

## Capstone project use cases: React Stream

**Development stack : MERN**

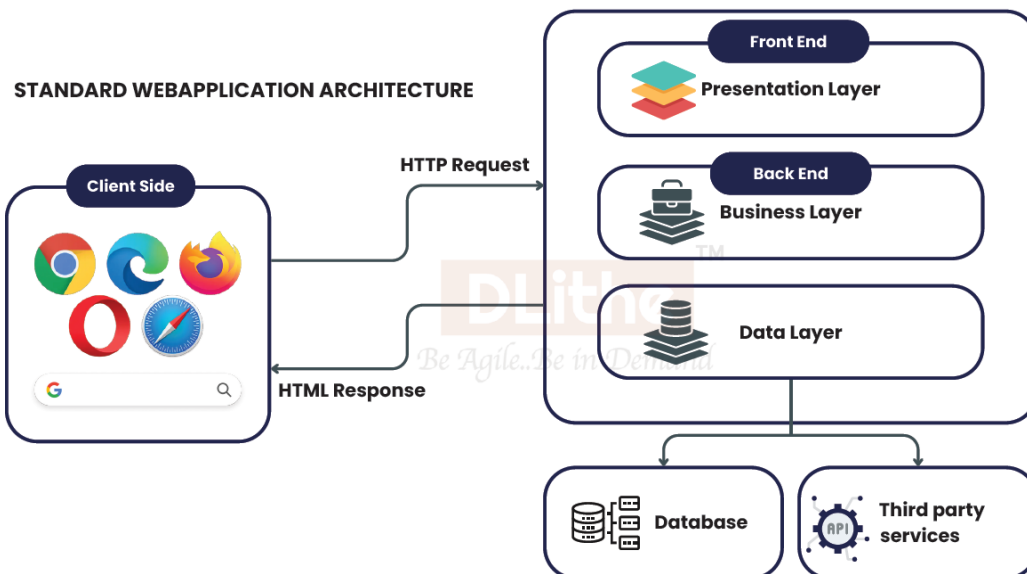
**Testing : Postman**

**Deployment:**

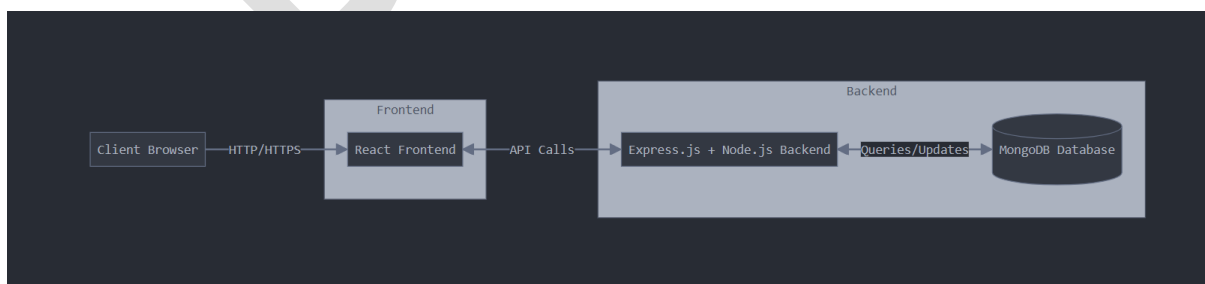
- Preparing React applications for production
- Deployment strategies (e.g., Netlify, Vercel)

**Documentation:**

- Git Repository



Source: DLithe Consultancy Services



## 1. E-commerce Platform

**Objective:** Build an online store with cart functionality, payment gateway, and order management.

### Modules and Milestones:

- User authentication and profile management
- Product catalog and inventory
- Cart and checkout integration
- Payment gateway and order history

### Tasks:

- Set up authentication using JWT
  - Create REST APIs for product management
  - Implement cart management and order processing
  - Integrate payment APIs (e.g., Stripe, PayPal)
- 

## 2. Social Media App

**Objective:** Create a social networking platform for users to post, like, comment, and share.

### Modules and Milestones:

- User profiles and authentication
- Post creation and media uploads
- Like, comment, and share functionality
- Feed and notification system

### Tasks:

- Implement OAuth2 for user login
  - Develop post management APIs
  - Build a real-time feed with socket.io
  - Integrate cloud storage for media files
- 

## 3. Online Learning Platform

**Objective:** Develop a platform for hosting courses, lessons, and student management.

### Modules and Milestones:

- User roles (Admin, Instructor, Student)
- Course creation and content management
- Student enrollment and progress tracking
- Quizzes and certification generation

