visualization:

- Expertise in data manipulation and analysis using NumPy and Pandas.
- Proficient in creating insightful visualizations using Matplotlib, Seaborn, and Plotly.

Mobile & Full-Stack Development:

- Developed cross-platform mobile apps using Flutter & Dart (e.g., mission tracking system for Pakistan Army).
- Built RESTful APIs with Django (Python) and integrated them with Flutter frontends.
- Skilled in ReactJS for dynamic web interfaces and familiar with HTML, CSS, JavaScript.
- Currently learning Flask for lightweight backend development and model deployment.

Databases & Backend:

- Worked with SQL databases (MySQL, PostgreSQL) for structured data storage and retrieval.
- Used HTTP/API integration (Django REST Framework, Fetch, Axios) for seamless frontend-backend communication.

Version Control & Collaboration:

- Experienced in version control using Git and GitHub for collaborative project development.
- Familiar with DVC (Data Version Control) for managing machine learning datasets and pipelines.

MLOps & Model Deployment:

- Knowledge of MLflow for experiment tracking, model packaging, and deployment.
- Currently learning Docker for containerization and deployment of machine learning models.
- Exploring additional MLOps tools to streamline model lifecycle management, including CI/CD pipelines and monitoring.

Programming & Tools:

- Strong programming skills in Python for machine learning, data analysis, and automation.
- Familiar with OS module for file handling and system operations.

Continuous Learning:

- Actively expanding knowledge in MLOps tools and practices to enhance model deployment and scalability.
- Enrolled in advanced courses and certifications to stay updated with the latest trends in AI, machine learning.