IBA SMCS

School of Mathematics and Computer Science

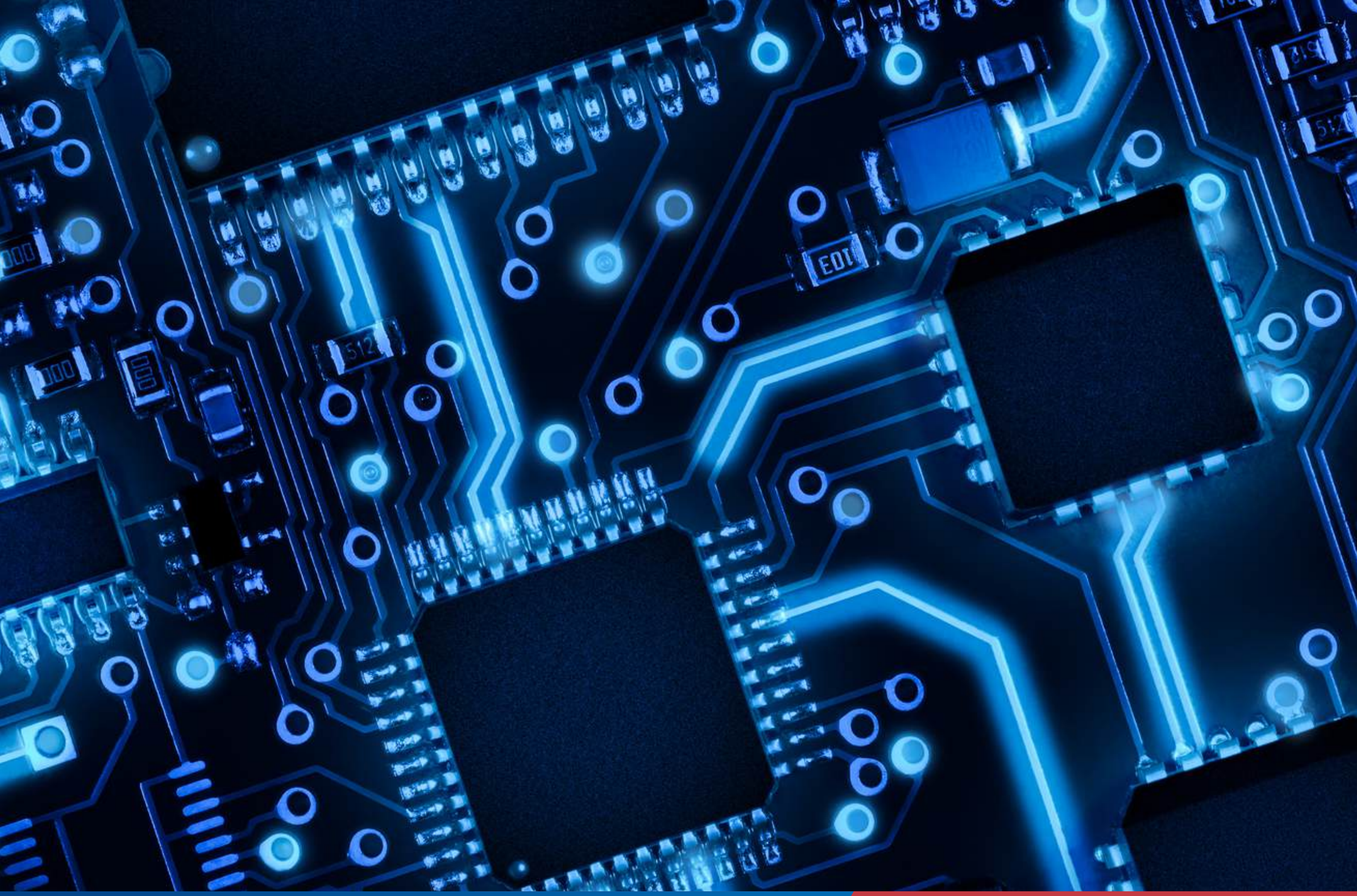
70 YEARS & BEYOND

Leadership ▪ Excellence ▪ Impact

BS(CS) FINAL YEAR PROJECTS

CATALOGUE (CLASS OF 2025)

DEPARTMENT OF COMPUTER SCIENCE,
SCHOOL OF MATHEMATICS & COMPUTER SCIENCE,
IBA-SMCS, KARACHI



Preface

The Final Year Project (FYP) Catalogue for the academic year 2024–2025 stands as a testament to the dedication, innovation, and scholarly rigor of our graduating students at the School of Mathematics and Computer Science, IBA Karachi. This compilation encapsulates the culmination of their undergraduate journey, reflecting a harmonious blend of theoretical knowledge and practical application.

Throughout their academic tenure, our students have been immersed in a curriculum designed to foster critical thinking, problem-solving, and research-oriented learning. We extend our heartfelt appreciation to the faculty members who have mentored and guided these students, providing invaluable insights and fostering an environment conducive to research and innovation.

As you delve into the pages of this catalogue, we invite you to explore the intellectual endeavors of our students, whose work not only contributes to academic discourse but also holds the potential to impact industry and society at large. We take immense pride in their achievements and look forward to their continued contributions to the fields of Mathematics and Computer Science.

Dr. Shakeel Khoja
Dean, School of Mathematics
& Computer Science (SMCS)

Table of Content

Preface	01
Advisory	03
AI Agents - Flomny	04
APIfy – A No-Code API Management Platform	05
Blind Navigation System	06
DataVeritas	07
Default Prediction Model	08
Edu-Rev	09
Envi	10
ETL/ELT Pipeline / Dataflow	11
Federated Learning - NeuroSymbolic AI	12
ForeStock – Stock Price Forecaster	13
Game-Based Recruitment Assessment	14
Harf Ba Harf	15
HDV – AI-Based Video Enhancement System	16
Indus Sahulat – Real-Time Ambulance Dispatch System	17
IntelliClaim – AI-Powered Vehicle Damage Assessment for Insurance Claims	18
Intelliscore – AI-Based Credit Scoring System	19
Jolyne – AI-Powered Psychiatric Assistant	20
LLM Auto Pentester – Autonomous Penetration Testing using Large Language Models	21
MindTrack – Adaptive Intelligence Assessment Platform	22
NeuralTrace – AI-Powered Memory Assistant for Cognitive Disorders	23
OrthoVision – AI-Powered Dental Diagnostic and Management System	24
Rawangi – Smart Public Transport Companion for Karachi	25
RecruitWise – AI-Powered Conversational Interviewing & Resume Screening Engine	26
Satellite-Based Crop Yield Predictor – CNN-Driven Agricultural Forecasting for Pakistan	27
SocialBrain – AI-Driven Trend-Based Social Media Content Scheduler	28
VidSense – NLP-Based Intelligent Video Summarization and Accessibility Tool	29
Vroo – Secure & Sustainable Ride-Sharing Platform for Trusted Communities	30

AdVisory

Group Members

Alina Afghan (24491)
Hiba Mallick (24015)
Sakina Ali (24509)
Ayesha Nayyer (25141)

Supervisors

Supervisors:
Dr. Ali Raza (Assistant Professor)

Industry Mentor:
Shakir Ghani (Co-Founder, SocialChamp)

Project Overview

In today's competitive digital advertising landscape, small businesses face major obstacles due to limited resources, lack of marketing expertise, and the time-consuming nature of competitor analysis and campaign planning. Traditional advertising tools are complex, manual, and inaccessible to non-experts.

AdVisory is an AI-powered digital marketing assistant that simplifies and automates key marketing processes, including ad creation, budget optimization, and competitor research. It uses advanced AI models to generate high-quality ads and captions, streamline analysis of competitor ads, and optimize marketing campaigns—making digital marketing smarter, faster, and more accessible.

Technologies Used

- AngularJS
- Node.js & Express
- Flask
- CVX Solver (Convex Optimization)
- GPT (OpenAI)
- FLUX
- Stable Diffusion
- CLIP & BLIP (Vision Models)
- Real-ESRGAN (Image Enhancement)
- Vercel (Serverless Backend Deployment)

Specifications

- **System Requirements:** Minimum 4GB RAM, modern JavaScript-enabled browser (Chrome, Firefox, or Edge), stable internet connection.
- **Backend:** Hosted serverlessly on Vercel. All APIs and model keys are securely stored in environment variables to ensure security.
- **Security & Compatibility:** The system supports cross-browser functionality and maintains a secure backend with minimal infrastructure requirements.

Methodology

The system is composed of multiple modules:

Competitor Analysis & Ad Generation Module
Uses data scraped from Meta Ad Library and generative AI models (GPT and FLUX) to create ads and captions inspired by competitors' active campaigns.

Product Ad Module
Employs Python libraries and Real-ESRGAN to process product images. Users can customize ad backgrounds using prompts (via Stable Diffusion), adjust image dimensions, and design social media-ready ads.

Budget Allocation Module
Applies convex optimization techniques using logarithmic return modeling to recommend ideal budget distribution based on KPIs like conversion rate.

Audience Prediction Module
Uses CLIP and BLIP to analyze visual and conceptual trends in uploaded images, predicting audience demographics and optimizing targeting strategies.

All modules are integrated through APIs and accessible through an AngularJS frontend. Backend services are built with Flask and Node.js, ensuring responsive performance and scalability.

Keywords

Generative AI, Vector Optimization, Digital Advertisement, Web Application

AI Agents - Flomny

Group Members

Muhammad Ammar (24456)
Muhammad Shehzad (24453)
Maleeha (24622)
Rayyan Masood (25168)
Muhammad Shayan Shamsi (24519)

Supervisors

Dr. Tahir Syed (Assistant Professor)

Industry Mentor:
None

Project Overview

Flomny is a multi-agent platform designed to empower advanced users by enabling the creation of custom workflows using supported integrations. The platform's standout capability is its facilitation of custom integrations: users can provide documentation for their server or API and seamlessly set up cross-platform workflows.

This system serves as a comprehensive solution for managing, triggering, and monitoring automated tasks across multiple services, ultimately streamlining business processes and software orchestration.

Technologies Used

- **Programming Languages:** Golang, Python, TypeScript / JavaScript
- **Frontend:** Next.js
- **Backend:** Gin-Gonic, gRPC, Redis Events, FastAPI (Websockets), LangChain / LangGraph
- **Datastores:** MongoDB, Pinecone, Amazon S3
- **Infrastructure & Deployment:** Docker Containers, AWS ECS (Elastic Container Service), Vercel (for frontend)

Methodology

The project was developed using the Agile methodology, with tasks divided sprint-wise among team members. Each sprint focused on building specific features, followed by integration, testing, and versioning through GitHub.
Two main internet-facing backends are implemented:
The **API Gateway** handles all CRUD operations, performs authorization, and routes requests to appropriate gRPC services.
A **FastAPI - based backend** utilizes WebSocket communication for real-time task updates and integration with the Agentic Platform.
Additionally, a producer-consumer model is implemented for capturing Redis events to store data into the Pinecone vector database. The system is designed with scalable cloud-based services and secure architecture for handling complex workflow tasks.

