My Academy

A Major Project Report Submitted in Partial Fulfillment for the Award of the Degree of

Bachelor of Technology in

Computer Science and Engineering

To

Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

Submitted By:

Smita Singh (2100100100170)

Sneha Jaiswal (2100100100171)

UNDER THE SUPERVISION OF

Mr. MOHIT SAXENA

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNITED COLLEGE OF ENGINEERING & RESEARCH, PRAYAGRAJ

May 2025

Candidate's Declaration

We, hereby declare that the project entitled "My Academy" submitted by us in partial fulfillment of the requirement for the award of degree of the B. Tech. (Computer Science & Engineering) submitted to Dr. A.P.J. Abdul Kalam Technical University, Lucknow at United College of Engineering and Research, Prayagraj is an authentic record of our own work carried out during a period from June, 2024 to May, 2025 under the guidance of Mr.MOHIT SAXENA (Guide name, Designation, Department of Computer Science & Engineering). The matter presented in this project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Signature of the Student
Smita Singh(2100100100170.)

Signature of the Student
Sneha Jaiswal (2100100100171)

Certificate

This is to certify that the project titled “My Academy” is the bona fide work carried out by Sneha Jaisawal (2100100100171) & Smita Singh(2100100100170)in partial fulfillment of the requirement for the award of degree of the B. Tech. (Computer Science & Engineering) submitted to Dr. A.P.J Abdul Kalam Technical University, Lucknow at United College of Engineering and Research, Prayagraj is an authentic record of their own work carried out during a period from June, 2024 to May, 2025 under the guidance of Mr.Mohit Saxena Designation, Department of Computer Science & Engineering. The Major Project Viva-Voce Examination has been held on ________________________.

Signature of the Guide
Mr. Mohit Saxena

Signature of Project Coordinator
Mr. Shyam Bahadur Verma

Signature of the Head of Department
Dr. Vijay Kumar Dwivedi

Acknowledgement

We express our sincere gratitude to the Dr. A.P.J Abdul Kalam Technical University, Lucknow for giving us the opportunity to work on the Major Project during our final year of B.Tech. (CSE) as an important aspect in the field of engineering.

We would like to thank Dr. Swapnil Srivastava, Principal and Dr. Vijay Kumar Dwivedi, Dean Academics & Head of Department, CSE at United College of Engineering and Research, Prayagraj for their kind support.

We also owe our sincerest gratitude towards Mr. Mohit Saxena for his valuable advice and healthy criticism throughout our project which helped us immensely to complete our work successfully.

We would also like to thank everyone who has knowingly and unknowingly helped us throughout our work. Last but not the least, a word of thanks for the authors of all those books and papers which we have consulted during our project work as well as for preparing the report.

Abstract

The fashion industry is swiftly transforming with the incorporation of technology, necessitating the development of digital platforms that address the ever-changing needs of budding models, fashion aficionados, and designers. MY Academy is an all-encompassing online platform aimed at transforming the fashion education experience by providing interactive and AI-driven tools. The site includes a smart chatbot that offers insights into beauty contests, fashion tips, and grooming advice, ensuring users receive timely guidance and updates. Moreover, the Ramp Walk Tutor uses pose detection technology to assess and enhance a user's walking technique, speed, and posture, delivering tailored feedback for professional growth. To maintain a competitive edge, the platform also features a Trend Demand Forecasting module that employs machine learning to anticipate future fashion trends based on past and current data. Additionally, MY Academy compiles information on upcoming pageants, designer contacts, and event updates, acting as a comprehensive resource for fashion education and career advancement. This initiative seeks to close the gap between fashion training and technology, enabling users to learn, adapt, and thrive in the fashion sector.

.

Table of Contents

Chapter 1: Introduction

The integration of fashion and technology has changed how individuals approach style, grooming, and professional modeling. In the digital age of the present time, the desire for platforms not merely imparting information but also enabling useful tools for education and self-development in the field of fashion has grown. Meeting this requirement, MY Academy has been established as a cutting-edge and interactive web application serving the changing needs of fashionistas, would-be models, and professionals alike.

My Academy is a holistic platform consisting of three fundamental elements. First, an AI chatbot provides immediate advice on various topics ranging from beauty contests to grooming routines and fashion hacks and tricks, making fashion advice interactive and readily available. Second, the Ramp Walk Tutor employs pose recognition technology to scan a user's gait and speed and turns and provides immediate feedback and tailored improvement recommendations. Third, the platform has a Trend Demand Forecasting module that predicts future fashion trends through machine learning algorithms based on historical and real-time data from various sources.

Besides all these features, MY Academy provides in-depth information about future beauty contests, contact numbers of designers, as well as fashion events, thus making a centralized platform for career development and fashion education. This project promises to fill the gap between conventional education for fashion and new-generation technology by providing a rich, insightful, and interactive experience of learning.

1.1 Problem Definition

The fashion industry, even if fast-evolving and colorful, has few easily available and tech-enabled platforms facilitating systematic instruction for those who wish to pursue or advance a career in the profession. Traditional modes of fashion instruction tend to be limited to classroom sessions or VIP workshops, which can be costly and geographically limited. Additionally, feedback on skills such as ramp walking or even personalized grooming advice proves hard to obtain in real time. Moreover, manual forecasting of fashion trends also proves time-consuming and incorrect on account of the ever-evolving nature of the industry. As a consequence, numerous would-be models and fashion connoisseurs fail to obtain a single all-encompassing platform offering instruction, training, industry information, and information about the latest developments on beauty pageants and trends.

1.2 Project Overview

My Academy is a web-based platform aimed at improving upon the means available in today's existing education system for fashion through the application of artificial intelligence, machine learning, and computer vision. My Academy brings together three key elements:

- AI Chatbot: A chat bot offering customers insight on beauty contests, style tricks, hair and make-up advice, as well as other style-related issues. The facility guarantees twenty-four-hour-a-day coverage and keeps customers educated and on their toes.

- Ramp walk tutor: Using pose detection technology, this module observes the user's walking pattern, speed andposes. It provides live feedback to assist users perfect their ramp walk, and is especially beneficial for thosegetting ready for modelling auditions or a fashion show.

- Trend Demand Forecasting: Utilizing machine learning algorithms, this feature examines historical and current fashion data to forecast upcoming fashion trends. It assists users, designers, and fashion marketers in staying ahead of changing styles.

Additionally, the platform compiles information on forthcoming beauty pageant events, designer contact details, and other vital resources. By integrating education, AI-driven feedback, and real-time insights, MY Academy offers a centralized and accessible solution for those seeking to build a career or remain informed in the fashion industry.

1.3 Hardware Specification

The MY Academy platform incorporates cutting-edge AI algorithms, real-time pose detection, and interactive web elements, necessitating specific hardware capabilities for optimal development, deployment, and user experience. During development, a multi-core processor like the Intel Core i5 or AMD Ryzen 5 is crucial for managing tasks such as AI model training, real-time data processing, and running multiple applications at once. For server deployment, a more robust multi-core CPU, such as the Intel Xeon, is needed to efficiently handle numerous user requests and AI computations. Sufficient memory is vital for high-speed data processing and multitasking. While 8 GB RAM is adequate for basic development, 16 GB or more is advised for training machine learning models and for server environments to effectively manage concurrent sessions and real-time analytics. Solid State Drives (SSD) are favored for their fast read/write speeds, which minimize latency when loading datasets, models, and application files. Development systems should have at least a 256 GB SSD, while server storage should be expanded to 500 GB to accommodate larger datasets and application storage needs. A GPU like the NVIDIA GTX 1050 or higher speeds up the training and inference stages of AI models, especially for computer vision tasks such as pose detection in the Ramp Walk Tutor. GPU acceleration significantly reduces processing time compared to CPU-only operations. The platform development is compatible with Windows, macOS, and Linux operating systems. Ubuntu Linux is often preferred for AI and machine learning development due to its superior support and compatibility with many AI frameworks. Camera: An HD webcam with a minimum resolution of 720p is necessary for accurately capturing user movements in the Ramp Walk Tutor module, enabling precise pose detection and feedback. A high-speed internet connection is essential both on the server side (100 Mbps or higher) to manage multiple concurrent users and data streaming, and on the user side (minimum 10 Mbps) to ensure smooth real-time interactions without lag or interruptions. End users can access MY Academy via desktops, laptops, or smartphones equipped with a webcam, a dual-core CPU, and at least 4 GB RAM, ensuring the platform remains accessible and responsive across common consumer devices.

1.4 Software Specification

The MY Academy platform's creation is backed by a suite of robust and adaptable software tools designed to deliver high performance, scalability, and user-friendliness. Python is the main programming language employed, thanks to its broad support for libraries in artificial intelligence, machine learning, and computer vision. The platform's entire interface is constructed using Streamlit, an open-source web framework based on Python, which facilitates smooth webpage integration and real-time interaction with AI models through a straightforward and intuitive interface. For the Ramp Walk Tutor module's pose detection, MediaPipe and OpenCV are used to effectively capture and analyze users' body movements. Machine learning elements, such as the Trend Demand Forecasting system, are implemented with libraries like Scikit-learn, XGBoost, Pandas, and NumPy. If needed, more sophisticated models can be developed using TensorFlow or PyTorch. The platform's chatbot feature is developed using either Dialogflow or Python-based NLP libraries, offering interactive responses related to beauty pageants, fashion advice, and grooming tips. For data storage and retrieval, such as user input, event listings, and forecast results, Firebase, MySQL, or MongoDB can be utilized depending on the deployment configuration. Version control is handled through Git and GitHub, enabling collaborative development and continuous updates. The system is compatible with major web browsers like Google Chrome, Mozilla Firefox, and Microsoft Edge, ensuring broad accessibility. Code development and testing are conducted using Visual Studio Code, PyCharm, and Jupyter Notebook.Finally, the platform is deployed using Streamlit Cloud, Heroku, or Firebase Hosting, providing a stable and scalable online presence for users globally.

1.5 System Specification

Hardware Requirements

Component	Specification
Processor	Intel Core i5 / AMD Ryzen 5 (Quad-core) for development; Intel Xeon or equivalent for deployment
RAM	Minimum 8 GB (Development); 16 GB or more (for AI model training and server deployment)
Storage	256 GB SSD (Development); 500 GB SSD or more (Server)
Graphics Card	NVIDIA GTX 1050 or higher (for pose detection and AI processing)
Operating System	Windows 10/11, Ubuntu Linux (preferred for AI/ML environment)
Camera	HD Webcam (720p or higher) for pose detection in Ramp Walk Tutor
Network	100 Mbps or higher (Server); Minimum 10 Mbps (User)
User Devices	Desktop/Laptop/Smartphone with webcam, dual-core CPU, and minimum 4 GB RAM

Software Requirements

Component	**Specification / Tools Used**
Operating System	Windows 10/11, Ubuntu Linux (for development and deployment)
Programming Languages	Python, JavaScript, HTML5, CSS3
Web Framework	Streamlit (for webpage integration and UI development)
Pose Detection Library	MediaPipe, OpenCV
Machine Learning	Scikit-learn, XGBoost, Pandas, NumPy, TensorFlow / PyTorch (optional for deep learning)
Chatbot Framework	Python NLP libraries / Dialogflow
Database	Firebase / MySQL / MongoDB
Web Hosting	Streamlit Cloud / Heroku / Firebase Hosting / Render
Version Control	Git, GitHub
Browser Compatibility	Google Chrome, Mozilla Firefox, Microsoft Edge
IDE / Code Editor	Visual Studio Code (VS Code), Jupyter Notebook, PyCharm

Chapter 2 Literature Review

2.1 Existing System

The fashion industry has experienced a notable increase in the use of Artificial Intelligence (AI), especially in areas like automated customer support, trend analysis, and virtual fitting experiences. Despite this, many current solutions function independently. For example, trend forecasting tools such as Google Trends or Edited rely on past market data but do not incorporate localized, real-time consumer preferences or AI-based prediction methods. Fashion chatbots like Levi's Virtual Stylist or H&M's Kik bot provide fashion advice but are confined to recommending products within their own platforms. Likewise, pose detection technology has been utilized in fitness or augmented reality filters (e.g., Snapchat, TikTok) but has not been adapted for structured fashion training, such as improving runway walking posture. Furthermore, most existing solutions focus on commercial uses, overlooking the educational and informational needs of individuals aspiring to enter the fashion field, such as models, fashion students, or event organizers. This fragmented system lacks a unified platform that combines training (like a runway walk instructor) and informational services (such as updates on pageants or fashion advice), particularly with an AI-powered backend.