**Tasks:**

- Design user role-based authentication
  - Develop REST APIs for course management
  - Build a lesson streaming feature
  - Implement quiz APIs and scoring system
- 

#### 4. Blog Platform

**Objective:** Create a content management system for writing and publishing blogs.

**Modules and Milestones:**

- User login and profile
- Blog creation, editing, and deletion
- Commenting system
- Search and category filtering

**Tasks:**

- Create a JWT-based authentication system
  - Develop REST APIs for blog CRUD operations
  - Build a commenting and reply system
  - Implement search and filtering endpoints
- 

#### 5. Task Management Tool

**Objective:** Create a to-do/task management app with team collaboration features.

**Modules and Milestones:**

- User authentication and team creation
  - Task creation and assignment
  - Task tracking and deadline management
  - Notifications and reminders
-

**Tasks:**

- Implement authentication and team roles
  - Develop APIs for task management
  - Build a real-time notification system with websockets
  - Integrate calendar for deadlines
- 

## 6. Event Booking System

**Objective:** Build a system for users to book tickets for events and manage bookings.

**Modules and Milestones:**

- User profiles and event listings
- Ticket booking and payment
- Event management and updates
- Notifications for upcoming events

**Tasks:**

- Create a REST API for event management
  - Build the booking and payment flow with Stripe
  - Integrate email notifications for event updates
  - Implement seat reservation functionality
- 

## 7. Job Board Platform

**Objective:** Develop a platform for job seekers and employers to post and apply for jobs.

**Modules and Milestones:**

- User roles (Job Seeker, Employer)
- Job posting and application
- Resume upload and tracking
- Employer dashboard for managing listings

**Tasks:**

- Build a role-based authentication system
  - Develop APIs for job listings and applications
  - Implement resume upload and storage
  - Create an employer dashboard for job management
-

---

## 8. Online Polling System

**Objective:** Create a system for users to create, share, and participate in polls.

**Modules and Milestones:**

- User profiles and poll creation
- Poll voting and result display
- Poll sharing via social media
- Poll result analytics

**Tasks:**

- Set up user authentication and poll creation APIs
  - Build a real-time polling system with results visualization
  - Develop social media sharing integration
  - Implement poll result analytics and insights
- 

## 9. Fitness Tracking App

**Objective:** Develop an app to track users' workouts and fitness goals.

**Modules and Milestones:**

- User profiles and fitness goals
- Workout logging and tracking
- Progress visualization and analytics
- Notifications and challenges

**Tasks:**

- Design user authentication and fitness goal APIs
  - Build a workout logging system
  - Implement progress tracking with charts
  - Integrate push notifications for challenges and milestones
- 

## 10. Real-Time Chat App

**Objective:** Create a real-time messaging platform for users to communicate.

---

### Modules and Milestones:

- User login and profile
- One-to-one and group chats
- Message notifications and media sharing
- Chat history and search

### Tasks:

- Set up WebSocket connections for real-time chat
  - Develop APIs for user authentication and chat management
  - Build a message history search feature
  - Implement media sharing and notifications
- 

## 11. CRM System

**Objective:** Build a customer relationship management platform for sales teams.

### Modules and Milestones:

- User authentication and roles
- Lead and contact management
- Sales pipeline tracking
- Reporting and analytics

### Tasks:

- Create role-based user management
  - Develop APIs for lead and contact tracking
  - Build sales pipeline and progress tracking
  - Implement reporting and analytics dashboards
- 

## 12. Inventory Management System

**Objective:** Develop a system for managing product inventory and stock levels.

### Modules and Milestones:

- User roles and permissions
  - Product and stock management
  - Stock level tracking and reordering
-

- Reporting and stock history

**Tasks:**

- Design user role-based access control
  - Build APIs for product and inventory management
  - Implement real-time stock level tracking
  - Develop reports for stock movement and reorder suggestions
- 

### 13. Expense Tracker

**Objective:** Build a tool for users to log and track personal or business expenses.

**Modules and Milestones:**

- User authentication and expense categories
- Expense logging and tracking
- Budget setting and alerts
- Reports and analytics

**Tasks:**

- Create authentication and category management
  - Develop APIs for expense logging
  - Implement budget tracking and alert system
  - Build visual analytics for expenses
- 

### 14. Video Streaming Platform

**Objective:** Develop a platform for users to upload, stream, and watch videos.

**Modules and Milestones:**

- User authentication and profile
- Video upload and processing
- Video streaming and playback
- Comments and recommendations