Specifications

- **System Design:**
  - Architecture defined using an SRS document
  - Microservices with gRPC-based communication Redis for event handling
  - WebSocket-based chat simulation for agent task updates
  - Hosted on AWS ECS for scalability
  - Cloud-native databases: MongoDB Atlas, Pinecone, S3
  - Frontend hosted on Vercel
- **Security & Scalability:**
  - APIs routed securely through gateways
  - All keys stored in environment variables
  - Scalable containerized backend for performance optimization

Keywords

Multi Agent Platform, LLMs, Web Application, Cloud Deployment, System Orchestration

APIfy – A No-Code API Management Platform

Group Members

Ukasha Khalid (24462)
Syed Hasan Ali (24068)
Burhanuddin Makra (24460)
Abdul Rehman Ansari (24487)
Saad Ahmed Khan (24440)

Supervisors

Dr. S.M. Faisal Iradat (Assistant Professor)

Industry Mentor:
Talal Nasir (Software Engineer-II, Astera)

Project Overview

APIfy is a no-code API management platform that empowers users to create, deploy, and monitor APIs without writing a single line of code. Designed as a web-based SaaS solution, APIfy enables database manipulation, auto-generates API documentation, and offers in-platform testing and monitoring tools.
With one-click deployment on secure infrastructure, this tool simplifies backend development for non-technical users, democratizing API creation and promoting fast, secure, and scalable application development.

Technologies Used

- **Programming Languages:**
  - JavaScript (Node.js for backend, React/Next.js for frontend)
- **Frameworks & Libraries:**
  - Node.js (RESTful API development)
  - Next.js (React-based frontend framework with SSR/CSR)
  - Mongoose (ODM for MongoDB)
  - React (UI Components)
  - JWT (for secure authentication)
- **Databases:**
  - MongoDB (NoSQL)
  - MongoDB Atlas (Cloud-hosted)
- **Tools & Infrastructure:**
  - Vercel (Frontend deployment)
  - Swagger (API documentation)
  - Docker (Containerization)
  - WebSocket (Real-time monitoring) CI/CD Pipelines
  - HTTPS/HTTP2 (Secure communication)

Keywords

No-code Development, API Management, API Monitoring & Testing, User Interface Design, Cloud Integration

Methodology

APIfy was built using a modular agile approach, focusing on rapid, iterative development. The system was divided into key functional modules:

- Authentication
- Database Management
- API Builder (drag-and-drop, no-code)
- Testing, Deployment, and Monitoring

The frontend and backend were developed in parallel, with continuous integration/deployment (CI/CD) ensuring reliability. The platform prioritizes user-centered design, security, and scalability, including real-time monitoring and robust authentication using JWT.

Specifications

- Web-based SaaS application, accessible via any modern browser
- Hosted on Vercel, with user APIs deployed via Node.js backend
- MongoDB Atlas used for scalable, cloud-based database storage
- Real-time monitoring with WebSocket
- Secure communication with HTTPS/HTTP2
- Handles up to 1000 concurrent API requests with
- <200ms response time
- 99.9% uptime ensured with Docker and CI/CD automation

Blind Navigation System

Group Members

Faris Ejaz (24470)
Ali Khurram (25250)
Umer Ahmed (25230)
Zeeshan Ismail Khan (24477)

Supervisors

Academic Supervisor:
Imran Khan (Professor)

Industry Mentor:
Shalin Amir Ali (Engineer – NLP, Folio3)

Project Overview

The Blind Navigation System is a mobile application designed to assist visually impaired individuals in navigating real-world environments safely and independently. The app integrates real-time obstacle detection with voice navigation cues, providing spatial awareness and guided directions from point A to point B.

Key features include lightweight, custom heuristics for depth estimation (to maintain performance on Android devices), and integration with map SDKs and voice translation tools. The solution is tailored to be cost-effective and practical for widespread use in real-world settings.

Technologies Used

- **Platforms & Tools:**
  - Android Studio
  - OsmAnd & OpenStreetMap (OSM) SDK
- **Programming Language:**
  - Kotlin
- **Libraries & Models:**
  - YOLOv8n (for real-time object and obstacle detection)
  - Google Translate (for voice navigation cues)

Methodology

The development approach follows a use-case-driven methodology, where each user flow is identified, built, and tested in iterative cycles. Group members divided responsibilities based on modules and use cases.

This allowed for:

- Targeted optimization of each feature
- Manual testing in real-world scenarios
- Refinement through iterative feedback and improvements

Special emphasis was placed on lightweight computation, avoiding processing-heavy models and implementing custom heuristics for depth estimation to ensure the app works effectively on standard Android devices.

Specifications

- **Supported Devices:** Android phones running Android 10 or higher
- **Installed SDK:** OsmAnd
- **Detection Module:** YOLOv8n optimized for mobile
- **Navigation:** Real-time spatial and voice guidance
- **Performance:** Prioritized for low-latency, real-time obstacle detection
- **Voice Assistance:** Google Translate TTS integration for accessible feedback

Keywords

Custom Heuristics, Computer Vision, Mobile Application, SDK Integration, Assistive Technology

DataVeritas

Group Members

Hasan Ali Vejlani (24466)
Muhammad Usman (24454)
Haider Saeed Khan (24123)

Supervisors

Academic Supervisor:
Maria Rahim (Lecturer)

Industry Mentor:
Ahsan Siddiq (Senior Software Engineer, Astera Software)

Project Overview

In the era of data-driven decision-making, organizations face increasing challenges in maintaining data quality, discoverability, and access control. DataVeritas is a web-based platform developed to address these issues through a robust and user-friendly data governance solution.
The system enables users to:

- Upload and profile CSV/TXT datasets
- Define and execute nested custom data quality rules via a powerful rule parser
- Evaluate dataset conformity based on these rules
- Manage secure and granular role-based access control (RBAC)
- Enhance dataset discoverability using intuitive tagging and search functionality

Designed with scalability and performance in mind, DataVeritas empowers organizations to implement governance practices without relying on complex or costly enterprise solutions.

Technologies Used

- **Frontend:**
  Next.js
  MUI CSS
- **Backend:**
  ASP.NET Core (C#)
  Entity Framework Core (EF Core)
  RESTful APIs
- **Database & Storage:**
  Microsoft SQL Server
  Local file storage for large datasets (up to 500MB)
- **Libraries & Tools:**
  CsvHelper (CSV/TXT parsing)
  Regex (rule expression parsing)

Methodology

The project followed an Agile development approach, with the team working in defined sprints, each focusing on specific feature implementations:

- File upload and profiling
- Custom rule parser
- RBAC and permissions
- Search and tagging

The development was guided by a modular, layered architecture, ensuring separation between the frontend, backend, and data layers. Continuous feedback loops with supervisors enabled iterative refinement and improvement.
The initial phase included a market analysis to evaluate gaps in existing data governance platforms, helping tailor the solution to real-world user needs with improved usability, flexibility, and performance.

Specifications

- **System Architecture:** Monolithic Web Application
- **Frontend:** Next.js with Material UI (MUI CSS)
- **Backend:** ASP.NET Core with EF Core ORM
- **Database:** Microsoft SQL Server
- **File Handling:**
  - Supports CSV/TXT files up to 500MB
  - Local storage with path indexing in database

  Security:
  - Role-Based Access Control (RBAC)
  - Hashed credentials
  - Permission expiry monitoring
- **API Support:** RESTful endpoints with async operations for rule execution and data profiling
- **Minimum Requirements:**
  - RAM: 16+ GB
  - CPU: Multi-core
  - OS: Windows 10/11
  - Disk Space: 100+ GB
  - Dependencies: .NET 8 Runtime, SQL Server v20.2

Keywords

Data Repository, Web Application, Data Quality Rules, Data Profiling, Role-Based Access Control

Default Prediction Model

Group Members

Zohaib Ali Mughal (24648)
Muhammad Saim Khalid (25256)
Muhammad Shaheer Patel (24480)
Muhammad Hashir Saeed (24492)
Sarfaraz Ahmed

Supervisors

Academic Supervisor:
Dr. Tariq Mahmood (Professor)

Industry Mentor:
Mr. Saad Suri (AVP/Manager Digital Financing, Meezan Bank)

Project Overview

The Loan Default Prediction Model is a machine learning-based solution designed to assist Meezan Bank in proactively managing credit risk. By analyzing financial and operational data of corporate borrowers, the model predicts the likelihood of loan default with high accuracy. This predictive tool empowers the bank to make data-driven lending decisions, reducing loan loss risks and enhancing the security of its credit portfolio. The model's integration into the bank's credit evaluation process supports faster, more reliable, and scalable risk assessments.

Purpose

To equip Meezan Bank with a predictive system that flags high-risk corporate borrowers and strengthens overall credit risk management.

Objective

To develop and validate a machine learning model that forecasts default probabilities using structured company and financial data.

Contribution

The project delivers a customized, scalable, and validated credit risk solution using advanced analytics. It enhances risk profiling and allows for more secure lending practices through intelligent automation and accurate prediction.

Technologies Used

- **Programming Language:** Python 3.8+
- **Machine Learning Libraries:** scikit-learn, pandas, NumPy
- **Development Environment:** Google Colab Pro (16GB RAM)
- **Backend API:** Flask
- **Containerization & Deployment:** Docker (for database and service portability)

Methodology

The project adopts an N-tier architecture with Flask handling API development and Docker managing the backend database container. The model was trained and tested on Google Colab Pro using large corporate datasets. Seven different ML algorithms were evaluated using K-fold cross-validation to prevent overfitting. Hyperparameter tuning was done through grid search to optimize model performance. Feature engineering and preprocessing pipelines ensured clean, relevant input data. The final model was deployed in a containerized environment, ensuring consistent performance across different systems.