2.2 Proposed System

MY Academy is an innovative web-based platform designed using Streamlit that bridges the gap between advanced technologies and the dynamic needs of the fashion industry. It seamlessly integrates multiple AI-driven modules to deliver fashion training, real-time updates, and predictive insights—all in one cohesive environment.

Core Modules:

- **AI Chatbot**: Powered by Python NLP frameworks and optionally integrated with tools like Dialogflow, the chatbot facilitates real-time, domain-specific interactions. It provides guidance on beauty pageants, fashion styling, dressing hacks, grooming techniques, and event preparation—trained on curated datasets and FAQs for expert-level responsiveness.

- **Ramp Walk Tutor**: Utilizing MediaPipe and OpenCV for pose estimation, this module analyzes users' posture, stride, and turning techniques. It offers real-time visual feedback to refine walking styles through structured virtual training. Optional support for TensorFlow Lite enables efficient edge-device implementation.

- **Trend Demand Forecasting**: This module applies machine learning models such as XGBoost and ARIMA to analyze historical fashion data, seasonal cycles, and social media trends. It provides multi-feature trend predictions—including seasonality, fabric types, and popular colors—empowering designers and boutiques with forward-looking insights.

- **Event & Pageant Tracker**: Aggregates fashion events and pageant opportunities using web scraping APIs or platforms like Eventbrite. It offers timely information on upcoming events, registrations, and deadlines to keep users engaged with the fashion ecosystem.

By consolidating training tools, trend intelligence, and event discovery into a single platform, MY Academy supports fashion aspirants and professionals alike. It empowers users with practical skills, data-driven insights, and timely opportunities—fostering growth and innovation across the fashion landscape.

2.3 Feasibility Study

To ensure the project's practical implementation and long-term success, a comprehensive feasibility analysis was conducted across three key dimensions:

2.3.1 Technical Feasibility

The selected technology stack is both robust and scalable, supporting real-time applications in the fashion-tech domain:

- **Programming Environment** : Python serves as the backbone, offering a wide array of AI/ML libraries such as Scikit-learn, XGBoost, Pandas, NumPy, and TensorFlow for efficient model development.

- **Pose Estimation** : MediaPipe ensures lightweight, real-time posture analysis suitable for both web and mobile platforms.

- **Deployment**: Streamlit enables rapid development and deployment of interactive, data-driven applications with minimal overhead.

- **Modularity** : The architecture is modular, allowing for future expansions such as integrating speech-to-text interfaces or AR-enhanced feedback.

- **Device Compatibility** : Optimized for mid-range hardware and can be containerized using Docker for streamlined cloud deployment.

- **Security & Scalability** : Secure authentication via Firebase is feasible, with backend scalability options including microservices or AWS Lambda.

- **Conclusion** : The system leverages widely adopted, open-source technologies aligned with current industry standards, ensuring high compatibility and maintainability.

2.3.2 Economic Feasibility

The project incurs minimal upfront costs due to strategic use of free and open-source tools:

- **Frameworks & Libraries** : Core components like Streamlit, MediaPipe, and Scikit-learn are open-source and cost-free.

- **Development Infrastructure** : No need for paid servers during development; platforms like Streamlit Cloud and Render offer free-tier hosting options.

- **Hardware Requirements** : A standard laptop with a webcam suffices for development and testing.

- **Scalable Investment** : Cloud resources and storage can grow based on user adoption, allowing for incremental cost scaling.

- **Revenue Potential** : Monetization avenues include premium training subscriptions, in-app advertising, affiliate marketing with fashion brands, and consulting services.
- **Conclusion** : The solution is economically sustainable, with a low barrier to entry and flexible growth potential.

2.3.3 Operational Feasibility

The platform is designed with user accessibility and real-world applicability in mind:

- **User Interface**: Built on Streamlit's intuitive components with chatbot integration, ensuring ease of use for all demographics.
- **Real-Time Interactivity**: Instant feedback from pose detection and conversational modules enhances learning and engagement.
- **Cross-Device Accessibility**: Operates on both desktops and mobile browsers, requiring no software installation.
- **Stakeholder Utility**: Models benefit multiple user groups—enhancing ramp walk skills, supporting trend-informed design, and enabling event promotions.
- **Conclusion**: The system is highly usable, requires minimal operational oversight, and promises a high satisfaction rate among its diverse users.

Chapter 3 System Analysis and Design

3.1 Introduction to Design

System analysis and design are crucial stages in the creation of any software application because they determine how the system operates, how the users will engage with it, and through which components data streams. This chapter presents an extensive analysis and design for the suggested system, which includes functionalities like demand forecasting, educational content, event news, and chatbot communication.

To model the internal structure and external interactions of the system, modeling tools like Use Case Diagram, Data Flow Diagram (DFD), System Flowchart, and Entity-Relationship (ER) Diagram are employed. These modeling tools assist in visualizing system behavior, determining user roles, charting processes, and defining the relationships between various data entities.

Through the explicit visualization of these design elements, the chapter guarantees the system is logically organized, user-friendly, scalable, and maintainable.

3.2 UML Diagrams

Unified Modeling Language (UML) is an established and deeply rooted practice that is used to visually represent the components of the design and the behavioral interactions of a system. It is an extremely powerful tool that is used as an invaluable asset for developers, designers, and stakeholders alike by giving them an easier way to view how the various components and elements of a system work together and function as a whole unit. By using UML diagrams, it is immensely easier to correctly plan, organize, and communicate the complex architecture of the system as well as the various user interactions that will occur, all prior to the actual development process being initiated. Of the various UML diagrams that are used, one of the most commonly used is the use case diagram, which is most often used in the illustration of the users' interactions and the system.

3.3 Use Case Diagram

The use case diagram is an external representation that emphasizes the different ways and situations in which the key users interact with the system designed by My Academy. From studying this diagram, it is quite effortless to understand the precise activities each user group can carry out and the precise function each of them serves in the overall framework of the system.

There are two general categories of users:

- **Visitor/User:** Refers to any regular user visiting the website. Visitors are able to view and use different features like requesting forecasts, viewing forecast results, viewing future events, reading blog posts, chatting with the chatbot, and viewing video tutorials. The admin will see to it that the platform's features and content are updated. This will include updating forecast forms, updating chatbot features, writing blogs, event management, and video tutorial content management.

Use Cases for Visitor/User:

- Request a forecast through the forecast form.
- View the result of prediction outcomes.
- Get details about upcoming fashion events.
- Read informative blog posts related to fashion trends and forecasting.
- Interact with the AI-powered chatbot for personalized fashion advice.
- Watch and view a range of video tutorials that discuss a broad range of fashion topics.

3.3.1 Use Cases for Admin:

- Review and update forecast forms to ensure accuracy.
- Enhance and enrich chatbot responses and overall functionalities and capabilities.
- Add, edit, or remove blog posts to keep the site up to date.
- Systematically organize and update information related to upcoming events for the users.
- Post video tutorials designed to educate users and keep them continuously updated.

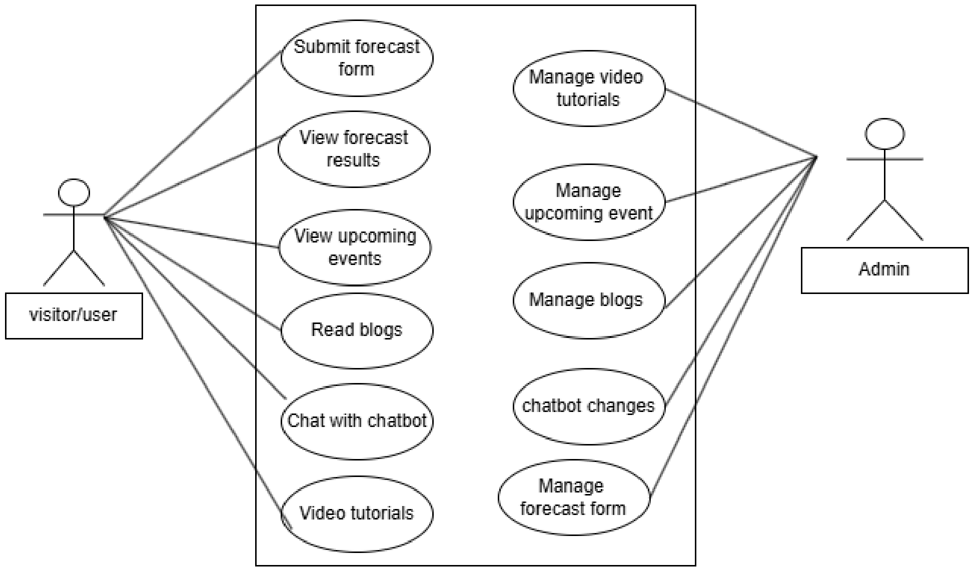


Figure 3.1: Use Case Diagram

The following use case diagram is employed to provide a general graphical representation of these interactions that are the core of the system.

3.4 Data Flow Diagrams (DFD)

Data Flow Diagram (DFD) is among the graphical methods used to display how data moves in a system. It highlights the origin points of the data, the processes that change it, and the destination points where it is delivered. DFDs are beneficial in system behavior and are widely used during software development planning and designing phases.

The DFD shows how administrators and users interact with the site and how information flows through different components of the system.

DFDs are organized hierarchically:

- **Context-level DFD:** Provides a high-level overview of the entire system and its interaction with external actors.
- **Level-1 DFDs:** Break down the context diagram into more detailed processes showing specific data flows and storage.

3.4.1 Level 0

The **Level 0 Data Flow Diagram (Context Diagram)** gives a general overview of the entire system as a single process. It focuses on the external users and how the system interacts with them.

External Entities:

Visitor/User: This is the average user of the site. They are able to :

- Submit forecast forms
- Pose questions to the chatbot
- Watch videos and read blogs
- Preview upcoming events
- Get forecast outputs and chatbot responses

Admin: The admin manages the features and content of the system. They can:

- Edit blog posts
- Rephrase chatbot answers
- Manage video tutorials and event details
- Manage forecast form settings

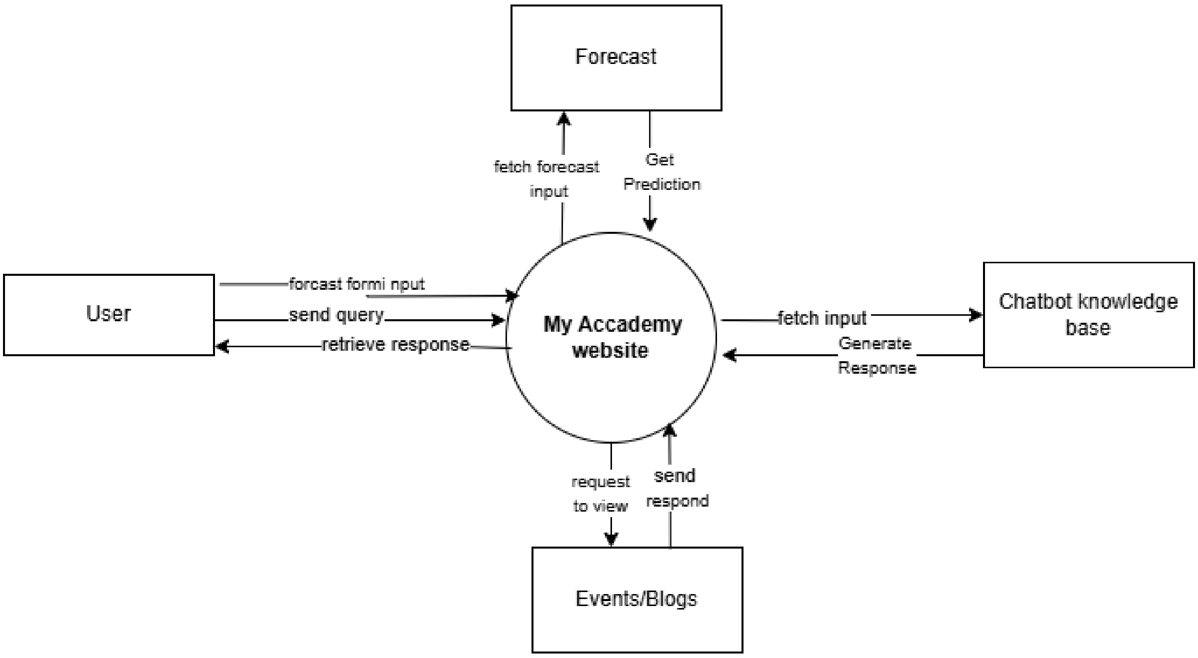


Figure 3.2: Context-Level Data Flow Diagram

Central System

My Academy Platform is an intermediary process. It receives data from admins and users, processes them, and provides the needed outputs like forecast outcomes, chatbot replies, and content display.