**Tasks:**

- Implement video upload and cloud storage integration
-

- Build APIs for video management and streaming
  - Create a comment and recommendation system
  - Integrate a video player with buffering optimization
- 

## 15. Crowdfunding Platform

**Objective:** Create a platform for users to raise funds for their projects.

### Modules and Milestones:

- User authentication and project creation
- Donation tracking and goal management
- Project updates and sharing
- Payment gateway integration

### Tasks:

- Set up user authentication and project APIs
  - Build donation tracking and goal monitoring system
  - Integrate payment APIs for contributions
  - Implement project update and sharing features
- 

## 16. Travel Booking System

**Objective:** Develop a platform for booking flights, hotels, and rental cars.

### Modules and Milestones:

- User profiles and travel search
- Booking and payment processing
- Booking history and cancellations
- Notifications and customer support

### Tasks:

- Build travel search and booking APIs
  - Integrate payment gateway for bookings
  - Implement booking history and cancellation feature
  - Develop customer support ticketing system
-



## 17. Online Marketplace

**Objective:** Create an online marketplace where users can buy and sell products.

**Modules and Milestones:**

- User authentication and roles (Buyer, Seller)
- Product listings and search
- Cart, checkout, and payment processing
- Order history and reviews

**Tasks:**

- Design user role-based authentication
  - Build APIs for product listing and search
  - Implement cart management and payment flow
  - Create a review and rating system for products
- 

## 18. Document Sharing Platform

**Objective:** Create a system for users to upload, share, and collaborate on documents.

**Modules and Milestones:**

- User authentication and document upload
- Sharing and collaboration features
- Document versioning and history
- Notifications and user management

**Tasks:**

- Develop document upload and storage APIs
  - Build sharing and collaboration functionality
  - Implement document version control
  - Set up notifications for document updates
- 

## 19. Ride Sharing App

**Objective:** Develop a platform where users can request and offer rides.

**Modules and Milestones:**

- User roles (Driver, Rider)
- Ride request and matching system
- Payment and trip tracking
- Notifications and trip history

**Tasks:**

- Set up user authentication and role management
  - Build APIs for ride request and matching
  - Integrate payment and trip tracking systems
  - Implement notifications and trip history
- 

## 20. Food Delivery System

**Objective:** Create a system where users can order food from nearby restaurants.

**Modules and Milestones:**

- User authentication and restaurant management
- Menu and order placement
- Delivery tracking and payment
- Notifications and reviews

**Tasks:**

- Develop APIs for restaurant and menu management
  - Build an order placement and tracking system
  - Integrate payment and delivery APIs
  - Implement notification and review functionality
- 

## 21. Event Management Tool

**Objective:** Build a platform to manage and organize events, from invitations to RSVPs.

**Modules and Milestones:**

- User authentication and event creation
  - Invitation and RSVP tracking
  - Event updates and notifications
  - Event history and analytics
- Tasks:**
- Build event creation and RSVP tracking APIs
  - Implement invitation and notification features
-

- Develop event history and analytics dashboard
  - Create user authentication and role-based access control
- 

## 22. Property Rental System

**Objective:** Develop a platform for listing and booking rental properties. **Modules and Milestones:**

- User roles (Owner, Renter)
  - Property listing and search
  - Booking and payment processing
  - Reviews and property management **Tasks:**
  - Set up role-based authentication
  - Build APIs for property listing and search
  - Implement booking and payment integration
  - Develop review and property management system
- 

## 23. Employee Management System

**Objective:** Build a tool to manage employee data, attendance, and performance. **Modules and Milestones:**

- User roles (Admin, Employee)
  - Attendance tracking and leave management
  - Performance review and analytics
  - Payroll and notifications **Tasks:**
  - Create authentication and role-based access control
  - Build APIs for attendance and leave tracking
  - Develop performance review and analytics system
  - Implement payroll integration and notifications
- 

## 24. Appointment Scheduling System

**Objective:** Create a platform for users to book appointments with service providers. **Modules and Milestones:**

- User authentication and service provider profiles
  - Appointment booking and calendar integration
  - Notifications and reminders
-

- Appointment history and feedback

**Tasks:**

- Build service provider profile and calendar APIs
- Implement appointment booking and notifications
- Develop appointment history and feedback system
- Set up reminders with email/SMS integration

---

## **25. Customer Support Ticketing System**

**Objective:** Develop a platform for managing customer queries and support tickets.

**Modules and Milestones:**

- User authentication and support roles
- Ticket creation and assignment
- Ticket tracking and status updates
- Reporting and analytics