Specifications

- Development on Google Colab Pro with 16GB high-memory runtime
- Docker for database containerization and environment consistency
- API layer built with Flask
- Compatible with Python 3.8+ and industry-standard ML libraries
- Model supports scalable deployment for real-time or batch predictions

Keywords

Default Prediction, Machine Learning, Predictive Analysis, Credit Risk, Web Application

Edu-Rev

Group Members

Muhammad Mustafa Faysal Khokhar (25231)
Muhammad Mashhood Uddin (24528)
Wasi Tahir Khan (25137)
Tariq Iqbal (19091)
Muhammad Ibad (25158)

Supervisors

Academic Supervisor:
Dr. Muhammad Saeed (Professor)

Industry Mentor:
Umair Ahmed (Senior Software Development Engineer, Securiti.ai)

Project Overview

Edu-Rev is a cutting-edge educational technology platform designed to deliver personalized, curriculum-aligned, and easily accessible learning experiences for O/A-Level and IGCSE students. Addressing common tutoring challenges like limited flexibility and geographic restrictions, Edu-Rev uses AI-driven tools to create an engaging and autonomous learning environment. The platform includes video lectures, concise notes, AI-evaluated practice questions, and a 24/7 chatbot to enhance interactivity and learning outcomes.

Purpose

To develop a digital educational platform that provides students with AI-based personalized learning, scalable access to resources, and intelligent progress feedback.

Objective

To create a comprehensive, self-paced learning solution integrating AI for answer evaluation, content retrieval, and learner support, aimed at improving educational outcomes for O/A-Level and IGCSE students.

Contribution

Edu-Rev offers an AI-integrated edtech solution that democratizes quality education by enabling location-independent, tailored, and scalable learning. It enhances learner engagement and academic performance using automation and advanced analytics.

Technologies Used

- **Frontend Development:** React.js, Tailwind CSS
- **Backend Development:** Node.js, Flask
- **Database:** MySQL, Microsoft SQL Server
- **AI/NLP & ML Models:** DistilRoBERTa (fine-tuned), Retrieval-Augmented Generation (RAG), BM25, FAISS, GPT4AllEmbeddings
- **Document Processing:** Tesseract OCR, LangChain
- **Hosting & Infrastructure:** AWS / Azure Cloud, Hugging Face Spaces
- **Authentication:** JWT, Two-Factor Authentication Version Control: Git, GitHub
- **Testing & Deployment:** Postman, JMeter, CI/CD (GitHub Actions)
- **Security & Compliance:** AES-256 Encryption, GDPR Compliance, RBAC
- **Design & Prototyping:** Figma
- **Video Hosting:** YouTube, Vimeo

Methodology

Edu-Rev followed an iterative Agile-inspired approach consisting of four stages:

1. **Design & Development:** Built user-friendly interfaces with React.js and Tailwind CSS, developed backend services using Node.js and Express.js, and set up Microsoft SQL Server for structured data storage.
2. **AI Integration:** Integrated DistilRoBERTa for answer grading and a hybrid RAG pipeline for content retrieval. FAISS and BM25 enabled accurate question searches, and AI feedback loops were added for performance tracking.
3. **Feature Improvement:** Enhanced system scalability, added a 24/7 AI chatbot trained on curriculum material, and implemented dynamic feedback features to guide student improvement.
4. **Testing & Deployment:** Conducted thorough unit, integration, and user testing. Enforced strict security (AES-256, GDPR compliance), and deployed via cloud with GitHub-based CI/CD workflows.

Specifications

- **Client-Side Requirements:**
  - OS: Windows 10+, macOS 10.14+, Ubuntu 20.04+
  - Browser: Chrome, Edge, Safari (latest)
  - RAM: 4 GB+
  - Internet: 2 Mbps+
- **Server-Side Requirements:**
  - OS: Ubuntu 20.04 or Windows Server 2019+
  - Processor: Intel i5 or higher (Quad-core)
  - RAM: 16 GB+
  - Storage: SSD, 100 GB+
  - Model Hosting: Hugging Face Spaces or AWS SageMaker
- **APIs and Communication:**
  - RESTful APIs, JWT-based authentication, WebSockets for real-time features
  - External APIs: Tesseract OCR, YouTube/Vimeo, Cloudflare
- **AI & NLP Components:**
  - DistilRoBERTa (fine-tuned), GPT4AllEmbeddings, RAG (FAISS + BM25), LangChain

Keywords

Personalized Learning, Natural Language Processing, Web Application, Chabot

Envi

Group Members

Muhammad Sarim Ul Haque (24493)
Anoosha Gurbani (25126)
Muhammad Siddique Khatri (25135)
Fatima Tanveer (25183)

Supervisors

Academic Supervisor:
Dr. Tariq Mahmood (Professor)

Industry Mentor:
Usman Hanif (BI Specialist, Contour Software)

Project Overview

ENVI is a data visualization tool aimed at democratizing access to Business Intelligence and Business Analysis for users with minimal to no technical skills. This tool is designed to empower businesses lacking the capacity to hire or train BI personnel, by enabling them to make informed long-term and short-term decisions based on their data. It automatically performs statistical tests and visualizes insights through intuitive dashboards.

Purpose

To develop a BI platform that enables non-technical users to easily extract insights from data and create dashboards without needing specialized analytical knowledge.

Objective

To automate the statistical analysis and dashboard creation process using a powerful backend engine and provide an accessible interface for seamless data interaction.

Contribution

ENVI lowers the barrier to entry for data-driven decision making, offering a plug-and-play platform that delivers actionable insights without the need for dedicated BI teams.

Technologies Used

- **Frontend Development:** Next.js
- **Backend Development:** Python, Django, Django REST API
- **Libraries/Packages:** Pandas, NumPy
- **Database:** PostgreSQL

Methodology

The project follows a modular development approach. The core engine automates statistical testing and visualization. Data is ingested and processed via the backend using Python and Django. The REST API facilitates communication between the backend and the Next.js frontend, allowing users to easily upload datasets, generate insights, and view visualizations.

Specifications

- Plug-and-play system with no unique hardware or software dependencies
- Compatible with general-purpose operating systems and modern browsers

Keywords

Data Visualization, Statistical Modelling, Business Intelligence, Database Integration, Web Application

ETL/ELT Pipeline / Dataflow

Group Members

Hazim Ghulam Farooq (25148)
Ammar Lokhandwala (25245)
Sameed Khatri (25249)
Kunal Kumar (25145)
Huzaifa Bin Tariq (25133)

Supervisors

Academic Supervisor:
Ms. Abeera Tariq (Lecturer)

Industry Mentor:
Humza Khatri (Associate Software Architect, Astera)

Project Overview

This project introduces a hybrid ETL/ELT pipeline that dynamically chooses between ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform) workflows based on the dataset's properties and system performance considerations. It incorporates AI-based schema design to support users in structuring their data more effectively and provides an easy-to-use configuration interface to manage pipeline behaviors.

Purpose

To build a next-generation data pipeline that adapts intelligently to optimize performance across diverse workloads while simplifying schema design for users.

Objective

To enhance efficiency in data processing through intelligent pipeline switching and AI-driven schema suggestions, and to offer enterprise - grade performance and scalability.

Contribution

This project significantly improves how organizations manage and transform data by combining flexibility, intelligent decision-making, and user-centric design in a scalable cloud-ready pipeline architecture.

Technologies Used

- **Frontend:** React, JavaScript
- **Backend:** .NET, C#, Python, FastAPI, Langchain
- **AI/LLM Tools:** Groq
- **Data Processing:** Apache Spark
- **Database Systems:** MySQL, PostgreSQL, SQL Server DevOps/Cloud: Docker, AWS, Azure

Methodology

The team followed Agile methodology with regular sprints to iteratively develop core components like the data characterization module, pipeline decision engine, and AI schema designer. Continuous integration and testing ensured flexibility, robustness, and system adaptability throughout the development cycle.

Specifications

- Cloud-deployable with minimum requirement: 8-core CPU, 32 GB RAM, SSD storage
- Recommended for Azure-hosted instances like m5.2xlarge
- Docker-based container deployment (2–4 vCPUs, 4–8 GB RAM per container)
- Apache Spark cluster setup for heavy transformations (Master: 8-core/32GB, Workers: 4–8 core/16GB)
- Integrated Groq APIs for schema design
- RESTful APIs via FastAPI and SQLAlchemy for backend logic and database communication

Keywords

Data Engineering, Data Pipelines, Containerization, AI Agents, Cloud Integration

Federated Learning - NeuroSymbolic AI

Group Members

Mirza Mustafa Ali Jah (24527)
Saira Batool (25182)
Babar Ali (25154)
Sikandar Ali (24947)
Irfan Karim (25254)

Supervisors

Academic Supervisor:
Dr. Faisal Iradat (Assistant Professor)

Industry Mentor:
Dr. Waseem Iqbal (Assistant Professor, Sultan-al-Qaboos University)

Project Overview

The rapid growth of IoT networks presents major security risks, necessitating scalable and privacy-preserving solutions. This project proposes NeSy-FL-IDS, a NeuroSymbolic Federated Learning-based Intrusion Detection System, that combines symbolic reasoning with neural networks to deliver interpretable, accurate, and robust threat detection without compromising user privacy. Federated learning ensures that only model updates—not raw data—are shared with a central server.

Purpose

To develop an intrusion detection system that is privacy-aware, explainable, and effective for real-time IoT environments.

Objective

To enhance IoT security by integrating NeuroSymbolic AI into a federated learning framework, offering both high detection accuracy and model transparency.

Contribution

This work pioneers the integration of Logic Tensor Networks into federated learning, offering an intrusion detection solution that surpasses traditional black-box approaches in both interpretability and performance.

Technologies Used

- **Programming Languages/Frameworks:** Python, PyTorch, TensorFlow
- **Federated Learning Framework:** Flower (flwr)
- **NeuroSymbolic Tools:** Logic Tensor Networks
- **Dataset:** KDDCup99
- **Security Tools:** Lightweight cryptography protocols for secure update transmission

Methodology

The NeSy-FL-IDS framework allows each IoT device (client) to train a local model using its own data. Only model updates are sent to a central server, which refines a global model iteratively. NeuroSymbolic reasoning adapts rule weights based on logical satisfiability scores to improve interpretability. The system is evaluated using the KDD99 dataset, achieving 97–99% accuracy and low loss values (0.015–0.008), showcasing superiority over traditional centralized systems.