This Level 0 diagram simplifies the entire system down to a single overall function and depicts how data is transferred to and from external entities.

3.4.2 Level 1

DFD Level 1 illustrates a comprehensive picture of the My Academy Website's internal processes that deal with external entities and data stores. It decomposes the top-level system process (from DFD Level 0) into subprocesses to show how user inputs are handled, data is retrieved, and outputs are provided. The system accommodates different features including:

- Forecast form submission and display of results
- Displaying events and blog posts
- Conversing with a chatbot for information
- Watching video tutorials

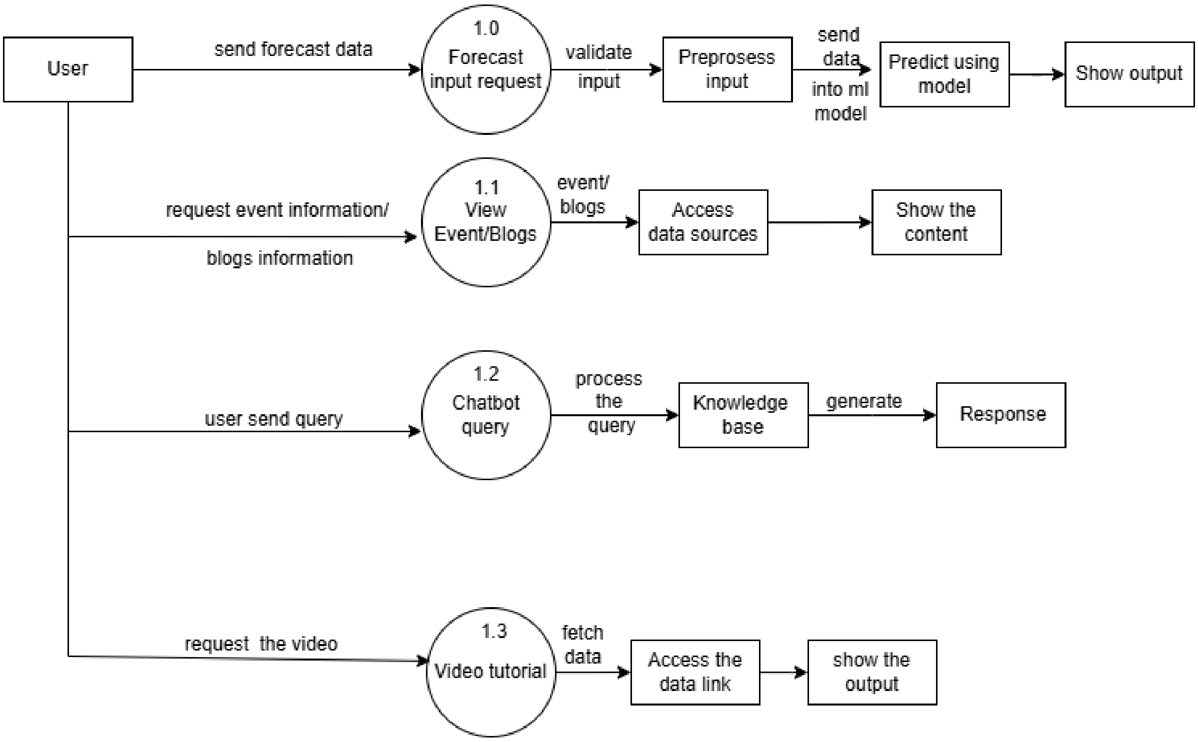


Figure 3.3: Level 1

This level identifies four major processes, their interaction with external parties (mainly users), and respective data stores.

3.5 Entity-Relationship (ER) Diagram

The ER diagram illustrates the database schema of My Academy, including the principal entities, attributes, and how they are interconnected. It is a visual illustration of how the system is utilized by the users and how the data is organized.

3.5.1 Key Entities and Attributes:

- **User**

  Attributes: *User_id*, *User_name*, *email*

  Represents any user of the site. Users can make use of various services like prediction-making, chatting with the chatbot, reading blogs/news, and watching video tutorials.

- **Forecast Form**

  Attributes: *Date*, *Hour*, *Sku value*, *Temperature*

Associated with the user through the "Generate forecast" relationship. Used by users to generate demand forecasts.

- **Blogs/Upcoming Events**

  Attributes: *Blog_id*, *Title*, *Date*, *Author*

  Users can view and examine blogs or event details through the "Read" relationship.

- **Video Tutorial**

  Attributes: *Video_id*, *Title*

  The audience can watch learning or instructional videos uploaded by the admin through the "Watch" relationship.

- **Chatbot**

  Attributes: *User_id*, *Message*, *Response*

  The chatbot system communicates with users, answering their questions. This is indicated by the "Interacts with" relationship.

3.5.2 Relationships

- **Create forecast:** Connects the User to the Forecast Form. Users input data to get predictions.

- **Read:** Connects User to Blogs/Upcoming Events. This allows users to view informative material and event listings.

- **Watch:** Binds User to Video Tutorials. Users can watch video-based content to learn or be taught.

- **Interacts with:** Connects the User with the Chatbot. The User can send questions, and the chatbot will reply accordingly.

This ER model ensures efficient interaction between users and different components of the My Academy system, supporting its core functionalities like demand forecasting, content delivery, and user engagement.

This ER model supports the key functionalities of my project, such as efficient content management, user interaction, and personalized forecasting.

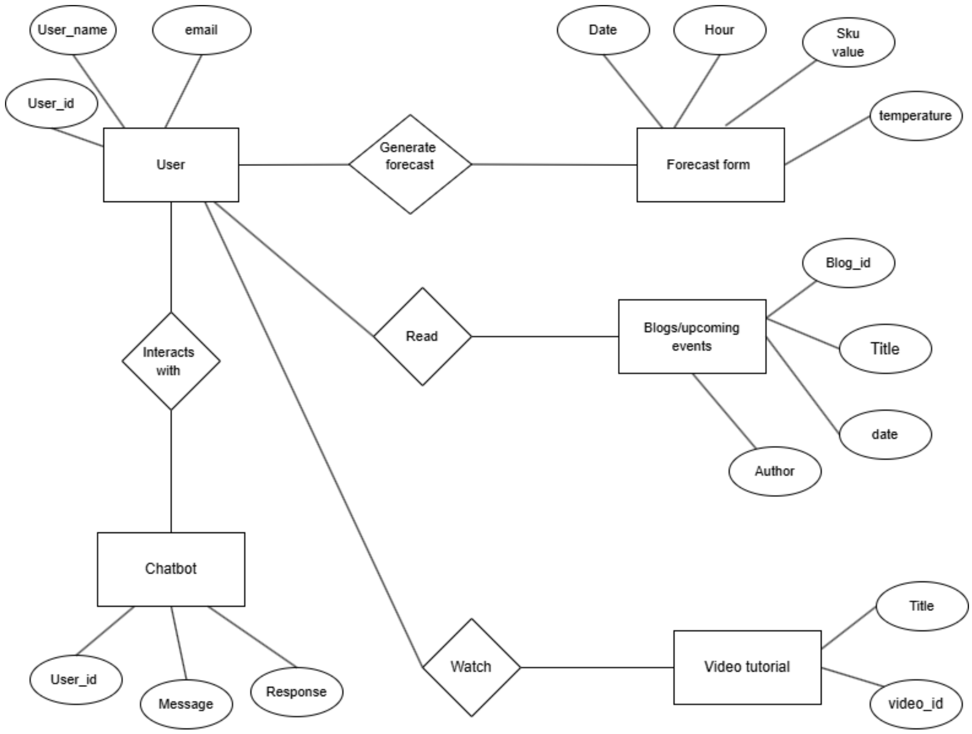


Figure 3.4: ER Diagram

3.6 Flowchart

A flowchart is a graphical illustration of the sequential process of operations of the My Academy platform. It illustrates how a user engages in various features, from visiting the website to performing activities like generating forecasts, browsing blogs or events, viewing tutorials, and talking with the chatbot. Flowcharts make it easy to comprehend the user flow and logic for every process.

3.6.1 Flowchart Description

The workflow starts with the opening of the My Academy site by the user, followed by either logging in or registering on the site. Once verified, the user is given several options depending on their needs and interests. The steps below illustrate the interaction:

- **Start:** The session is started by the user accessing the site.
- **Open Website:** The site's homepage is opened where the user can log in or register.
- **Login/Register:** User enters login credentials or registers a new account.