**Tasks:**

- Set up user authentication and role-based access
- Build APIs for ticket management and tracking
- Implement status updates and notifications
- Develop a reporting and analytics dashboard

## AIML Stream

1. End-to-End Predictive Model
2. Packaging ML Models
3. Dynamic Pricing Strategy
4. Music Popularity Prediction
5. Real Estate Price Prediction
6. Retail Price Optimization
7. Compare Multiple Predictive Models
8. Food Delivery Time Prediction
9. Diamond Price Prediction
10. Salary Prediction
11. House Rent Prediction
12. Classification on Imbalanced Data
13. Google Search Queries Anomaly Detection
14. Anomaly Detection in Transactions
15. Loan Approval Prediction
16. Classification with Neural Networks
17. Text Emotions Classification
18. Credit Score Classification
19. Ads Click-Through Rate Prediction
20. Consumer Complaint Classification
21. User Profiling and Segmentation
22. Credit Scoring & Segmentation
23. App User Segmentation
24. Credit Card Clustering
25. Topic Modelling
26. Clustering Music Genres

27. Music Recommendation System
28. Fashion Recommendations using Image Features
29. End-to-End Chatbot
30. YouTube Video Chaptering
31. Text Generation Model
32. Next Word Prediction Model
33. Generative AI Model From Scratch
34. Text Summarization Model with LLMs
35. Code Generation Model with LLMs
36. Text Generation with LLMs
37. Website Traffic Analysis & Forecasting
38. Ads CTR Forecasting
39. Multivariate Time Series Forecasting
40. Demand Forecasting & Inventory Optimization
41. Netflix Subscriptions Forecasting
42. Currency Exchange Rate Forecasting
43. Instagram Reach Forecasting
44. Time Series Forecasting with ARIMA
45. Weather Forecasting

### Generic architecture



Use cases:

### 1. End-to-End Predictive Model

- **Use Case Description:** Develop a pipeline that automates data collection, preprocessing, model training, and deployment for predictions.
- **User Goals:** Enable users to generate predictions seamlessly without manual intervention.
- **Main Flow:** Users upload data → Data preprocessing occurs → Model trains → Predictions are generated and displayed.
- **Alternate Flow:** Users can select specific models or parameters.
- **Exception Handling:** Handle missing data, invalid inputs, or server errors during model training.

### 2. Packaging ML Models

- **Use Case Description:** Create a framework for packaging machine learning models into deployable units (e.g., Docker containers).
- **User Goals:** Ensure easy deployment and scalability of ML models.
- **Main Flow:** User uploads model files → System packages files into a container → User deploys container to production.
- **Alternate Flow:** User customizes container configuration.
- **Exception Handling:** Handle packaging errors or missing dependencies.

### 3. Dynamic Pricing Strategy

- **Use Case Description:** Implement ML models to adjust product prices dynamically based on demand and market trends.
- **User Goals:** Maximize revenue by optimizing prices.
- **Main Flow:** System collects sales data and market trends → Model predicts optimal prices → Prices are updated in real-time.
- **Alternate Flow:** Users can manually override suggested prices.
- **Exception Handling:** Address pricing model failures or anomalies in pricing.

### 4. Music Popularity Prediction

- **Use Case Description:** Predict the popularity of songs based on features like genre, tempo, and artist.
- **User Goals:** Identify potential hits and optimize playlists.
- **Main Flow:** User inputs song features → Model predicts popularity score → User receives recommendations.
- **Alternate Flow:** User adjusts input features for different results.
- **Exception Handling:** Handle errors due to missing song data or corrupted files.

## 5. Real Estate Price Prediction

- **Use Case Description:** Predict property prices based on features like location, size, and amenities.
- **User Goals:** Help buyers/sellers make informed decisions.
- **Main Flow:** User inputs property details → Model predicts price → User reviews the estimated price.
- **Alternate Flow:** User refines property features for different predictions.
- **Exception Handling:** Address invalid inputs or feature missing from the dataset.

## 6. Retail Price Optimization

- **Use Case Description:** Use ML to determine optimal pricing strategies for retail products.
- **User Goals:** Maximize profit while maintaining competitiveness.
- **Main Flow:** System collects sales data and user preferences → Model suggests optimal prices → Prices are updated.
- **Alternate Flow:** User manually adjusts price suggestions.
- **Exception Handling:** Handle incorrect data inputs or API failures.