Specifications

- Hardware Requirements: Intel i5+ processor, 16GB RAM
- Lightweight cryptographic protocols for privacy
- Compatible with Python ML frameworks
- Tested on KDDCup99 dataset
- Secure federated updates with real-time threat detection capabilities

Keywords

NeuroSymbolic Federated Learning, Privacy Preserving IDS, Explainable AI, IoT Security

ForeStock – Stock Price Forecaster

Group Members

Mohammad Mohsin Panhwar (22837)
Sahil Kumar (25172)
Sneha Batra (25054)
Neelesh (25160)

Supervisors

Academic Supervisor:
Ms. Tasbiha Fatima (Lecturer)

Industry Mentor:
Mr. Hunzala Mushtaq (Software Engineer, Folio3)

Project Overview

Stock market prediction remains a complex challenge due to its dynamic and volatile nature, influenced by both historical trends and rapidly changing news. ForeStock introduces a modern approach to stock forecasting by combining machine learning models with real-time sentiment analysis of financial news and social media content. The system utilizes LSTM neural networks enriched with sentiment data to provide timely and accurate stock price predictions. It also offers a centralized dashboard for traders and analysts, featuring real-time updates, emergency alerts, and sector-specific insights.

Purpose

To develop a hybrid forecasting system that improves stock price prediction by integrating market sentiment and news analysis with traditional time-series data.

Objective

To enhance the accuracy and responsiveness of stock forecasts by leveraging LSTM models and real-time NLP-based sentiment analysis in a user-friendly, web-based platform.

Contribution

This project delivers a next-generation stock prediction tool that goes beyond numerical trends, interpreting human emotions and current events to adaptively forecast market shifts, providing users with a reliable decision-making assistant.

Technologies Used

- **Programming Languages:** Python
- **Machine Learning:** TensorFlow, Keras, Keras Tuner
- **NLP & Sentiment Analysis:** TextBlob, NLTK
- **APIs:** Yahoo Finance (via yfinance), NewsAPI
- **Backend Framework:** Flask
- **Frontend:** HTML, CSS, JavaScript, Chart.js
- **Visualization & Data Handling:** Matplotlib, Pandas, NumPy
- **Version Control:** Git

Methodology

The development followed an agile methodology with iterative sprints involving continuous feedback from academic and industry mentors. The project was divided into modules: data gathering, model development using LSTM, sentiment integration, and web interface design. Real-time news and sentiment streams were linked to the prediction engine, allowing the system to evolve in sync with market behavior. Each sprint culminated in testing and refinement to ensure usability, performance, and relevance.

Specifications

- AI-powered market sentiment integration
- Emergency alert system for significant price shifts
- LSTM-based stock trend forecasting
- Web-based dashboard with interactive charts
- Secure backend and privacy-compliant data handling
- Compatible with common systems (Intel i5+, 8GB RAM)

Keywords

Stock Price Prediction, Machine Learning, Sentiment Analysis, Time-Series Forecasting, Financial News Integration

Game-Based Recruitment Assessment

Group Members

Abdul Rafay (25151)
Zarlish Mohtasham (24452)
Maira Munir (24481)
Hafsa Nasir (24394)
Zara Mahroyan Kashif (25164)

Supervisors

Academic Supervisor:
Prof. Jawwad Farid

Project Overview

This project presents a gamified recruitment assessment platform that reimagines traditional personality tests such as the Myers-Briggs Type Indicator (MBTI) and the Big Five model. Recognizing that these tests, while insightful, are often perceived as lengthy and uninspiring, the platform transforms them into an engaging and efficient game-based experience. Candidates complete scenario-driven challenges that retain the psychological rigor of the original tests while increasing participation and reducing fatigue. By incorporating a Unity-based interface, the system provides real-time scoring and detailed personality insights to assist employers in more dynamic and data-driven hiring decisions.

Purpose

To modernize recruitment assessments through a gamified system that enhances candidate engagement while maintaining psychological test accuracy.

Objective

To design and implement a personality-assessment game that shortens testing time, ensures accurate profiling, and improves the recruitment experience for both employers and candidates.

Contribution

The project innovatively integrates validated psychological models into a modular game-based platform, enabling organizations to assess candidates in a more interactive and scientifically grounded manner. It bridges the gap between entertainment and evaluation, offering a scalable tool for personality-based recruitment.

Technologies Used

- **Frontend:** React.js, Tailwind CSS, shadcn/ui, React Router
- **Backend:** Node.js, Express.js, REST APIs
- **Game Engine:** Unity (WebGL module integration)
- **Database:** MongoDB Atlas (cloud-hosted)
- **Authentication:** JWT
- **Tools & Platforms:** Git/GitHub, Visual Studio, Axios, Lucide Icons

Methodology

The team employed the agile SCRUM methodology using Jira to manage iterative sprints, continuous feedback, and milestone-based feature releases. Development began with thorough requirement gathering and system architecture planning. The backend was developed as a modular API service, with the frontend built in React and integrated with a Unity-powered game engine. Each sprint cycle included functional and usability testing, ensuring a robust and user-friendly interface throughout development. Scenario-based question logic was embedded into the gameplay to derive personality scores with high accuracy.

Specifications

- Minimum 4GB RAM system requirement
- Compatible with modern browsers (Chrome, Edge)
- Unity WebGL game integrated with React frontend
- JWT-secured authentication
- RESTful APIs for candidate data and test result handling
- Cloud-based MongoDB for scalable data storage
- Dashboard for test progress and result visualization

Keywords

Web Application, User Game, Interface Design, Gamification, Personality Testing

Harf Ba Harf

Group Members

Muhammad Saad Lakhani (24471)
Nimra Humayun (24450)
Talal Khan (25253)
Maham Ahmed (24518)

Supervisors

Academic Supervisor:
Mr. Muhammad Zain Uddin (Lecturer)

Industry Mentor:
Mr. Fahad Rehman (Team Lead, Softech Worldwide)

Project Overview

Conversational speech is a vital component of human communication, but Urdu remains underrepresented in technological solutions for transcription and speaker diarization. Harf Ba Harf addresses this gap by providing a unified, intuitive, and user-friendly platform for accurately transcribing Urdu speech and identifying individual speakers. While such tools exist in other major languages, the lack of localized support places millions at a disadvantage. This project leverages state-of-the-art models to process both public and locally sourced datasets, ultimately delivering a practical tool tailored for Pakistan's growing digital ecosystem.

Purpose

To create an accessible and accurate system for Urdu language transcription and speaker diarization, bridging a major technological gap in regional language processing.

Objective

To develop a single-platform Urdu transcription and diarization tool that ensures high accuracy, user-friendliness, and easy deployment for everyday use.

Contribution

This project pioneers a localized NLP solution for Urdu by combining Whisper's transcription capabilities with PyAnnote's diarization. The result is a cohesive platform that supports inclusive technological advancement in Pakistan.

Technologies Used

- Programming Language & Backend: Python, FASTAPI
- **Speech Processing:** Whisper (for transcription), PyAnnote (for diarization)
- **Mobile Application:** Flutter
- **Database & Hosting:** Firebase
- **Deployment & Access:** Google Colab (with T4 GPU), NGROK for public endpoints
- **Cloud Configuration:** Firebase Remote Config for dynamic link access

Methodology

Audio input is first cleaned and passed through the PyAnnote model for speaker diarization, identifying distinct voices in the recording. Each segmented audio section is then transcribed using Whisper, resulting in a JSON output that maps each speaker to their spoken content. The backend is built using FASTAPI and integrated with Firebase for database and configuration management. The Flutter-based mobile app retrieves output from the model hosted on Google Colab using NGROK links configured in Firebase, offering users a seamless experience.

Specifications

- Model hosted on Google Colab with NVIDIA T4 GPU (15GB VRAM)
- Android-based Flutter application
- Real-time diarization and transcription processing
- JSON output containing speaker-tagged transcripts
- NGROK tunnel for remote model access via Firebase Remote Config

Keywords

Transcription, NLP, Flutter, Diarization Urdu

HDV – AI-Based Video Enhancement System

Group Members

Shahzaib Shoaib (24461)
Muhammad Zauraiz (24486)
Shaikh Zia Saeed (24175)
Syed Muhammad Zuhair Shah (25130)

Supervisors

Academic Supervisor:
Sir Saeed (Visiting Professor)

Industry Mentor:
Mr. Umair Ahmed (Technical Lead Software Development, Securiti)

Project Overview

HDV is an AI-powered desktop application designed to enhance the quality of low-resolution digital video content. The system integrates advanced deep learning models and traditional image processing techniques to perform upscaling, noise reduction, sharpening, and frame interpolation. It provides users with a straightforward graphical interface, eliminating the need for deep technical knowledge while ensuring professional-grade video restoration. The project strikes a balance between performance and output quality, empowering content creators and editors with an efficient offline enhancement tool.

Purpose

To develop a user-friendly, AI-driven tool that enhances video quality using state-of-the-art super-resolution and frame interpolation techniques.

Objective

To provide a desktop application that intelligently improves digital video content through deep learning and image processing, making high-quality video enhancement accessible to non-experts.

Contribution

This project introduces an integrated offline video enhancement solution using AI models such as Real-ESRGAN, SRCNN, and FSRCNN. It offers an end-to-end experience for improving low-quality video content, combining intuitive usability with high-performance processing capabilities.

Technologies Used

- **Programming Languages & Frameworks:** Python, Electron.js, Flask
- **Deep Learning Models:** Real-ESRGAN, SRCNN, SRGAN, FSRCNN, EMA-VFI
- **Image Processing Techniques:** Bilinear/Bicubic Interpolation, Optical Flow, Gaussian Blur, Unsharp Masking
- **Platform:** Desktop Application (Electron.js GUI + Flask backend)

Methodology

The project followed a research-driven iterative development process. Multiple AI and traditional models were evaluated for their effectiveness in video enhancement. Visual quality, processing speed, and hardware efficiency were continuously tested across tasks like upscaling, denoising, and frame interpolation. Experiments focused on identifying optimal model combinations while keeping the system usable on high-end consumer hardware. Continuous comparison ensured the best trade-off between quality and performance in real-world video editing scenarios.

Specifications

- High RAM and GPU recommended for deep learning models (e.g., Real-ESRGAN, SwinIR, DBPN)
- Desktop-based processing; does not require cloud services
- Frame interpolation via Optical Flow
- Offline video enhancement through Electron.js frontend and Flask backend
- Supports integration of CUDA for accelerated performance

Keywords

Video Enhancement, Deep Learning, Super Resolution, CUDA, Video, Frame Interpolation

Indus Sahulat – Real-Time Ambulance Dispatch System

Group Members

Azizullah Khan (24931)
Shaheer Alam (25144)
Syed Muhammad Irtiza Zaidi (23973)
Muhammad Hassaan Farooq (25143)
Muhammad Taha Samji (25186)

Supervisors

Academic Supervisor:
Prof. Sajjad Haider

Industry Mentor:
Ms. Zainab Tariq Soomro (Data Scientist, Indus Hospital and Health Network)

Project Overview

Indus Sahulat is a mobile-first emergency healthcare solution aimed at transforming ambulance dispatch in Pakistan. The project addresses critical delays in emergency medical response by offering an automated, real-time system that notifies nearby hospitals and dispatches ambulances instantly upon patient requests. With integrated live tracking and hospital coordination, the platform ensures that patients receive timely aid and that hospitals are prepared for incoming emergencies. The system is designed to support low-resource environments, ultimately aiming to reduce preventable fatalities and improve healthcare efficiency.