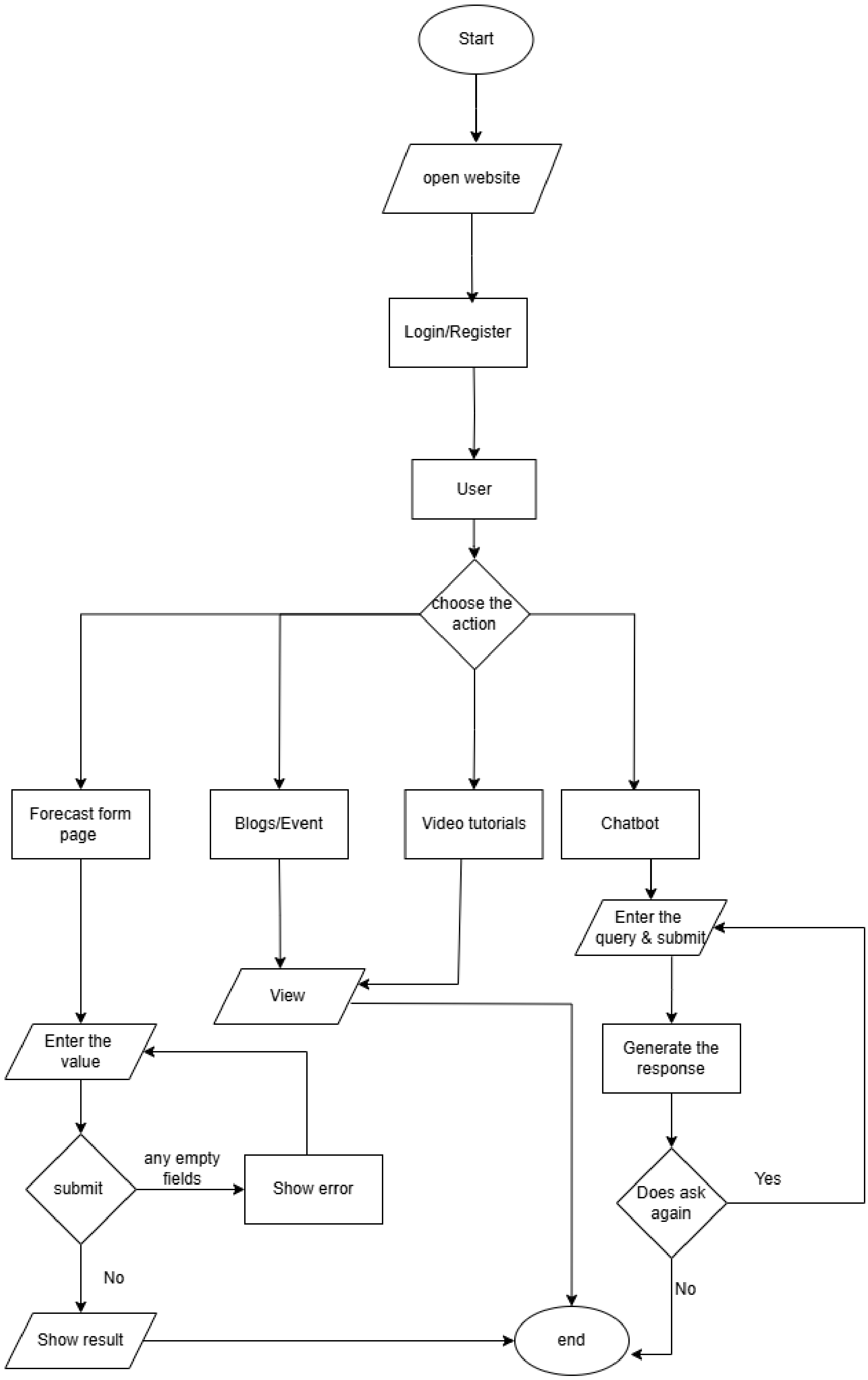


Figure 3.5: Flowchart of the My Academy Platform

- **User Dashboard:** Upon successful login, the user is presented with a dashboard consisting of four prominent options.
- **Forecast Form Page:** User requests a forecast submission. Inputs required values like date, hour, SKU, and temperature. If any field is left blank, an error message is displayed. If all fields are entered, the form is submitted, and the system generates and displays the forecast result.
- **Blogs/Events:** User can see the newest blogs or upcoming event announcements, keeping them up to date on fashion information.
- **Video Tutorials:** Educational videos illustrating platform features and tools are accessible to the user.
- **Chatbot:** User inputs a query and sends it to the chatbot. A response is presented based on the query. If the user wishes to proceed, they can input another query. Otherwise, the process ends.
- End: The user finishes their tasks and leaves the system.

Chapter 4 Project Module

4.1 Concepts and Technologies

4.1.1 Concept

The MY Academy project centers on developing a unified, AI-driven digital platform tailored to the needs of fashion enthusiasts, aspiring models, and designers. It aims to offer intelligent guidance, practical skill development, and trend forecasting in one accessible environment. The platform brings together several modules, including:

- An AI-powered chatbot that delivers real-time fashion advice.
- A ramp walk tutor leveraging pose detection for posture correction and walking guidance.
- A trend prediction engine utilizing machine learning models to forecast fashion trends.

This all-in-one solution streamlines the preparation process for beauty pageants and fashion events while providing users with valuable insights into upcoming industry trends and opportunities.

4.1.2 Technologies Used

The platform is developed using a combination of modern programming tools, AI libraries, and data visualization technologies:

- Python: The core language for backend development, AI modeling, and system orchestration.

- Streamlit: Enables the creation of responsive, interactive web interfaces with real-time data integration.
- MediaPipe: Provides efficient, real-time pose detection capabilities for the ramp walk tutor.
- OpenCV: Supports webcam input and image processing tasks for body movement analysis.
- NLTK / Transformers: Used in the chatbot module to understand natural language queries and provide intelligent responses.
- Scikit-learn & XGBoost: Power the trend forecasting module through advanced machine learning algorithms.
- Pandas & NumPy: Essential for data cleaning, transformation, and numerical computations.
- BeautifulSoup / Scrapy: Facilitate web scraping for collecting real-time information on fashion events and pageants.
- Matplotlib & Seaborn: Used for creating insightful visualizations to communicate data analytics and forecasting results.
- SQLite: A lightweight database to store structured information such as FAQs, user interactions, and prediction logs.

4.2 Testing

4.2.1 Introduction to Testing

Testing plays a vital role in the software development life cycle by ensuring the reliability, accuracy, and performance of the system. For the MY Academy platform, rigorous testing was conducted at both module and system levels to confirm that each component functioned as intended and contributed to a smooth, cohesive user experience.

4.2.2 Testing Methodologies Used

To comprehensively validate the MY Academy system, multiple testing methodologies were implemented:

- Unit Testing: Each core module—Chatbot, Ramp Walk Tutor, and Trend Forecasting—was tested independently to verify that individual functions and logic blocks executed correctly in isolation. Mock data inputs were used to test edge cases and ensure robustness.
- Integration Testing: After successful unit testing, the modules were integrated within the Streamlit interface. Tests were performed to ensure data exchange and interaction between modules occurred without errors, validating workflow continuity and inter-module communication.
- System Testing: The complete platform was tested as a unified system. This phase evaluated overall functionality, performance, UI responsiveness, and data accuracy, simulating real-world use cases and scenarios to identify potential issues.
- User Acceptance Testing (UAT): A sample group of target users—including fashion students, aspiring models, and enthusiasts—interacted with the platform to assess its real-time usability, the relevance and accuracy of trend forecasts, and the responsiveness of the chatbot. Their feedback informed final refinements to the interface and functionality.

4.2.3 Bug Tracking and Improvements

Throughout the initial testing phases, several minor issues were identified and addressed to enhance system stability and user experience:

- Delayed Chatbot Responses: Caused by improper intent classification within the NLP model. Solution: The NLP pipeline was fine-tuned to improve context understanding and reduce response latency.
- Overly Sensitive Pose Feedback: The Ramp Walk Tutor provided overly strict feedback, making it difficult for users to meet posture benchmarks. Solution: Pose

detection thresholds were recalibrated to offer a more user-friendly experience while maintaining accuracy.

- Graph Label Overlap on Smaller Screens: Trend forecasting graphs had overlapping axis labels when viewed on smaller displays.
- Solution: The UI was optimized for responsiveness within the Streamlit framework, ensuring better visualization across devices.

These iterative improvements contributed to a more refined, user-centric system, aligning functionality with real-world usability expectations.

4.3 Software Development Life Cycle (SDLC)

The Software Development Life Cycle (SDLC) is a systematic framework used to guide the development of high-quality software applications. It ensures that each phase of development—from planning to deployment—is conducted in a structured, organized manner. For the development of MY Academy, an AI-driven platform designed for the fashion industry, the Waterfall model of SDLC was adopted. This linear and sequential approach was chosen for its simplicity and clear structure, making it especially suitable for academic projects.

4.3.1 Planning

The planning phase began with clearly defining the project's objectives, scope, and key functionalities. This involved identifying the target users, understanding their needs, and outlining the modules required to address them. Major components such as the chatbot, ramp walk tutor, and trend forecasting engine were conceptualized during this phase. Timelines were estimated, responsibilities were assigned, and a feasibility study was conducted to assess the project's technical, economic, and operational viability. Additionally, the necessary resources—both hardware and software—were identified to ensure a smooth development process.

4.3.2 Requirement Analysis

This phase focused on gathering and analyzing all the requirements the system needed to fulfill. Functional requirements such as providing fashion guidance, delivering pose-based ramp walk feedback, and forecasting seasonal trends were identified through discussions with potential users and a review of similar platforms. Non-functional requirements, including system responsiveness, ease of use, reliability, and scalability, were also considered. The outcome of this phase was a detailed Software Requirements Specification (SRS) document that served as a blueprint for the design and development phases that followed.

4.3.3 System Design

In the design phase, the architecture of MY Academy was carefully planned. The platform was divided into four main modules: the chatbot, ramp walk tutor, trend forecasting engine, and event listing page. User interface layouts were sketched using wireframes, while diagrams such as data flow diagrams were used to map how data would move through the system. Technologies like Python, Streamlit, MediaPipe, and XGBoost were chosen for their suitability in handling natural language processing, real-time pose detection, and machine learning. The system was designed with modularity in mind to allow for future updates and scalability.

4.3.4 Implementation

Once the design was finalized, the actual development work began. Each module was implemented as an independent component to allow for easier testing and debugging. The chatbot was developed using Python with a rule-based approach to handle fashion-related queries. The ramp walk tutor leveraged MediaPipe's pose estimation capabilities to monitor body posture and provide feedback. The trend forecasting engine used the XGBoost algorithm, trained on seasonal fashion datasets, to generate predictions. All these components were integrated into a single platform using Streamlit, providing users with a seamless and interactive experience.

4.3.5 Testing

Thorough testing was conducted to ensure that the system functioned correctly and delivered a smooth user experience. Unit testing was performed on each module to check for individual correctness, while integration testing ensured that all modules worked together without issues. System testing was then carried out to evaluate the platform under realistic conditions. Additionally, User Acceptance Testing (UAT) was conducted with fashion students and mentors, who provided valuable feedback on the platform's usability and performance. Any bugs or issues identified were addressed promptly, resulting in a more refined and reliable product.

4.3.6 Deployment

After successful testing, the system was deployed locally using Streamlit's built-in tools. The application was tested across different devices and web browsers to confirm compatibility and responsiveness. Users were able to interact with the chatbot, use the ramp walk tutor, and explore trend forecasts in real time. While the current deployment is local, plans are in place to move the platform to the cloud using services such as Heroku or AWS. This would allow for broader access, better performance, and easier maintenance in the future.

4.3.7 Maintenance

Although MY Academy is currently deployed as a prototype, it has been designed with long-term maintainability in mind. Future improvements include enhancing the chatbot with machine learning for better intent recognition, automating event data updates through real-time web scraping, and upgrading the trend forecasting model using social media trend analysis. The system's modular structure allows for easy updates and the addition of new features without disrupting existing functionality. Open-source tools and version control systems are also used to simplify the update and debugging process.

4.3.8 Waterfall Model

The development of the MY Academy website followed the Waterfall Model, a structured and linear approach that ensures each phase is completed before moving to the next. This methodology suited our project well, allowing us to carefully design and implement each module—namely, the Fashion Chatbot, Ramp Walk Tutor using Pose Detection, and Fashion Trend Forecasting Engine—with clarity and control.

- Requirement Analysis: In this phase, we gathered all necessary information regarding user expectations and functional goals of the project. The main aim was to create a platform that provides fashion guidance, event updates, and interactive features for users interested in the fashion and beauty pageant industry. The chatbot, ramp tutor, and trend forecasting features were identified as the core modules during this stage.

- System Planning: After understanding the requirements, we prepared a detailed plan and system design. Architecture diagrams, module flows, and UI sketches were developed. Technologies like Python, MediaPipe, and Streamlit were finalized for use. Data flow and module interaction were defined for smooth development and integration of all features.

- Module Development: Each module was implemented individually, beginning with the chatbot that answers fashion-related queries. The ramp walk tutor was created using pose detection with MediaPipe, and the trend forecasting module was developed using machine learning models like XGBoost. All components were integrated into a cohesive interface using Streamlit for seamless navigation and user interaction.

- System Testing: Testing involved checking the chatbot's accuracy, ensuring correct pose detection in the tutor, and validating the forecasting model outputs. Integration testing confirmed that all modules worked together without conflict. We also collected user feedback and improved the interface for better usability.

- Deployment: The completed system was deployed locally using Streamlit. The website interface allowed users to interact with all features directly—chatting with the fashion bot, practicing ramp walking through webcam-based feedback, viewing trend insights, and accessing pageant updates—all in a single platform.