Purpose

To develop an intelligent ambulance dispatch platform that facilitates faster emergency response and strengthens coordination among patients, ambulances, and hospitals.

Objective

To minimize delays in emergency healthcare delivery by building a GPS-enabled mobile application that automates ambulance dispatch and hospital readiness in real time.

Contribution

This project provides a comprehensive, scalable solution for real-time emergency dispatch in Pakistan. It leverages modern communication and location-based technologies to improve patient outcomes and streamline medical logistics in critical situations.

Technologies Used

- **Programming Languages & Frameworks:** Java, Flutter
- **Backend:** Spring Boot, Spring Security
  Database & Caching: PostgreSQL, Redis
- **APIs & Services:** Real-time GPS/location tracking, REST APIs
- **Platform Support:** Mobile and Desktop
- **Security:** Spring Security for access control and user authentication

Methodology

A modular, mobile-first development approach was adopted, focusing on real-time communication and automated workflows. The team divided the system into key functional modules: backend infrastructure, mobile frontend, and hospital/ambulance integration. Live location tracking and alert generation were integrated to ensure timely response and coordination. Agile practices guided the development cycle, allowing the team to iteratively develop, test, and refine features based on stakeholder feedback from hospitals and emergency responders.

Specifications

- Real-time GPS tracking for ambulances
  REST API-driven backend
- Push notifications for dispatch alerts and hospital readiness
- Runs on Android smartphones and desktop browsers
- Stakeholders include patients, ambulance staff, and hospital personnel
- Requires stable internet and GPS access

Keywords

Ambulance Dispatch System, Healthcare Automation, Real-time Tracking, Emergency Medical Services, Mobile Health Tech

IntelliClaim – AI-Powered Vehicle Damage Assessment for Insurance Claims

Group Members

Raine Ramchand (24476)
Beena Ahmed (25240)
Teesha (25247)
Beenish Ahmed (25147)

Supervisors

Academic Supervisor:
Ms. Tasbiha Fatima (Lecturer)

Industry Mentor:
Mr. Jehan Zeb (Project Manager, TPL Insurance)

Project Overview

IntelliClaim is an artificial intelligence-based system aimed at revolutionizing the vehicle insurance claims process through automation and computer vision. The platform leverages YOLOv8 for damage detection, custom ML models for severity classification, and cost prediction to minimize manual intervention and enhance claim accuracy. By enabling insurance employees to assess damages quickly and reliably, IntelliClaim reduces human error, shortens processing time, and enhances customer satisfaction through a modern, tech-enabled approach to claim evaluation.

Purpose

To automate the vehicle insurance claim process using AI-based image analysis, reducing assessment time and increasing accuracy in repair cost estimation.

Objective

To design a user-friendly AI solution that analyzes vehicle damage images, determines severity, and predicts repair costs to assist insurance companies in fast-tracking claim processing.

Contribution

The system integrates advanced computer vision models with a structured database and responsive frontend to provide an end-to-end digital claim management tool. It significantly reduces the manual workload of claim assessors while improving consistency and reliability in the evaluation process.

Technologies Used

- **Frontend:** React TypeScript
- **Backend:** FastAPI
- **Database:** PostgreSQL
- **AI/ML:** YOLOv8 (damage detection), Custom ML models (severity classification and cost estimation)
- **API Integration:** Ngrok (for exposing Colab-hosted AI models)
- **Collaboration & Version Control:** GitHub

Methodology

The project uses a three-tier architecture comprising presentation, application, and data layers. A hybrid development approach was adopted—combining Agile practices with iterative enhancements. AI models are hosted on Google Colab and accessed via Ngrok, enabling high-performance cloud computation while keeping the application locally responsive. Event-driven processing is used to manage tasks such as image upload, analysis, and claim status updates. Role-based access control and secure JWT authentication ensure data security and compliance.

Specifications

- **System Requirements:** Dual-core 2.0 GHz processor, 4–8 GB RAM, 500MB storage
- **OS Compatibility:** Windows 10/11, Linux (Ubuntu 20.04+), macOS
- **External Services:** Google Colab (AI model hosting), Ngrok (secure access)
- **Database Structure:** Tables for users, claims, vehicles, images with relational mappings
- **Security:** JWT-based authentication, role-based access system

Keywords

Artificial Intelligence / Machine Learning / Computer Vision, Insurance Claims Automation, Vehicle Damage Assessment, AI Image Analysis, Repair Cost Prediction

Intelliscore – AI-Based Credit Scoring System

Group Members

Shahmeer Khan (25156)
Nosherwan Shaikh (25237)
Fazul Al Rehman (24961)
Abdur Rehman (24449)

Supervisors

Academic Supervisor:
Miss Tasbiha Fatima (Professor)

Industry Mentor:
Mr. Usman Siddiqui (Shariah Compliance Department, Meezan Bank)

Project Overview

INTELLISCORE is an AI-driven credit scoring platform that aims to modernize and democratize financial evaluations by providing real-time, accurate, and fair credit assessments. Unlike traditional credit scoring systems that rely on narrow datasets, INTELLISCORE utilizes machine learning, predictive analytics, and explainable AI to analyze a broad spectrum of financial data—such as income, liabilities, and spending behavior. The platform is designed as a secure, scalable, and user-friendly web application that serves banks, fintech firms, and underserved borrowers. Through neural network-based modeling and automated natural language reporting (via Gemini AI), the system improves loan approval accuracy, reduces default risks, and fosters financial inclusion with transparency and data-driven insights.

Purpose

To replace outdated credit evaluation models with an AI-powered system that ensures fairness, accuracy, and real-time decision-making for financial institutions and borrowers.

Objective

To develop a secure and scalable web-based credit scoring system using predictive modeling and explainable AI tools that help assess credit risk more effectively and inclusively.

Contribution

INTELLISCORE bridges the gap between complex machine learning and accessible financial technology by integrating neural network predictions, SHAP-based interpretability, and Gemini-generated reports into a responsive, GDPR-compliant credit evaluation platform.

Technologies Used

- **Frontend:** Django Templates, Tailwind CSS, Chart.js, Heroicons
- **Backend:** Django REST Framework, PostgreSQL
- **Machine Learning & Data Tools:** Scikit-learn, Pandas, Custom Neural Network Model
- **AI/NLP Integration:** Gemini API for automated report generation
- **Security & DevOps:** AES-256 encryption, CSRF protection, environment-secured credentials

Methodology

The project followed a structured approach starting with data collection and preprocessing, including normalization and feature engineering. Multiple algorithms were tested, and a neural network was selected based on its performance (lowest MSE). The backend, built with Django, handles form inputs, authentication, and report generation, while the frontend utilizes Tailwind CSS and Chart.js to visualize income trends and credit scores. SHAP values and Gemini AI enhance transparency by generating natural language explanations of credit decisions. The system is optimized for performance with ASGI/WSGI deployment and supports over 10,000 concurrent users.

Specifications

- **Credit Score Output:** 300–850 range via neural network
- **PDF Generation:** Auto-generated reports using Django and Gemini AI
- **Analytics Dashboard:** Real-time charts and comparative tables
- **Security:** AES-256 encryption, GDPR-compliant storage, JWT authentication
- **Performance:** Optimized ORM queries, scalable deployment (ASGI/WSGI)
- **UI/UX:** Responsive, print-optimized, mobile-friendly

Keywords

Predictive Analytics, Neural Networks, Web Application, User Interface Design, Dashboarding and Reporting

Jolyne – AI-Powered Psychiatric Assistant

Group Members

Arbaz Asif (25123)
Muhammad Dawood (25181)
Muhammad Muneeb Faraz (25251)
Yumna Masood (24490)

Supervisors

Academic Supervisor:
Dr. Faisal Iradat (Assistant Professor)

Industry Mentor:
Dr. Sana Durvesh (Psychiatrist, Greenstar Marketing)

Project Overview

Jolyne is an AI-based psychiatric assistant designed to support mental health professionals in evaluating and monitoring patients' emotional well-being. Leveraging natural language processing (NLP), computer vision, and audio signal analysis, the platform offers a multi-modal approach to sentiment and emotion recognition. By analyzing patient inputs—text, facial expressions, and voice—the system provides real-time emotional reports to aid in psychiatric diagnosis and treatment planning. The platform enhances traditional psychiatric practices by introducing AI-assisted, data-driven insights, improving diagnostic precision and monitoring capabilities in a privacy-conscious and intuitive web interface.

Purpose

To assist psychiatrists in real-time emotional and psychological assessments using AI-powered multi-modal emotion recognition.

Objective

To provide an intelligent web-based assistant that uses text, voice, and facial inputs to evaluate a patient’s emotional state and deliver comprehensive diagnostic reports to support mental health professionals.

Contribution

Jolyne bridges the gap between traditional psychiatry and advanced AI by offering a unified tool that streamlines emotion recognition across multiple inputs. It aids in more accurate assessments and long-term tracking, ultimately supporting better mental health outcomes.

Technologies Used

- **Programming Languages:** Python, JavaScript
- **Libraries & Frameworks:** TensorFlow, PyTorch, Hugging Face Transformers (BERT, RoBERTa), OpenCV, NLTK, DeepFace
- **Audio Processing:** Librosa, SpeechRecognition, PyAudio
- **Version Control:** GitHub
- **Web Development:** Python-based web interface

Methodology

A modular, data-driven approach was used, starting with a needs assessment of psychiatric workflows. Emotion classification models were trained using public datasets like GoEmotions (text), RAVDESS (audio), and FER2013 (facial expressions).

- **Text Input:** Preprocessed using NLTK and analyzed via fine-tuned BERT/RoBERTa models.
- **Audio Input:** Features like MFCC, pitch, and energy extracted via Librosa, then classified using ML models.
- **Visual Input:** Facial expressions processed using OpenCV or DeepFace for real-time emotion recognition.
  A unified web platform accepts inputs and generates multi-modal emotional reports, integrating model predictions into a comprehensive output. The tool was iteratively tested and validated using labeled data for each modality.