- Maintenance Scope: Though the current deployment is a working prototype, the system is designed for expansion. The chatbot can be improved with advanced

NLP, forecasting can be updated with real-time data, and the UI can be enhanced for mobile responsiveness. Future versions may include cloud hosting and user login features.

chapter 5: Implementation

The implementation phase signifies the transition from design to functional development, where conceptual plans are translated into operational modules. For the MY Academy platform, this stage focused on building three primary components—Fashion Chatbot, Ramp Walk Tutor utilizing Pose Detection, and Trend Demand Forecasting—and integrating them into a unified web interface using Streamlit. Each module was developed with suitable technologies, emphasizing modular architecture, scalability, and efficient performance.

5.1 Chatbot Implementation

The My Academy website includes a chatbot that is an intelligent virtual assistant that answers users in real-time. It navigates users through the site, answers queries related to fashion forecasting demand, and offers dynamic content such as blogs, events, and forecasting results.

5.1.1 Technology Used: Botpress Framework

The chatbot was developed on the Botpress platform, an open-source conversational AI solution with a well-known modular architecture and easy-to-use interface. Botpress provides robust natural language understanding (NLU) tools, flow-based conversation design, and seamless integration with external databases and API calls. Its knowledge base module, which can be added as an inbuilt module, allows the chatbot to save and retrieve question-answer pairs for common user questions quickly. JSON (JavaScript Object Notation) is extensively employed in Botpress to structure conversational data like dialogue flows, user intents, entity definitions, and response templates. The JSON configuration facilitates easy updating and maintenance of chatbot content and ensures easy API communication for enabling dynamic as well as personalized user interactions.

5.1.2 Knowledge Base Integration

A specialized knowledge base was set up to store repeatedly asked questions and their respective answers. For example, when users ask something like "What is fashion demand forecasting?" or "What events are forthcoming in the near future?", the chatbot searches in the knowledge base and provides accurate responses. This automation reduces the use of manual responses from site administrators.

5.1.3 Database Connectivity for Dynamic Responses

The bot is connected to the backend database of My Academy's platform via API endpoints. This enables the bot to fetch dynamic information such as the latest blog posts, upcoming event details, and user-defined forecast results based on user input so that it can have more contextual and timely interactions.

5.1.4 User Interaction Flow and Intent Recognition

Botpress's NLU feature allows the chatbot to identify different user intent, such as queries pertaining to forecasts, blogs, and events. For each identified intent, there is a corresponding conversation flow that is initiated, leading users step-by-step through the related information, thereby providing seamless and user-friendly interactions.

5.1.5 Embedding on Website

The chatbot is implemented as a floating widget on all pages of the website. The user can easily collapse or expand it and it is themed to match the theme of the entire site, hence ensuring consistency of branding. This model offers immediate assistance to users without having to abandon the current page.

5.1.6 Benefits and Impact

It enhances user experience through real-time support and auto-answer to common questions. It minimizes administrator workload and retains users in active use with timely, pertinent information, enhancing the overall usability and satisfaction of the platform.

5.1.7 Challenges and Solutions

It was one of the biggest challenges to train the chatbot to properly understand diverse user questions and avoid misinterpretations. It was addressed by continuous testing, adding more varied user statements to the training data and fine-tuning the NLU model for better intent classification accuracy.

5.2 Ramp Walk Tutor with Pose Detection

The Ramp Walk Tutor serves as a key module within the MY Academy website, aimed at assisting users in mastering ramp walking techniques through real-time body pose analysis. This feature delivers immediate feedback on posture and movement by comparing the user's poses with predefined ramp walk standards. By simulating a virtual training environment, the module enables individuals to practice runway walking and receive corrective guidance dynamically.

5.2.1 Pose Detection Overview

Pose detection is a computer vision method used to identify and track key landmarks on the human body within images or video streams. It focuses on estimating the spatial positions and orientations of joints such as the head, shoulders, elbows, hips, knees, and ankles. These landmarks are then connected to create a skeletal representation that reflects the body's posture. This technology plays a significant role across various domains, including fitness monitoring, motion analysis, animation, healthcare, gaming, and interactive applications like virtual fashion guidance.

5.2.2 History and Evolution of Pose Detection Libraries

The evolution of pose detection technologies began with traditional computer vision methods that utilized handcrafted features and template matching techniques. These early approaches, while foundational, were limited in accuracy and scalability. A significant advancement occurred with the introduction of deep learning-based models, which marked a new era in human pose estimation.

One of the pioneering libraries in this space was OpenPose, developed by Carnegie Mellon University in 2016. It introduced the concept of Part Affinity Fields, enabling accurate detection and association of body parts for multiple individuals within a single frame. This innovation set a new benchmark for multi-person pose estimation.

Subsequently, Google introduced PoseNet, a lightweight library optimized for real-time inference directly within web browsers through TensorFlow.js. PoseNet extended accessibility by enabling pose estimation on devices with limited computational resources.

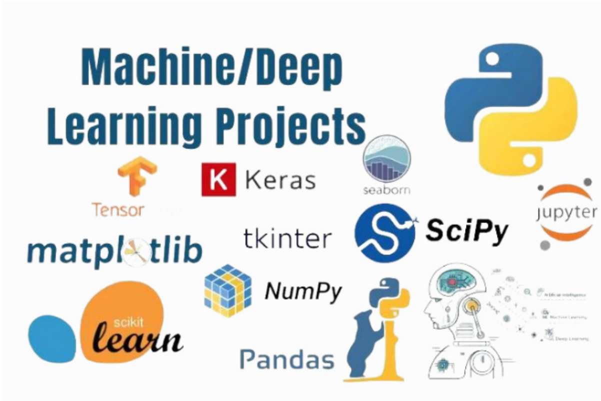
The latest advancement is MediaPipe Pose, also developed by Google. This library offers a highly efficient, cross-platform solution capable of detecting 33 body landmarks with high precision. Designed for real-time applications, MediaPipe Pose supports deployment across web, desktop, and mobile environments, making it well-suited for interactive systems requiring robust performance.

5.2.3 Types of Pose Detection Libraries

A variety of pose detection libraries are available, each offering unique capabilities tailored to different use cases:

- OpenPose is renowned for its ability to detect multiple individuals simultaneously with high accuracy.
- PoseNet offers a lightweight architecture suitable for deployment in web browsers and mobile applications.
- MediaPipe Pose provides real-time performance across platforms and supports both 2D and 3D landmark detection.
- AlphaPose specializes in high-precision, single-person pose estimation.
- HRNet (High-Resolution Network) preserves high-resolution features throughout the network to achieve superior pose estimation accuracy.

These libraries are compatible with multiple programming languages and platforms. However, Python remains the most widely adopted language, owing to its simplicity and the extensive ecosystem it offers for artificial intelligence and computer vision development.



5.2.4 Library Used in This Project

The MY Academy project utilizes the MediaPipe Pose library in conjunction with the Python programming language. MediaPipe was selected due to its high performance, accuracy, and straightforward integration with Python-based systems. It is capable of detecting 33 keypoints of the human body in real time and operates efficiently with live webcam input—an essential requirement for the Ramp Walk Tutor module.

5.2.5 Platform and Environment

The pose detection model is deployed on Google Colab, a cloud-based Jupyter Notebook environment that supports Python and offers free access to powerful computational resources. In this project, the model operates with GPU acceleration provided by the NVIDIA Tesla T4, significantly boosting the performance and responsiveness of the MediaPipe Pose library. This setup ensures efficient real-time frame processing and enables smooth, continuous analysis of body posture during ramp walk practice sessions.

- **NVIDIA Tesla T4 GPU :**

  The NVIDIA Tesla T4 is a high-performance GPU designed for AI inference, machine learning, deep learning, and cloud-based computing tasks. As part of NVIDIA's data center GPU lineup, it is widely adopted in cloud platforms such as Google Colab to accelerate computational workloads in real-time applications, including pose detection and video processing.

  Built on NVIDIA's Turing architecture, the Tesla T4 supports both FP32 (single-precision floating point) and reduced-precision formats such as INT8 and FP16.

This mixed-precision capability enhances the speed of AI inference while maintaining reliable accuracy.

Key features of the Tesla T4 include:

- 2,560 CUDA cores and 320 Tensor Cores, enabling high-throughput parallel processing for deep learning workloads.
- Optimization for real-time AI inference, making it well-suited for tasks such as pose estimation, image recognition, and video analytics.
- A low-power design (70W), contributing to energy-efficient performance in cloud environments.
- Compatibility with TensorRT, NVIDIA's deep learning inference optimizer, which further boosts model execution efficiency.

In this project, the Tesla T4 GPU available through Google Colab delivers the computational acceleration required for running the MediaPipe Pose model efficiently. It enables fast, frame-by-frame pose estimation with minimal latency, supporting a smooth and responsive experience during live ramp walk training. Leveraging GPU resources instead of relying solely on CPU processing allows the system to handle real-time video input and complex model computations more effectively.

5.2.6 Implementation in MY Academy

Within the MY Academy platform, the Ramp Walk Tutor module utilizes MediaPipe Pose to deliver real-time feedback to users. Video input from the webcam is processed frame by frame through the pose detection model, enabling continuous analysis of the user's skeletal posture. The system evaluates key elements such as body alignment, walking angles, and turning techniques, offering corrective guidance aligned with standard ramp walk practices.

This implementation transforms conventional ramp walk training into an accessible, AI-driven virtual experience. Developed using Streamlit and integrated with Python and MediaPipe on Google Colab, the module provides a responsive and interactive learning environment tailored for aspiring models and fashion enthusiasts.

5.3 Trend Forecasting Module

The Trend Forecasting Module within the MY Academy platform is developed to predict emerging fashion trends through data-driven methodologies and machine learning techniques. This module follows a systematic pipeline designed to ensure accuracy, contextual relevance, and actionable insights. The process encompasses key stages including data collection, preprocessing, feature selection, model training, and performance evaluation.

To provide a clearer understanding of this workflow, the following diagram outlines the core components involved in implementing the forecasting system.

5.3.1 System Workflow Diagram

- **Data Acquisition :**

  The forecasting module initiates with the collection of data from multiple sources, including historical retail sales records, curated fashion trend datasets, and contextual external factors such as holidays, festivals, and promotional events. These diverse inputs are consolidated into a unified data structure, establishing a comprehensive and reliable foundation for subsequent model training.

- **Data Preprocessing and Feature Engineering :**

  The preprocessing stage involves conducting exploratory data analysis (EDA) to address missing values, detect distributional skewness, and identify multicollinearity among features. Feature engineering is applied to construct informative variables such as seasonal indicators, discount periods, and promotional flags. Categorical data is transformed using One-Hot or Label Encoding, while numerical features are standardized to maintain a uniform scale, enhancing model accuracy and performance.