Specifications

- **OS Compatibility:** Windows 10, macOS, Linux
- **RAM:** 8 GB minimum (16 GB recommended)
- **Processor:** Intel i5+, Apple M1, or AMD equivalent
  GPU: Recommended for faster inference (e.g., NVIDIA CUDA)
- **Hardware:** Standard webcam and microphone
- **Tools & Libraries:** Python 3.8+, Hugging Face Transformers, OpenCV, Librosa, NLTK, SpaCy, SpeechRecognition
- **Security:** Local processing and secure data handling for patient privacy

Keywords

Emotion Recognition, Deep Learning, Web Application, Multimodal Analysis, Mental Health Technology

LLM Auto Pentester – Autonomous Penetration Testing using Large Language Models

Group Members

Ahmar Ayaz (24468)
Aqib Mahmood (25235)
Muhammad Faraz (25142)
Subata Naveen Khan (18119)
Ali Asghar Hussain (24525)

Supervisors

Academic Supervisor:
Dr. Faisal Iradat (Assistant Professor)

Industry Mentor:
Mr. Pete Herzog (Co-Founder, ISECOM)

Project Overview

LLM Auto Pentester is an experimental cybersecurity platform designed to automate penetration testing through the dynamic reasoning capabilities of large language models (LLMs). Targeted at small and medium businesses with limited cybersecurity budgets, the system simulates a real-world pentest scenario without the need for a professional pentester. It autonomously performs scans, analyzes results, and decides the next course of action using Retrieval-Augmented Generation (RAG) with tools like OpenVAS and Kali Linux. By continuously evaluating findings and adapting the strategy, it generates detailed reports highlighting vulnerabilities, severity levels, and suggested remediations. The project explores the feasibility of deploying LLMs for intelligent, affordable, and automated vulnerability assessments.

Purpose

To research and develop an AI-powered system capable of autonomously performing penetration testing using LLMs, reducing dependency on expensive human experts.

Objective

To create an LLM-based automated pentesting framework that dynamically analyzes scan results, makes decisions, and performs actions in a continuous feedback loop—mimicking the actions of a skilled cybersecurity expert.

Contribution

This project contributes a pioneering approach to cybersecurity automation by integrating RAG, Dockerized environments, and LLMs for end-to-end vulnerability assessment. It enables scalable and cost-effective solutions for SMBs seeking to improve their security posture without hiring pentesting professionals.

Technologies Used

- **Languages & Frameworks:** Python, Django
- **LLM & AI Tools:** Langchain, RAG, LLAMA, Mistral, OpenAI (via APIs)
  Pentesting Tools: OpenVAS/GVM, Kali Linux
- **Infrastructure & Integration:** Docker, Google Colab, GROQ, ChromaDB, Elastic Search
- Search/Indexing: Elastic Search API
- **Vector Storage:** ChromaDB for RAG pipelines

Methodology

A target web server or application is provided to the system, which then initiates a vulnerability scan using a Dockerized Kali Linux instance. The results are fed into the LLM (accessed via GROQ or hosted using Google Colab), which uses Retrieval-Augmented Generation (RAG) to determine the next step. Commands are dynamically selected, executed, and the results fed back into the system, allowing continuous refinement. At every step, vulnerability findings are logged, and a report is generated. The system halts when the LLM determines the assessment is complete, after which a final comprehensive report is compiled with severity ratings and remediation suggestions.

Specifications

- **System Requirements:**
  - Windows/Linux OS
  - Minimum 16GB RAM, 40GB storage
  - 8-core CPU (8th Gen or higher)
    Docker compatibility
- **Cloud/External Services:**
  - OpenAI & Mistral APIs
  - LLAMA3 via GROQ
  - Google Colab Pro or equivalent for LLM processing
  - Elastic Search for data indexing and storage
- **Storage Requirements:** ChromaDB vector files, scan result logs
- **Security Considerations:** Secure API key handling, isolated containers

Keywords

Cyber Security LLM, Pentesting, RAG, Automation

MindTrack – Adaptive Intelligence Assessment Platform

Group Members

Nafisa (25239)
Laiba Zubair Ellahi (24472)
Abdullah Shahood (24714)
Misbah Samee (25159)
Hammad ul Haq (25252)

Supervisors

Academic Supervisor:
Mr. Muhammad Zain Uddin (Lecturer)

Industry Mentor:
Mr. Zubair Hamid (Department Head – Technology & Automation, Jubilee Life Insurance)

Project Overview

MindTrack is an adaptive intelligence testing platform developed to assess the cognitive abilities of university students and recent graduates. By utilizing reinforcement learning—specifically the UCB (Upper Confidence Bound) algorithm—the system dynamically adjusts the difficulty of visual, auditory, and text-based exercises based on real-time user performance. Designed to serve both individuals and recruiters, MindTrack provides a dual-interface experience: students receive personalized feedback to identify their strengths and weaknesses, while recruiters gain access to detailed performance analytics to inform hiring decisions. The platform supports personalized evaluations and delivers actionable insights through interactive dashboards.

Purpose

To provide a personalized, adaptive cognitive assessment tool that benefits students through self-awareness and recruiters through informed talent evaluations.

Objective

To build an intelligent, user-responsive platform using reinforcement learning to offer dynamic IQ testing and performance insights via a seamless mobile and web-based experience.

Contribution

MindTrack introduces a data-driven, AI-enhanced assessment system tailored for recruitment and personal development. Its integration of reinforcement learning in cognitive testing is a step toward more accurate, individualized evaluation experiences.

Technologies Used

- **Mobile App Development:** Flutter
- **Frontend Web Dashboard:** ReactJS
- **Backend:** FastAPI (Python), Node.js
- **Database:** PostgreSQL
- **Algorithms & Libraries:** Python (UCB Bandit), NumPy
- **DevOps & Integration:** REST APIs, GitHub

Methodology

The development of MindTrack followed Agile methodology with iterative sprint cycles and continuous feedback integration. The system employs a modular monolithic architecture with two main interfaces: a mobile app for users and a web-based dashboard for recruiters. Both interfaces are connected via a centralized FastAPI backend. The core of the adaptive testing mechanism is a UCB algorithm implemented in Python, which determines question difficulty based on user interactions. Data collection, test calibration, and performance reporting were refined through real-time testing and collaborative teamwork.

Specifications

- **Platforms:** Mobile (Flutter for Android/iOS), Web (ReactJS)
- **Backend APIs:** FastAPI (Python), Node.js (module-specific support)
- **Database:** PostgreSQL
- **Algorithm:** UCB Bandit for adaptive logic
- **Minimum Requirements:** Compatible with modern smartphones and browsers, PostgreSQL-compatible hosting environment

Keywords

Personalized User Experience, Adaptive IQ Assessment, Cognitive Evaluation, Recruitment-Oriented Evaluation, Data-Driven Performance Insights

NeuralTrace – AI-Powered Memory Assistant for Cognitive Disorders

Group Members

Anfa Majid (25155)
Moiz Zulfiqar (24543)
Abdullah Maqsood (24448)
Umair Ahmed (24377)

Supervisors

Academic Supervisor:
Dr. Muhammad Saeed (Visiting Professor)

Industry Mentor:
Mr. Umair Nazeer (Technical Lead Software Development, Securiti.ai)

Project Overview

NeuralTrace is an AI-enabled mobile application designed to support individuals suffering from memory-related disorders such as Alzheimer's and dementia. Acting as a personal cognitive assistant, the app helps users recall interactions, locate misplaced items, manage daily routines, and remain safe through features like geofencing and caregiver alerts. Utilizing advanced voice and visual processing, wearable integration, and Retrieval-Augmented Generation (RAG), NeuralTrace delivers real-time assistance while ensuring user data remains secure through encrypted communication protocols. The app blends machine learning, natural language processing, and context-aware computing into a single user-friendly platform, aiming to enhance daily independence for those facing cognitive challenges.

Purpose

To assist individuals with memory disorders by providing AI-powered support for daily task management, object recognition, and situational awareness through a mobile application.

Objective

To develop a secure and intelligent mobile platform that leverages speech, vision, and location data to provide real-time cognitive assistance and safety support.

Contribution

NeuralTrace offers a comprehensive digital companion for individuals with memory impairments, combining object detection, voice-based recall, and geofencing in a single mobile application. The project introduces an innovative application of RAG for memory assistance and emphasizes privacy-first design in healthcare-related AI applications.

Technologies Used

- **Programming Languages:** Python, JavaScript
- **Mobile Development:** React Native
- **AI & ML Libraries:** LangChain, PyTorch, OpenCV, TensorFlow, Transformers
- **Backend & APIs:** FastAPI
- **Database:** PostgreSQL with vector database extensions
- **Integration:** Wearable device support (e.g., smart glasses, headphones)
- **Security:** HTTPS, TLS encryption for secure communication

Methodology

The project employed a modular and collaborative development approach. Functional modules included voice command processing, object detection via smartphone cameras, and geofencing through location APIs. A Retrieval-Augmented Generation (RAG) pipeline was integrated for conversational memory recall. Real-time notifications and text-to-speech (TTS) enabled smooth interaction between the system and users. The app was built using React Native for cross-platform compatibility and FastAPI for efficient backend communication. All members contributed equally to the design, implementation, and testing phases.

Specifications

- **OS Compatibility:** Android 13+, iOS 14+
- **Hardware Requirements:** Smartphone, optional wearable devices
- **Languages & Tools:** Python, React Native, FastAPI
- **ML Features:** Object recognition, voice interaction, RAG-based memory recall
- **Security:** TLS encryption, local + cloud data processing
- **Database:** PostgreSQL with vector extensions for contextual search

Keywords

AI Memory Assistance, Natural Language Processing, Mobile Application, Object Detection, Geofencing

OrthoVision – AI-Powered Dental Diagnostic and Management System

Group Members

Danial Habib Moten (24437)
Muhammad Abdullah Khan (24507)
Zain Anis (25127)

Supervisors

Academic Supervisor:
Ms. Abeera Tariq (Lecturer)

Industry Mentor:
Mr. Farrukh Saif (Head of Production – Web & Apps, Salsoft Technologies)

Project Overview

OrthoVision is an AI-driven web application designed to revolutionize dental diagnostics and clinic management in Pakistan. Utilizing deep learning techniques and advanced image analysis, the system can automatically detect and classify conditions such as caries, crowns, periapical lesions, and root canals from periapical dental X-rays. In addition to AI-based diagnostic support, the platform offers comprehensive hospital and clinic management functionality including real-time appointment scheduling, report management, and role-based dashboards for patients, doctors, and admins. OrthoVision aims to bridge the accessibility and efficiency gaps in dental healthcare, offering a scalable, secure, and user-centric solution that supports both urban and underserved communities.

Purpose

To assist dentists in delivering more accurate and efficient diagnoses using AI-powered dental X-ray analysis and to streamline clinic operations through an integrated hospital management system.

Objective

To develop a web-based, AI-integrated platform that combines diagnostic automation with real-time clinic operations to improve both clinical decision-making and administrative workflows in dental practices.

Contribution

OrthoVision introduces an intelligent, real-time platform that blends object detection models with dental healthcare workflows. It empowers medical professionals with AI-assisted diagnostic tools and enables digital clinic transformation through centralized appointment, patient, and report management systems—offering a meaningful advancement in Pakistan's healthcare technology ecosystem.

Technologies Used

- **Frontend:**
  - Next.js (React framework)
  - TypeScript, Tailwind CSS, React 19, Axios
  - UI Libraries: Radix UI, lucide-react, react-day-picker, sonner
- **Backend:**
  - Django, Django REST Framework, Channels
  - PostgreSQL, Redis
  - Authentication: JWT with Role-Based Access Control (RBAC)
- **AI/ML & Image Processing:**
  - PyTorch, TensorFlow, OpenCV, Scikit-learn
  - YOLO (Ultralytics), custom-trained models for dental image segmentation
- **Infrastructure & Tools:**
  - Kaggle (GPU model training), Git (version control)

Methodology

The project was developed using a modular monolithic architecture with separate services for core operations, appointment management, AI-based X-ray analysis, notifications, and authentication.

Frontend used modular Next.js architecture for scalability, supporting API communication and real-time updates via WebSocket.
Backend was built using Django with services organized around domain responsibilities (e.g., X-ray processing, appointment booking, clinic operations).
AI Integration: YOLO variants were trained and tested on custom dental datasets, with YOLOv11s selected for production based on segmentation accuracy.
Real-Time Systems: Redis and WebSockets enabled instant alerts for appointments and X-ray outcomes.
Security: JWT tokens, RBAC, rate limiting, input validation, and transactional data handling ensured system integrity.

Specifications

- **Software:**
  - Python 3.x, Django, PostgreSQL, Redis
  - Django REST and WebSocket APIs
  - YOLO integration for image segmentation
- **Hardware:**
  - GPU-enabled machines for AI training and inference
  - Standard web servers for deployment

Keywords

Dental X-ray Analysis,Healthcare Web Application,Medical Image Segmentation,Hospital Management System

Rawangi – Smart Public Transport Companion for Karachi

Group Members

Maaz Karim (24503)
Muhammad Musab Iqbal (24495)
Muhammad Bilal (24443)
Yahya Ahmed Khan (24442)

Supervisors

Academic Supervisor:
Dr. Imran Rauf (Assistant Professor)

Project Overview

Rawangi is a smart public transport companion app built to modernize the commuting experience in Karachi. Addressing the city's fragmented and unreliable bus system, Rawangi provides real-time features such as live bus tracking, accurate estimated arrival times (ETAs), and dynamic occupancy levels. The platform serves three user groups: passengers (to plan and track trips), drivers (to report delays/incidents), and administrators (to manage routes and system configurations). Using GPS technology, routing algorithms (Dijkstra's and A*), and cloud-hosted mobile and web technologies, Rawangi creates an integrated ecosystem that enhances public transport accessibility, safety, and efficiency—ultimately encouraging sustainable urban mobility.

Purpose

To create a smart, real-time navigation and tracking platform that improves public transportation efficiency and usability in Karachi.

Objective

To assist commuters with real-time bus data, reduce congestion and wait times, and streamline city-wide transport operations using a unified platform for passengers, drivers, and admins.

Contribution

Rawangi brings structure and innovation to Karachi's disorganized public transport system. Through real-time tracking, route optimization, and multi-user coordination, it not only enhances commuter experience but also supports more sustainable and data-informed transit operations in the city.

Technologies Used

- **Frontend & Mobile:**
  - Dart (Flutter) – Mobile application
  - React, Next.js – Admin panel
- **Backend:**
  - Node.js with Express.js
  - RESTful APIs and WebSockets
- **Database & Cloud Services:**
  - Firebase (Realtime Database, Authentication)
  - Mapbox (routing and geolocation services)
  - OpenStreetMap
  - Hosting: AWS, Vercel

Methodology

The project followed a modular client-server architecture and Agile development lifecycle. Key subsystems included mobile apps for passengers and drivers, GPS-based location services, and a web-based admin panel. Real-time responsiveness was achieved using Firebase and WebSocket communication, while routing decisions were guided by Dijkstra's and A* algorithms. Functional testing, error handling, and performance validations ensured system stability and user satisfaction. Each stakeholder interface was developed with scalability and real-world adaptability in mind.

Specifications

- **Architecture:** Modular client-server
- **APIs:** REST APIs + WebSockets for real-time updates
- **Routing:** Custom logic using Dijkstra's and A* algorithms
- **Data Sources:** Mapbox API (for GUI), OpenStreetMap
- **Platform Support:** Android, Web Admin Panel
  Cloud Services: AWS/Vercel for backend hosting

Keywords

Routing Algorithms, Public Transport, Mobile Application, Real-Time Tracking, Urban Mobility

RecruitWise – AI-Powered Conversational Interviewing & Resume Screening Engine

Group Members

Syed Shaheer Sabri (24522)
Zuhair Arif (24504)
Khizar Asad (23988)
Muhammad Danish Raza (24796)
Muhammad Humayun Raza (24464)

Supervisors

Academic Supervisor:
Mr. Imran Rauf (Lecturer)

Industry Mentor:
Mr. Nabeel Farooqui (Senior Software Engineer, Folio3)

Project Overview

RecruitWise is an intelligent recruitment automation system that transforms traditional applicant screening and interviewing through the use of advanced natural language processing and proctoring technologies. The system combines semantic resume screening, AI-driven conversational interviews powered by LLAMA3.3 via Groq Cloud, and integrated anti-cheating protocols. It uses Retrieval-Augmented Generation (RAG) to personalize interview questions based on each candidate's resume and job description. The interviews, conducted via TTS/STT-enabled voice interaction, are automatically transcribed and evaluated using a scoring algorithm mapped to specific job skills. In parallel, webcam and browser-based proctoring ensures interview integrity, identifying suspicious behavior using object detection (YOLO V8), gaze tracking, and screen enforcement. RecruitWise offers recruiters comprehensive candidate rankings along with recorded footage for manual review, combining automation, fairness, and accountability in one scalable platform.

Purpose

To enhance the recruitment process through automated AI interviews and smart resume screening while ensuring integrity through integrated proctoring.

Objective

To develop a secure, intelligent recruitment engine that combines LLM-based dynamic interviews, AI-driven scoring, and cheating detection to streamline and strengthen hiring decisions.

Contribution

RecruitWise provides a robust, scalable, and innovative hiring solution that goes beyond static resume reviews and recorded responses. By integrating real-time AI interaction, semantic scoring, and multi-layered proctoring, it brings structure, personalization, and trust to the virtual recruitment pipeline.

Technologies Used

- **Frontend:** React.js, Tailwind CSS
- **Backend:** Nest.js, FastAPI
- **Databases:** MongoDB, Redis (for refresh tokens & in-memory caching)
- **AI/ML & NLP:**
  - LLAMA3.3 via Groq Cloud
  - RAG (Retrieval-Augmented Generation)
  - Resume format detection (custom-trained model)
  - Scoring Engine: NLP-based candidate ranking
- **Speech & Vision:**
  - TTS/STT: Eleven Labs APIs
  - Object Detection: YOLO V8
  - Gaze Tracking: MediaPipe
- **Infrastructure & Deployment:**
  - Docker (containerization)
  - Vercel (frontend), Azure Containers (backend & microservices)
  - AWS S3 (cloud storage for resumes and recordings)
- **Security:** JWT (access & refresh tokens), HTTPS/WSS, Redis

Methodology

Developed using Agile methodology with bi-weekly sprints. Core modules—resume screening, AI interviewing, scoring system, and proctoring—were iteratively developed and refined. RESTful APIs (Nest.js, FastAPI) handled candidate data, TTS/STT sessions, and score computations. Interview logic was enhanced using RAG with personalized follow-ups. Cheating detection was powered by YOLO and MediaPipe, while browser-based tracking ensured active tab monitoring. The final system was stress-tested and designed for scalability, with secure token-based authentication and support for screen recordings.

Specifications

- **System Requirements (Candidate):**
  - Modern browser, webcam, mic, stable internet, single screen
- **System Requirements (Recruiter):**
  Modern browser, stable internet
- **Processing Requirements:**
  - CUDA-enabled GPU for proctoring (local)
- **APIs Used:**
  - Groq Cloud API (LLAMA3.3), AWS S3, Eleven Labs, custom REST APIs
- **Security & Authentication:**
  - JWT tokens, Redis token storage, secure WSS channels
- **Accessibility Features:**
  - Dark/Light Mode, text scaling, colorblind compatibility

Keywords

Large Language Model, AI-Interviews, Recruitment, Web Application, Containerization

Satellite-Based Crop Yield Predictor – CNN-Driven Agricultural Forecasting for Pakistan

Group Members

Moeen Haider (24505)
Ahmed Raza (25255)
Muddasir Rizwan (24494)
Hassan Atiq (24517)

Supervisors

Academic Supervisor:
Dr. Faisal Iradat (Assistant Professor)

Industry Mentor:
Mr. Arbaz Khan (Senior Software Engineer, Folio3)

Project Overview

Satellite-Based Crop Yield Predictor is a deep learning-based system designed to estimate crop yields (in kg/ha) across major agricultural zones in Pakistan using raw satellite imagery. The system focuses on key crops like rice, wheat, and maize, leveraging multi-temporal Sentinel-2 satellite data and NDVI-derived insights to forecast yields. The pipeline includes cloud masking, image compositing across four crop growth stages, and feature extraction using NDVI metrics. These image stacks are input into a custom-built Convolutional Neural Network (CNN) trained to regress yield values. With a focus on scalability, openness, and regional specificity, the project aims to offer a robust, accessible decision-support tool for agricultural planners and policymakers in Pakistan.

Purpose

To provide data-driven yield predictions for Pakistan's key crops using satellite imagery and deep learning, improving agricultural planning and food security strategy.

Objective

To develop a CNN-based open-source crop yield forecasting tool that leverages NDVI features from Sentinel-2 imagery, helping policymakers and farmers make informed decisions.

Contribution

This project delivers a localized, deep learning model that integrates satellite data and vegetation indices to estimate yields accurately. It fills a critical technological gap in Pakistan's agricultural sector and enhances the usability of satellite data for national food security initiatives.