- **Model Training and Testing**

  The forecasting module involves training and evaluating multiple regression algorithms, including Linear Regression, Random Forest, and XGBoost. The dataset is partitioned into training and testing subsets, typically following an 80:20 ratio. Each model undergoes hyperparameter tuning to optimize performance. Among

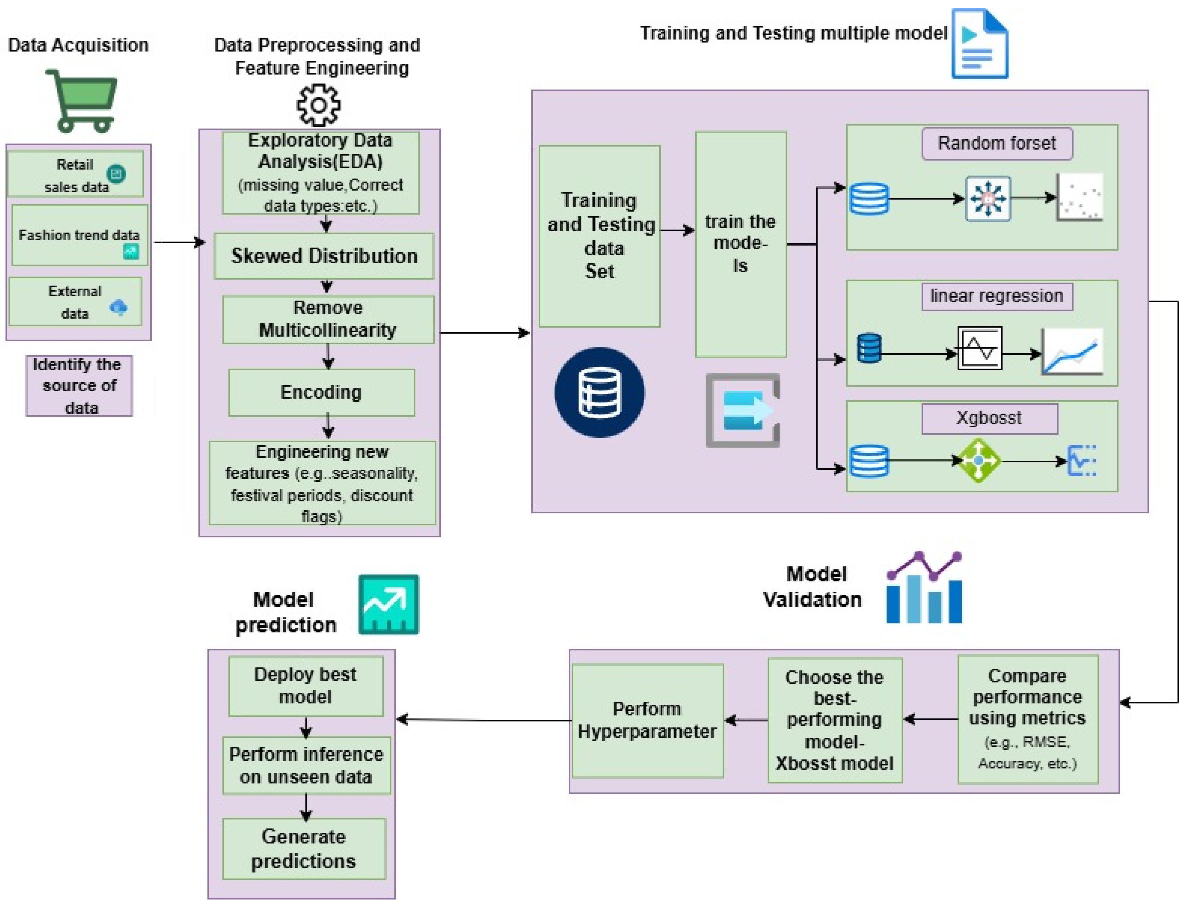


Figure 5.1: Trend Forecasting Workflow Diagram

the tested algorithms, XGBoost demonstrates superior results, consistently achieving the highest accuracy and lowest error rates. This performance is attributed to its gradient boosting framework, which incorporates additive tree models and advanced regularization techniques.

- **Model Validation**

  Model evaluation is conducted using standard regression metrics, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2). The XGBoost model achieved an R^2 score of 0.89, reflecting strong predictive capability and reliability in forecasting fashion demand trends.

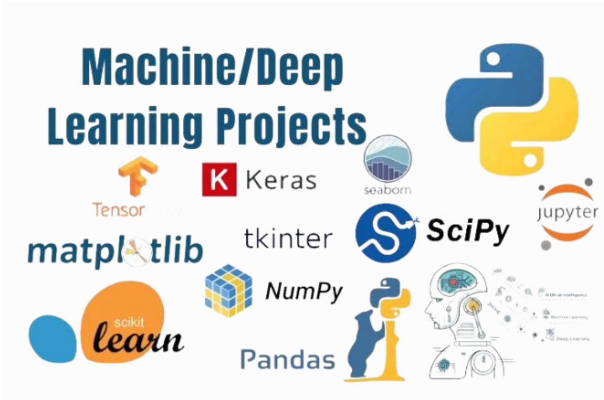
- **Model Deployment and Prediction**

  The final, optimized model is deployed within the Jupyter Notebook environment. Once trained, it is capable of generating real-time predictions based on unseen data inputs. The forecasting process involves passing the input features through an ensemble of decision trees, with predictions aggregated to produce final demand estimates. This is enhanced by XGBoost's use of second-order optimization and built-in regularization, which contribute to high computational efficiency and reduced risk of overfitting.

- **Platform Used**

  The Trend Forecasting Module is developed using the Python programming language within the Jupyter Notebook environment. Jupyter Notebook is an open-source, interactive platform that enables the creation and execution of live code, integrated with visualizations and descriptive text. Its user-friendly, cell-based interface makes it well-suited for data-driven tasks, including preprocessing, feature engineering, model training, and results analysis.

  Python was chosen for its clarity, flexibility, and the extensive ecosystem of machine learning libraries it supports. Tools such as Pandas, NumPy, Scikit-learn, XGBoost, Matplotlib, and Seaborn were utilized to facilitate data manipulation, statistical computation, and the implementation of predictive models. These libraries streamline complex processes and enable the development of sophisticated forecasting solutions.



Jupyter Notebook's modular execution framework allows for incremental development, real-time observation of intermediate results, and efficient model experimentation. This functionality is particularly advantageous for tasks such as hyperparameter tuning, model comparison, and evaluation of forecasting accuracy. The platform's strong support for rich visual outputs further enhances its utility in analyzing patterns, trends, and performance metrics.

While cloud-based environments such as Google Colab and AWS are commonly used for similar projects, this module was implemented locally in Jupyter Notebook to retain full control over the data pipeline and configuration settings. Nonetheless, the system is designed to be portable and can be easily transitioned to a cloud platform for future deployment, if required.

Overall, the combination of Jupyter Notebook and Python's powerful libraries provided a robust, transparent, and highly customizable environment for the successful development and validation of the trend forecasting module.

5.4 Website Integration Using Streamlit

Website integration is a crucial part of the *My Academy* project, as it connects the backend machine learning models and data processing with a user-friendly frontend interface. Streamlit was chosen for this integration because it simplifies the development of interactive web applications using Python, without requiring deep frontend development expertise. This enables rapid prototyping, smooth deployment, and efficient interaction between users and the forecasting system.

5.4.1 Streamlit Version Used

For this project, **Streamlit version 1.22.0** was utilized. This version includes various enhancements in stability, performance, and widget features that fit the project needs.

To realize the website integration for the My Academy project, the following technologies were utilized:

- **Streamlit (v1.22.0)**: Used to develop the general frontend interface, facilitating rapid development of web applications using pure Python.
- **Python**: Used as the underlying backend programming language, powering the logic, data processing, and ML model integration.
- **Pandas and NumPy**: Utilized for efficient data manipulation and preprocessing before offering data to ML models.
- **Streamlit Widgets**: Used for user input such as forms, sliders, dropdowns, and buttons.
- **HTML/CSS Customization** (through Streamlit themes): Used for branding purposes, i.e., positioning of the logo, color palettes, and fonts to align with the My Academy look.

5.4.2 Why Streamlit 1.22.0 Was Chosen

- Stability and reliability for a smooth and uninterrupted user experience.
- Advanced widgets such as sliders, dropdowns, buttons, and forms for interactive user input.
- Improved performance for fast rendering and real-time data updates.
- Easy integration with Python-based ML models and data pipelines.
- Cross-platform compatibility for seamless usage across devices.
- Strong community support and comprehensive documentation.

5.4.3 Features of the My Academy Website Using Streamlit

- Interactive and responsive UI with navigation across sections like Home, Forecast, Trends, About, and Contact.
- Real-time prediction output based on user inputs.
- Dynamic charts and graphs for visualizing demand and trends.
- Custom branding with logos, colors, and fonts matching the My Academy identity.
- Team introduction section and functional contact form.
- Lightweight deployment process with easy maintenance and updates.

5.4.4 Advantages of Using Streamlit for Website Integration

- Rapid application development with minimal code.
- Direct integration with Python's scientific and ML libraries.
- Enhanced user engagement with responsive controls and visual feedback.
- Compatible across devices including desktops and mobile platforms.
- Open-source and extensible framework for future growth.

5.5 Backend and Data Handling

The backend architecture of the MY Academy platform functions as the core technical infrastructure supporting its interactive components, including the Fashion Chatbot, Ramp Walk Tutor, and Trend Forecasting Module. Developed using the Python programming language, the backend is designed with a modular approach to efficiently handle user inputs, manage real-time video processing, execute data analytics, and interface with the front end via the Streamlit framework.

This backend system is engineered to accommodate various data types—text, visual inputs, and structured tabular data—ensuring compatibility across modules. Its scalability and robustness are critical for maintaining smooth application performance and responsiveness, particularly in real-time scenarios that demand efficient data flow and minimal latency.

5.5.1 Backend Architecture and Technologies

The backend of the MY Academy platform is developed entirely in Python, selected for its simplicity, flexibility, and extensive library support for machine learning, natural language processing, and computer vision. The system adopts a modular architecture, where each component is built to interact efficiently with others while remaining independently maintainable. Streamlit serves as the interface layer, seamlessly connecting backend logic with real-time, browser-based user interaction.

The core technologies and their roles within the backend system are as follows:

- Python: Serves as the foundational language for implementing backend logic, data processing routines, and inter-module communication.
- Streamlit: Facilitates the development of an interactive web-based user interface directly linked to Python code, enabling real-time feedback and visualization.
- OpenCV: Handles video stream capture and processing, primarily utilized in the Ramp Walk Tutor for webcam-based pose tracking.
- MediaPipe: A Google library employed for detecting 33 human body landmarks, essential for analyzing user posture during ramp walk sessions.
- Pandas and NumPy: Provide robust tools for data manipulation, cleaning, and feature engineering, particularly in the Trend Forecasting Module.
- Scikit-learn and XGBoost: Applied in the training, evaluation, and optimization of machine learning models used for fashion trend prediction.
- Matplotlib and Plotly: Used for generating static and interactive visualizations to support trend analysis and present model performance metrics.
- JSON: Employed to store configuration files and intent-response mappings, particularly in the chatbot module for managing dialogue flows.

This comprehensive technology stack ensures that the backend remains scalable, efficient, and adaptable to evolving project requirements while maintaining smooth user interaction and reliable system performance.

5.5.2 Streamlit Integration

Streamlit serves as the primary interface framework that bridges the backend modules with the user-facing application. It enables backend functionalities—ranging from data processing and pose detection to natural language interaction—to be accessed directly through an interactive, browser-based interface. Streamlit provides built-in UI components such as text inputs, buttons, sliders, and data visualizations, facilitating seamless and intuitive user engagement.

Practical implementations include:

- A text input field allows users to communicate with the chatbot and receive instant responses.
- A button activates the webcam, enabling real-time analysis for the Ramp Walk Tutor.
- Interactive charts visualize trend forecasting results based on user-selected time frames or uploaded datasets.

The reactive, serverless architecture of Streamlit removes the need for complex front-end development while still delivering real-time responsiveness and dynamic user feedback.

5.5.3 Data Types and Handling Strategies

The backend of the MY Academy platform manages multiple data types, each processed through specialized strategies to ensure accuracy, efficiency, and real-time responsiveness across modules.

- **Textual Data (Chatbot Module)** Text-based data used in the chatbot module is stored in .json format. Each entry defines user intents, comprising example input patterns and corresponding response sets. Upon receiving input, the backend tokenizes and matches it to predefined patterns, retrieving an appropriate response. Streamlit's session state is utilized to maintain conversational flow and context across interactions.

- **Visual Data (Pose Detection in Ramp Walk Tutor)** Live visual data is captured via webcam using OpenCV, processed frame by frame. Each frame is passed to MediaPipe, which returns 2D coordinates of 33 human body landmarks. These landmarks are analyzed to compute joint angles and spatial relationships, enabling posture evaluation during ramp walk sessions. For privacy and performance, all processing occurs in-memory without storing image data.

- **Structured Tabular Data (Trend Forecasting Module)** Structured data, typically in CSV format, includes historical sales figures, seasonal indicators, and promotional metadata. Pandas is used to load and preprocess the data. Preprocessing steps include handling missing values, encoding categorical fields, normalizing numerical variables, and generating derived features such as event flags and seasonality indicators. The cleaned dataset is then used to train machine learning models—primarily XGBoost—to produce future trend forecasts. These predictions are visualized using Plotly and presented within the Streamlit interface.