Technologies Used

- **Frontend:** React
  Backend: FastAPI
  Database: PostgreSQL
- **Machine Learning & Processing:**
  - PyTorch (for CNN model)
  - NDVI generation via Sentinel-2 band computation
  - Google Earth Engine (imagery acquisition)
- **Others:**
  Kaggle (T4 GPU training)
  APIs for real-time model interaction

Methodology

The project followed a four-phase pipeline:

Data Collection & Preprocessing:
Data from HLSL30 Sentinel sources (Punjab & Sindh in Pakistan; Punjab in India)
Cloud masking, temporal compositing, spatial clipping, and normalization

Feature Extraction:
NDVI values computed using Red (Band 4) and NIR (Band 5)
Seasonal composites (t1–t4) stacked into 4-channel tensors for CNN input

Model Architecture:
Custom CNN built in PyTorch, regressing stacked NDVI data to yield values
Hyperparameter tuning and validation on multi-year regional data

Visualization & API Deployment:
Model exposed via FastAPI
Integrated with a React frontend and PostgreSQL database for report storage

Specifications

- **System Requirements:**
  - Prediction can run on Intel i5 (8th Gen) with 8GB RAM
  - Training performed on Kaggle T4x2 GPU environment
- **APIs Used:**
  - Google Earth Engine for imagery
  - FastAPI for model deployment
  - Web Hosting: Browser-based access supported

Keywords

Convolutional Neural Networks, Crop Yield Prediction, Satellite Imagery,Web Application, Agricultural Forecasting

SocialBrain – AI-Driven Trend-Based Social Media Content Scheduler

Group Members

Nawab Ahmed (25048)
Sameer Hussain (25242)
Yasir Jalal (25258)
Muhammad Latif (25120)

Supervisors

Academic Supervisor:
Dr. Tariq Mahmood (Professor)

Industry Mentor:
Mr. Shakir Ghani (Co-Founder, Social Champs)

Project Overview

SocialBrain is an intelligent content creation and scheduling platform designed to assist social media creators and marketers in developing 30-day content plans based on real-time trends. Unlike traditional content planning tools, SocialBrain combines large language models (LLMs) with advanced trend analysis to generate personalized posts, captions, media ideas, and optimal posting schedules. The system outputs content in grid, list, or calendar views and integrates with popular social platforms to automate scheduling. With AI models like GPT, BERT, and DALL·E, along with NLP techniques like RAKE, NER, and VADER, SocialBrain provides a scalable, data-driven solution to streamline content strategy, save time, and boost engagement.

Purpose

To automate and personalize trend-based content creation for social media using AI, enabling creators to efficiently plan and post relevant content.

Objective

To build a smart platform that uses real-time trends and NLP to generate 30-day social media calendars complete with text, media suggestions, and post timings.

Contribution

SocialBrain redefines content planning by combining prompt engineering, social listening, and transformer-based models into an automated tool for consistent, timely, and high-engagement content creation. It eliminates guesswork in social media strategy and reduces manual effort through smart automation.

Technologies Used

- **Frontend:** React.js, Redux, Tailwind CSS, Jest
- **Backend:** Node.js, Express.js
- **Database:** MongoDB (Cloud Atlas)
- **AI/ML Tools:**
  - Large Language Models: GPT, BERT, GEMINI
  - Image Generation: DALL·E
  - NLP Techniques: TF-IDF, RAKE, Named Entity
  - Recognition (NER), VADER sentiment analysis
- **APIs & Scheduling:**
  - Social Media APIs: Facebook, Instagram, Twitter,
  - LinkedIn
  - Google Trends API
  - Cron Jobs for post automation

Methodology

The project followed a modular client-server architecture and Agile development lifecycle. Key subsystems included mobile apps for passengers and drivers, GPS-based location services, and a web-based admin panel. Real-time responsiveness was achieved using Firebase and WebSocket communication, while routing decisions were guided by Dijkstra's and A* algorithms. Functional testing, error handling, and performance validations ensured system stability and user satisfaction. Each stakeholder interface was developed with scalability and real-world adaptability in mind.

Specifications

- **Architecture:** Modular client-server
- **APIs:** REST APIs + WebSockets for real-time updates
- **Routing:** Custom logic using Dijkstra's and A* algorithms
- **Data Sources:** Mapbox API (for GUI), OpenStreetMap
- **Platform Support:** Android, Web Admin Panel Cloud Services: AWS/Vercel for backend hosting

Keywords

Routing Algorithms, Public Transport, Mobile Application, Real-Time Tracking, Urban Mobility

VidSense – NLP-Based Intelligent Video Summarization and Accessibility Tool

Group Members

Muhammad Wasay (24497)
Muhammad Bilal Taha (25132)
Ali Iqbal (24529)
Muhammad Muaz Arif (25125)
Syed Bilal Ali (24501)

Supervisors

Academic Supervisor:
Dr. Sayeed (Professor)

Industry Mentor:
Mr. Umair (Technical Lead Software Development, Securiti.ai)

Project Overview

VidSense is an AI-powered video assistant platform developed to enhance accessibility and efficiency in consuming educational and informative video content. By leveraging advanced Natural Language Processing (NLP) techniques, VidSense generates extractive video highlights, enabling users to consume long videos more effectively. It supports both Arabic and English languages and includes multilingual transcription, topic segmentation, semantic scoring, and dubbing features. Additionally, it offers chatbot-assisted navigation (powered by Retrieval-Augmented Generation), subtitle generation, and podcast-style question-answer audio outputs. All processing is done locally, ensuring privacy and independence from cloud services. VidSense significantly reduces viewing time and enhances content understanding for learners, researchers, and content creators alike.

Purpose

To reduce time spent on watching long-form educational videos by automatically generating intelligent video highlights and interactive features using local NLP and LLM processing.

Objective

To create a privacy-respecting, locally-run tool that transcribes, segments, summarizes, and enriches videos with dubbing, subtitling, and conversational exploration.

Contribution

VidSense provides an all-in-one, locally hosted system that combines speech-to-text, semantic segmentation, language modeling, and media editing to deliver accessible, interactive, and time-efficient educational video experiences.

Technologies Used

- **Frontend:**
  React.js (Web application only)
- **Backend:**
  - Python with FastAPI
  - MoviePy & FFmpeg for video editing
- **AI/ML & NLP Models:**
  - Speech-to-Text: Whisper v3-large-turbo
  - LLM: deepseek-r1:7b (via Ollama)
  - Embedding: paraphrase-multilingual-MiniLm-L12-v2
  - Topic Modeling: BERTopic
  - Language Detection: xlm-roberta-base-language-detection
- **Data Storage:** Fully local (no external cloud or APIs)

Methodology

VidSense follows a sequential multi-module pipeline. Users upload a video (or input a YouTube link under 30 mins), which is transcribed using Whisper. The transcription is then segmented and scored using PageRank and sentiment analysis. Top-ranked segments are selected and merged into a summarized highlight reel using MoviePy. Additional functionality is powered by the locally-run LLM (deepseek-r1:7b), which generates custom outputs such as chatbot responses, dubbing, subtitles, and podcast-style Q&A. Parallel processing ensures the system operates efficiently without needing cloud infrastructure.

Specifications

- **Minimum Requirements:**
  - Modern multi-core CPU, 16GB RAM
  - 20GB storage for models and processing
  - Optional GPU (CUDA-compatible) for Whisper and LLM inference
- **Languages & Tools:**
  - Python 3.13+, FastAPI backend, React.js frontend
  - FFmpeg, MoviePy for video editing
  - Fully local processing except for internet requirement during YouTube downloads

  Performance:
  - 30-minute transcription: ~1–2 minutes
  - Highlight generation and chatbot: <1 minute

Keywords

NLP, Semantic Analysis, RAG, Video Summarization, Multilingual Accessibility,

Vroo – Secure & Sustainable Ride-Sharing Platform for Trusted Communities

Group Members

Humera Raheel (24446)
Noor us Sabah (25173)
Hritika Rai (24513)
Manahil Aamir (24441)

Supervisors

Academic Supervisor:
Dr. Syed Ali Raza (Assistant Professor)

Industry Mentor:
Mr. Nabeel (Senior Software Engineer, Folio3)

Project Overview

Vroo is a closed-community ride-sharing platform developed to address urban mobility challenges by connecting individuals from the same organization or institution. It promotes eco-friendly and cost-effective commuting by matching drivers and riders based on shared routes and organizational affiliation. Unlike public carpooling apps, Vroo ensures high trust and safety through domain-based email verification, allowing only official members of a specific organization (e.g., universities, corporate offices) to sign up. With real-time ride-matching, secure authentication, and map-based coordination, Vroo offers a smart and efficient solution tailored to the culture of shared mobility within trusted environments.

Purpose

To offer a safe and convenient ride-sharing system exclusively for verified organizational members, reducing travel costs, carbon emissions, and urban congestion.

Objective

To design and implement a secure, real-time mobile carpooling app with intelligent route matching and user verification through official email domains.

Contribution

Vroo introduces a community-driven, closed-network ride-sharing model that blends sustainability with safety. Its tailored design for organizations improves user trust, simplifies coordination, and builds environmentally conscious commuting habits among professionals and students.

Technologies Used

- **Frontend & Mobile Development:** Flutter, Dart
- **Backend Development:**
  - Python (Flask), Node.js (Express.js)
  - MongoDB with Mongoose ODM
- **Authentication & Cloud:**
  - Firebase Admin SDK
  - JWT (JSON Web Tokens)
  - Azure, Vercel for hosting
- **APIs & Tools:**
  - Google Maps API
  - Next.js for web components

Methodology

Vroo followed an Agile development methodology with multiple sprints focused on core features: secure authentication, email verification, ride-matching, live location tracking, and notifications. Regular sprint reviews, testing cycles, and stakeholder feedback ensured continuous iteration and functional enhancement throughout the development lifecycle.

Specifications

- **Supported Platforms:** Android (minimum version: Android 8.0 / API 26)
- **Internet:** Required for all major features (ride-matching, maps, notifications)
- **Permissions Required:**
  - Location services
  - Internet access
  - Push notifications
- **Security:** Domain-based signup + JWT authentication

Keywords

Algorithms, Google Maps API, Mobile Application, Firebase, Cloud Integration

IBA

70 YEARS & BEYOND

Leadership ▪ Excellence ▪ Impact

Main Campus
University Enclave, University Road,
Karachi - 75270 Pakistan
Phone : +92-(21) 3810-4700

City Campus
Plot # 68 & 88 Garden / Kayani
Shaheed Road, Karachi - 74400 Pakistan
Phone : +92-(21) 3810-4701

☎ +92 21 111 422 422 www. iba.edu.pk ibakhiofficial