- **Real-Time Visualizations** Dynamic charts and dashboards are generated in response to user inputs and system outputs. The backend employs Matplotlib and Plotly for creating both static and interactive visualizations. These tools integrate seamlessly with Streamlit's rendering engine to provide real-time graphical feedback, enhancing user interaction and data interpretation.

5.5.4 Modular Backend Design

The backend architecture of the MY Academy platform follows a modular design, with each core functionality implemented as a separate Python script or function. This approach enhances code organization, maintainability, and scalability, while also enabling isolated testing and debugging of individual components.

- chatbot: Manages intent recognition and response generation for the Fashion Chatbot module, using predefined patterns and structured responses.

- Ramp Walk Tutor.ipynb: Processes video frames from the webcam and computes body landmarks using MediaPipe, supporting real-time pose analysis in the Ramp Walk Tutor.
- Trend_Demand.ipynb: Handles data loading, preprocessing, machine learning model training, and generation of trend predictions in the Trend Forecasting Module.

The modular structure ensures that each script can be developed, tested, and updated independently, allowing for efficient integration of new features and easier maintenance over time.

5.5.5 Data Flow and Storage

As a prototype, the MY Academy platform currently operates without a persistent database. Instead, data handling is managed through lightweight and temporary storage methods appropriate for development and testing purposes:

- CSV Files: Used to store structured data such as historical fashion trends, promotional metadata, and seasonal indicators for the Trend Forecasting Module.
- JSON Files: Contain intent definitions and response mappings for the Fashion Chatbot.
- In-Memory Processing: Applied for real-time video input, pose analysis, and prediction outputs to ensure low latency and maintain user privacy.

While the present implementation is optimized for prototyping and local execution, the system architecture is designed with scalability in mind. Future enhancements may include integration with cloud-based storage solutions (e.g., AWS S3 or Google Cloud Storage), relational databases (e.g., MySQL or PostgreSQL), or NoSQL platforms like Firebase to enable persistent data tracking, user-specific customization, and broader deployment capabilities.

The backend architecture of MY Academy is structured for efficiency, modularity, and scalability. It facilitates real-time chatbot interactions, body posture analysis, and fashion trend forecasting—all implemented using a single programming language, Python, and seamlessly integrated through the Streamlit interface. The system processes diverse data types, including textual, visual, and tabular formats, using robust and optimized backend logic. This ensures high responsiveness, minimal latency, and a smooth user experience. While tailored to support the current academic prototype, the architecture is designed with future scalability in mind, enabling straightforward enhancement and deployment across broader platforms.

Chapter 6: Testing

6.1 Testing Overview

Testing is a vital phase within the Software Development Life Cycle (SDLC), aimed at verifying that the system operates reliably, fulfills both functional and non-functional requirements, and provides a seamless user experience. In the context of the MY Academy project—which integrates multiple AI-powered components including a Fashion Chatbot, a Ramp Walk Tutor leveraging pose detection, and a Trend Forecasting Module based on machine learning—comprehensive testing was conducted at both unit and system levels.

The key objectives of the testing process were as follows:

- Detect and resolve defects within individual modules
- Ensure seamless integration and interoperability among all modules
- Validate the accuracy and responsiveness of AI-driven functionalities
- Assess system performance across a range of input scenarios

This structured approach to testing helped ensure the stability, reliability, and user-centered functionality of the platform.

6.2 Types of Testing Performed

The MY Academy platform underwent multiple levels of testing to ensure functionality, reliability, and usability. These included unit testing, integration testing, system testing, and user acceptance testing (UAT), each targeting specific quality assurance goals.

- **Unit Testing :** Unit testing was conducted to validate individual functions and modules in isolation. Each Python script—whether related to chatbot logic, video processing, or machine learning—was tested using mock inputs to confirm the correctness of outputs. Examples include:

Verifying the chatbot engine's ability to match user input to appropriate intents.

Ensuring the pose detection module accurately identified body landmarks from webcam feeds.

Testing trend forecasting models to produce correct regression outputs using known test datasets.

- **Integration Testing :** Following unit validation, integration testing was performed to assess how the independently developed modules functioned together within the Streamlit interface. Key areas tested included:

  Input-output flows in the chatbot component via the user interface.

  Real-time video capture and feedback delivery in the pose detection module.

  Seamless display of forecasting results from machine learning models within the dashboard.

- **System Testing :** System testing evaluated the complete application under realistic usage conditions. The goal was to assess the platform's end-to-end behavior across all modules. Test scenarios included:

  Interacting with the chatbot through conversational inputs.

  Using the webcam for real-time analysis in the Ramp Walk Tutor.

  Generating and interpreting visual trend predictions from uploaded datasets. The system's performance, response time, and stability were monitored under varying input conditions.

- **User Acceptance Testing (UAT) :** UAT was conducted with a group of sample users, including students and individuals from both fashion and computer science backgrounds. Feedback was collected on usability, accessibility of features, output clarity, and overall user satisfaction. Based on the input received, several refinements were made, including minor UI adjustments and improvements to chatbot content.

6.3 Tools Used for Testing

A combination of automated and manual tools was utilized to ensure the accuracy, reliability, and usability of the MY Academy platform throughout the testing process:

- Python's unittest Framework and assert Statements: Employed for conducting unit tests on individual functions and scripts. These tools facilitated validation of expected outputs under controlled input conditions.

- Manual Testing: Used to evaluate integration points and user interface flows within the Streamlit environment. This approach helped identify logical inconsistencies and UI-related issues during module interaction.

- Streamlit Test Sessions: Enabled real-time verification of component behavior within the browser interface. Interactive elements such as input fields, buttons, and visualizations were tested under various usage scenarios.

- Peer Feedback and Observational Testing: Conducted during User Acceptance Testing (UAT) sessions with target users. Observations and feedback provided insights into usability, responsiveness, and feature accessibility, leading to targeted refinements.

6.4 Bug Tracking and Resolution

During early integration testing, the following issues were identified and addressed:

- Chatbot Response Failure : The chatbot occasionally failed to handle unknown inputs. This was resolved by implementing fallback responses to ensure consistent interaction.

- Pose Detection Lag : On lower-end machines, pose detection exhibited noticeable lag. Frame rate and resolution optimizations were applied to enhance performance.

- Forecast Graph Misalignment : Inaccurate alignment in forecast graphs due to date parsing and axis scaling issues was corrected, resulting in proper visualization.

All bugs were meticulously logged, resolved through iterative debugging, and re-tested to ensure system stability prior to final deployment

6.5 Performance Metrics

- Chatbot Response Time : Approximately 0.5 seconds per query, ensuring near-instant interaction.
- Pose Detection Frame Rate : Maintains 20–25 FPS on a mid-range laptop, delivering smooth performance for real-time use.
- Forecasting Model Accuracy : Achieved an R^2 score of approximately 0.89, indicating high predictive accuracy.
- Application Uptime : Demonstrated 100% stability during continuous usage exceeding 30 minutes in test scenarios.

Comprehensive testing confirmed that all modules of MY Academy functioned as intended and integrated seamlessly into the final platform. Each component underwent both unit and system-level testing to verify its functionality, performance, and usability.

This rigorous testing phase significantly enhanced overall system stability and ensured a smooth user experience. The successful execution of all test cases, coupled with positive user feedback, validates that the platform is stable and ready for academic presentation as well as potential future deployment.

chapter 7: Results and Discussion

My Academy web portal has been able to combine different key considerations such as data science, machine learning, and new-generation web technologies to provide a wide and complete solution that specifically deals with the issue of fashion trend and demand forecasting. The portal has been built with utmost caution using Streamlit, offering a smooth and seamless engagement between the users and the advanced backend prediction models driving the portal.

Key outcomes and observations include:

- **User-Friendly Interface:**The website has a simple and user-friendly interface, enabling users to navigate pages like Home, Forecast, Trends, About, Team, and Contact easily.

- **Real-Time Prediction Output:**Attributes such as season, sex, age range, or fashion type can be entered by users to receive real-time predictions. The outputs are automatically visualized with dynamic bar charts and graphs for better understanding and user-friendliness.

- **Model Accuracy and Performance:**The internal machine learning models, especially XGBoost, achieved an R^2 value of 0.89. This signifies that the model is effective at making accurate fashion demand trend predictions.

- **Dynamic Visualizations:**Trend forecasts and trends are presented with interactive visual aids, such that the information is made more palatable and actionable for the users, including designers and retailers.

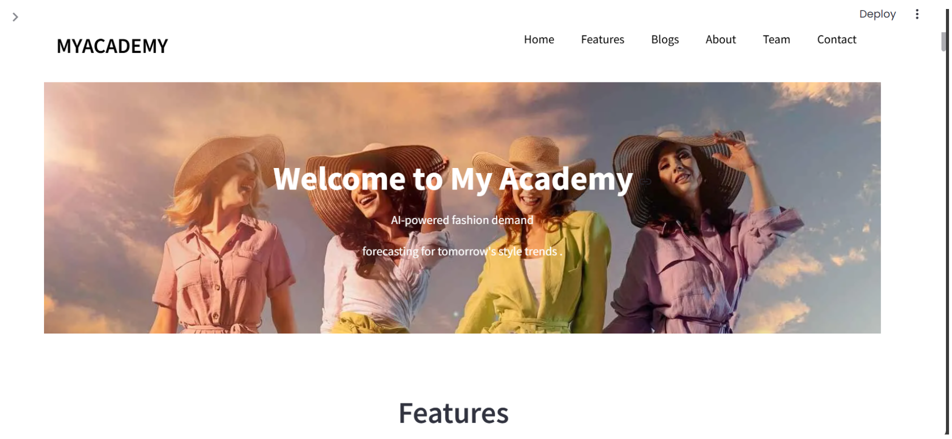
- **Responsiveness Across Devices:**The site is optimized for desktops, tablets, and smartphones so that the site can be accessed by more people.

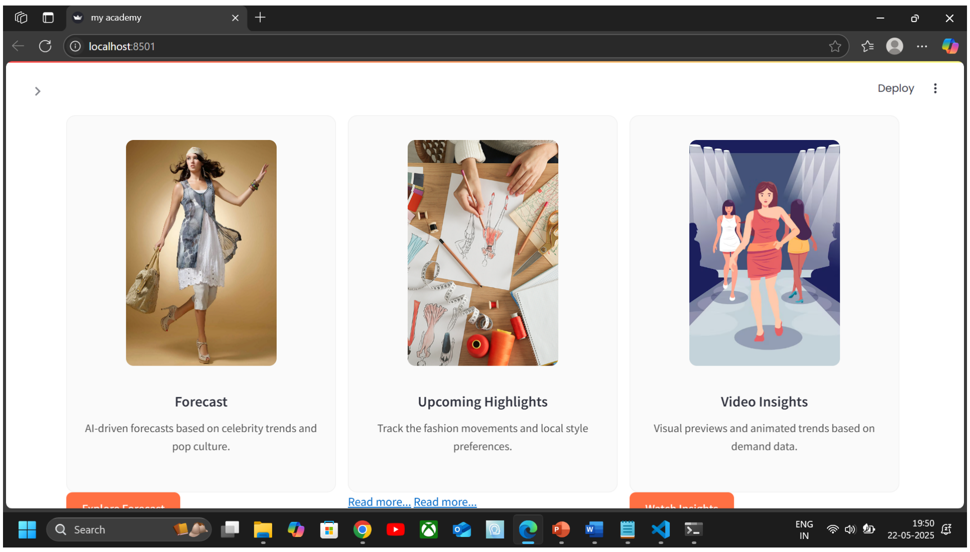
- **Integration of Branding Elements:**The site has personalized branding with the "My Academy" logo, uniform color schemes, and a professional look so as to establish trust and identity.

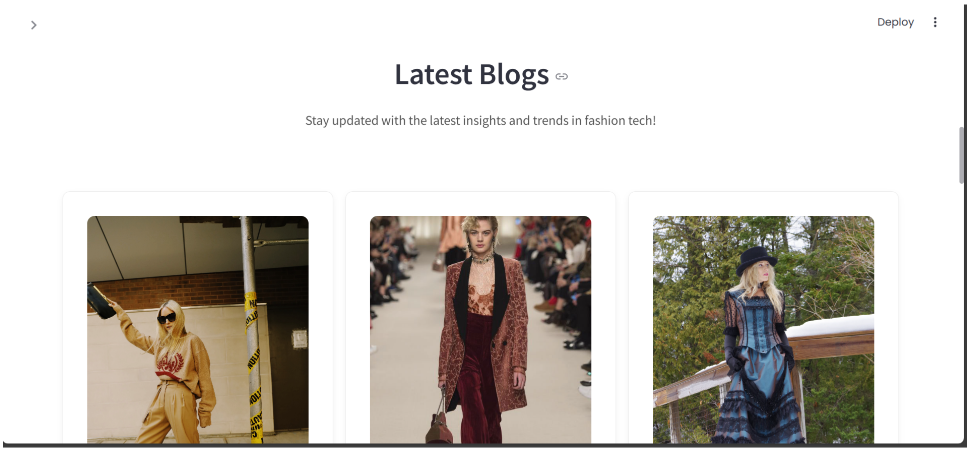
- **Backend–Frontend Synchronization:**The integration of the ML models to the frontend is smooth, enabling real-time processing and presentation of data without requiring technical expertise from the user.

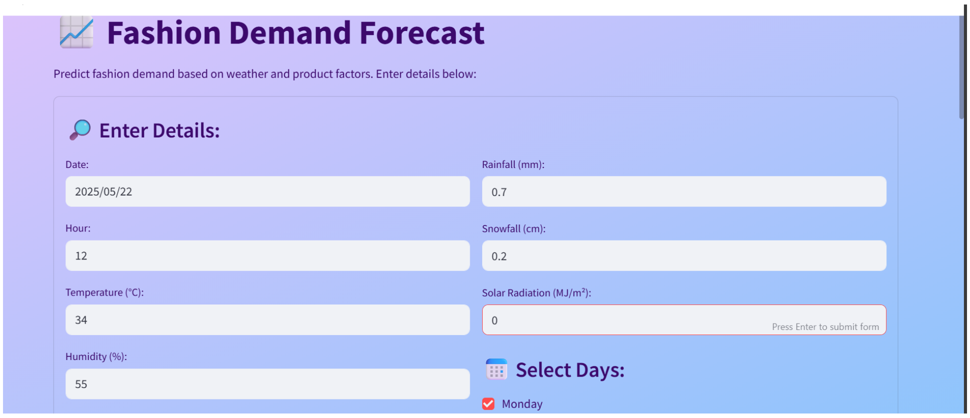
In conclusion, the My Academy website demonstrates how machine learning can be effectively applied to a real-world problem—fashion demand forecasting—while maintaining a focus on user experience, performance, and practical usability. The project validates the potential of AI-powered platforms in supporting strategic decisions in the fashion and retail industry.

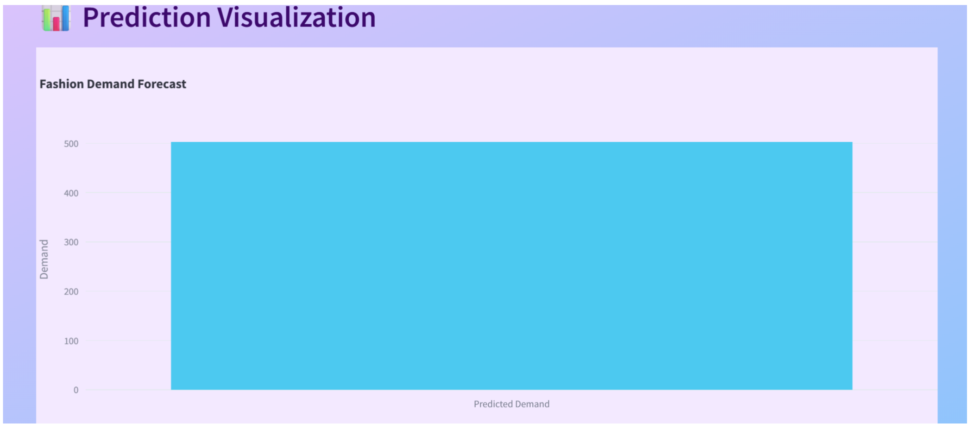
Output

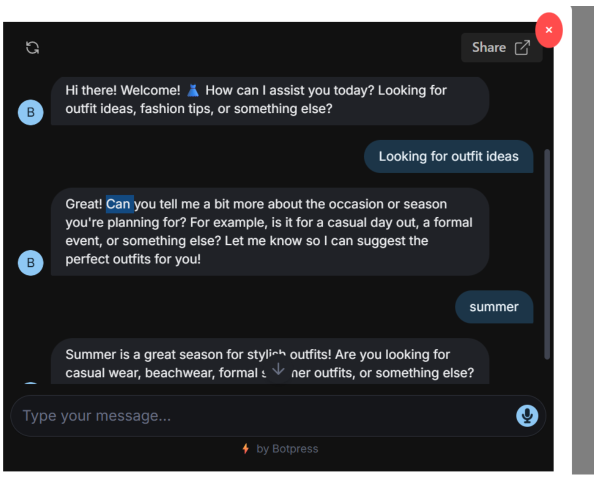





Deploy
Latest Blogs
Stay updated with the latest insights and trends in fashion tech!


Fashion Demand Forecast
Predict fashion demand based on weather and product factors. Enter details below:
Enter Details:
Date:
2025/05/22
Rainfall (mm):
0.7
Hour:
12
Snowfall (cm):
0.2
Temperature (°C):
34
Solar Radiation (MJ/m²):
0
Press Enter to submit form
Humidity (%):
55
Select Days:
Monday


Prediction Visualization
Fashion Demand Forecast
500
400
300
200
100
0
Demand
Predicted Demand


Share
Hi there! Welcome! How can I assist you today? Looking for outfit ideas, fashion tips, or something else?
Looking for outfit ideas
Great! Can you tell me a bit more about the occasion or season you're planning for? For example, is it for a casual day out, a formal event, or something else? Let me know so I can suggest the perfect outfits for you!
summer
Type your message...
by Botpress

chapter 6 Conclusion and Recommendations

The development of My Academy marks a significant step toward digitizing and democratizing fashion education and engagement. The platform successfully brings together AI-driven fashion guidance, interactive tutorials, real-world event opportunities, and designer networking into one cohesive system.

This user-centered approach has the potential to support aspiring models, designers, and fashion enthusiasts by offering guidance, exposure, and professional connections. The Fashion Bot and Ramp Walk Tutor modules particularly demonstrate how AI and digital education tools can reshape the learning curve in fashion grooming.

Recommendations for future improvements:

- Integrate video feedback in the Ramp Walk Tutor using webcam support for real-time correction.
- Add multilingual support to cater to diverse user groups across India and beyond.
- Improve Fashion Bot training with more datasets, including trending global fashion feeds.
- Enable chat-based designer consultations or bookings via integrated messaging.
- Deploy the platform as a progressive web app (PWA) to improve mobile access and offline use.

Overall, My Academy sets the stage for a new wave of interactive, accessible fashion learning and collaboration, opening up opportunities in both educational and professional fashion domains.

Bibliography

[1] F. Chen and C.-J. Lu (2021) Demand Forecasting for Multichannel Fashion Retailers by Integrating Clustering and Machine Learning Algorithms.

[2] M. Li, X. Fu, and D. Li (2020) Diabetes Prediction Based on XGBoost Algorithm.

[3] P. K. Singh, Y. Gupta, N. Jha, and A. Rajan, (2019) Fashion Retail: Forecasting Demand for New Items.

[4] M. Amjad, I. Ahmad, M. Ahmad, P. Wróblewski, U. Amjad, and P. Kamiński (2022) Prediction of Pile Bearing Capacity Using XGBoost Algorithm: Modeling and Performance Evaluation.

[5] S. Ramraj, N. Uzir, S. R., and S. Banerjee. Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets.

[6] S. Chen, J. Fan, E. Pishgar, K. Alaei, G. Placencia, and M. Pishgar 25 Feb (2025). XGBoost-Based Prediction of ICU Mortality in Sepsis-Associated Acute Kidney Injury Patients Using MIMIC-IV Database with Validation from eICU Database.

[7] Xin Zou, Qiaoyun Wang, Yinji Chen, Jilong Wang, Shunyuan Xu, Ziheng Zhu, Chongyue Yan, Peng Shan, Shuyu Wang, YongQing Fu (2025) Fusion of convolutional neural network with XGBoost feature extraction for predicting multi-constituents in corn using near infrared spectroscopy.

[8] Liuyi Ling,Liyu Wei,Bin Feng, Zhipeng Yu, Long Wang (2024) sEMG-Based Knee Angle Prediction: An Efficient Framework with XGBoost Feature Selection and Multi-Attention LSTM.

[9] Yali Guo*, Meng Chen, Can Yuan, Fei Zheng (2024) Research on Campus Network Security Situational Awareness Technology Based on XGBoost Machine Learning.

[10] Amal Asselman, Mohamed Khaldi,Souhaib Aammou(2021) Enhancing the prediction of student performance based on the machine learning XGBoost algorithm.

[11] Tamanna,Ms. Shivani Kamboj,Lovedeep Singh ,Tanvir Kaur (2024) Automated Fraud Detection in Financial Transactions using Machine Learning: An Ensemble Perspective.

[12] Bingyue Pan (2018). Application of XGBoost algorithm in hourly PM2.5 concentration prediction.

[13] C. C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages.

[14] Peter R.Winters , Forecasting sales by exponentially weighted moving averages.

[15] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, Time Series Analysis: Forecasting and Control 5th Edition.

[16] Ajex J.Smola and Bernhard Scholkopf, A tutorial on support vector regression.

[17] Leo Breiman, Random forests Machine Learning, vol. 45, pp. 5–32, 2001.

[18] Tianqi Chen and Carlos Gulstrin, XGBoost: A scalable tree boosting system

[19] Real Carbonneau, Kevin Laframboise, Rustam Vahidov,Application of machine learning techniques for supply chain demand forecasting.

[20] Roufeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, Dhruv Madeka,A multi-horizon quantile recurrent forecaster.

[21] K. Bandara, C. Bergmeir, and H. Hewamalage,LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns.

[22] Shenglong Li, Xiaojing Zhang (2019) Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm

[23] International beauty pageants and the onstruction of egemonic images of female beauty

Swapnil Srivastava

report.pdf

25

BioTech

Institut Seni Indonesia Surakarta

Document Details

Submission ID

trn:oid:::1:3257170251

Submission Date

May 22, 2025, 4:43 PM GMT+7

Download Date

May 22, 2025, 4:45 PM GMT+7

File Name

report.pdf

File Size

301.3 KB

40 Pages

7,229 Words

43,627 Characters

turnitin Page 1 of 45 - Cover Page Submission ID trn:oid:::1:3257170251

10% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

2/45

Match Groups

- **34** Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- **6** Missing Quotations 2%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 9% Internet sources
- 1% Publications
- 8% Submitted works (Student Papers)

Match Groups

- **34** Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- **6** Missing Quotations 2%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
1% Publications
8% Submitted works (Student Papers)

3/45

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	bbsbec.edu.in	2%
2	Student papers	Engineers Australia	1%
3	Internet	www.coursehero.com	1%
4	Internet	global.oup.com	<1%
5	Student papers	nith	<1%
6	Internet	www.united.ac.in	<1%
7	Internet	github.com	<1%
8	Student papers	National Institute of Technology, Silchar	<1%
9	Student papers	Camarines Sur Polytechnic Colleges	<1%
10	Student papers	PSB Academy (ACP eSolutions)	<1%