## 7. Compare Multiple Predictive Models

- **Use Case Description:** Allow users to compare the performance of different ML models on a given dataset.
- **User Goals:** Select the best model for their specific needs.
- **Main Flow:** User uploads dataset → System trains multiple models → Displays comparison metrics.
- **Alternate Flow:** User filters out models they are not interested in.
- **Exception Handling:** Handle errors in model training or incompatible datasets.

## 8. Food Delivery Time Prediction

- **Use Case Description:** Predict estimated delivery times based on factors like distance, weather, and order volume.
- **User Goals:** Improve delivery efficiency and customer satisfaction.
- **Main Flow:** User inputs delivery parameters → Model predicts delivery time → Time estimate is displayed.
- **Alternate Flow:** User adjusts parameters like time of day or location.
- **Exception Handling:** Handle outliers such as extreme weather conditions.

## 9. Diamond Price Prediction



- **Use Case Description:** Predict diamond prices based on features like cut, clarity, carat, and color.
- **User Goals:** Help buyers and sellers determine fair prices.
- **Main Flow:** User inputs diamond features → Model predicts price → User receives price estimate.
- **Alternate Flow:** User modifies diamond specifications.
- **Exception Handling:** Address missing or inconsistent feature values.

#### 10. Salary Prediction

- **Use Case Description:** Predict employee salaries based on experience, education, and industry.
- **User Goals:** Assist employers and job seekers with salary benchmarking.
- **Main Flow:** User inputs job details → Model predicts salary range → User reviews estimate.
- **Alternate Flow:** User adjusts input parameters for different estimates.
- **Exception Handling:** Handle missing or inaccurate data inputs.

#### 11. House Rent Prediction

- **Use Case Description:** Predict rental prices for houses based on location, size, and other amenities.
- **User Goals:** Help renters and landlords determine appropriate rental rates.
- **Main Flow:** User inputs house details (location, size, etc.) → Model predicts rental price → User views the estimate.
- **Alternate Flow:** User changes parameters like lease length or amenities.
- **Exception Handling:** Manage missing data or outlier inputs like unusual locations.

#### 12. Classification on Imbalanced Data

- **Use Case Description:** Train a classification model using techniques like oversampling/undersampling to handle imbalanced datasets.
- **User Goals:** Improve the performance of classification models on minority classes.
- **Main Flow:** User uploads dataset → System applies balancing techniques → Model is trained and evaluated.
- **Alternate Flow:** User selects a specific balancing technique (e.g., SMOTE).
- **Exception Handling:** Handle errors due to extreme imbalance or invalid sampling parameters.

#### 13. Google Search Queries Anomaly Detection

- **Use Case Description:** Detect anomalies in search trends using historical data.
- **User Goals:** Identify sudden spikes or drops in search interest.
- **Main Flow:** User inputs query and time range → Model identifies anomalies → Anomalous trends are displayed.
- **Alternate Flow:** User adjusts the sensitivity of anomaly detection.
- **Exception Handling:** Handle missing data or API connection failures.

#### 14. Anomaly Detection in Transactions

- **Use Case Description:** Detect fraudulent or suspicious transactions using ML models.
- **User Goals:** Reduce fraud and improve security.
- **Main Flow:** System receives transaction data → Model flags anomalies → Alerts are sent to users.
- **Alternate Flow:** User tunes anomaly thresholds.
- **Exception Handling:** Manage false positives or undetected fraud cases.

#### 15. Loan Approval Prediction

- **Use Case Description:** Predict whether loan applications will be approved based on applicant details.
- **User Goals:** Help financial institutions make quick and accurate lending decisions.
- **Main Flow:** User inputs applicant details → Model predicts approval status → Decision is displayed.
- **Alternate Flow:** User adjusts criteria or threshold.
- **Exception Handling:** Handle missing applicant information or out-of-bounds values.

#### 16. Classification with Neural Networks

- **Use Case Description:** Train a neural network for classification tasks like image or text categorization.
- **User Goals:** Achieve high accuracy on complex classification problems.
- **Main Flow:** User uploads dataset → Model trains on data → Predictions are made and evaluated.
- **Alternate Flow:** User modifies network architecture or hyperparameters.
- **Exception Handling:** Handle training failures due to overfitting or vanishing gradients.

#### 17. Text Emotions Classification

- **Use Case Description:** Classify text into different emotional categories (e.g., happy, sad, angry).

- **User Goals:** Understand user sentiments for applications like customer feedback.
- **Main Flow:** User inputs text data → Model predicts emotion → Results are displayed.
- **Alternate Flow:** User adjusts classification categories.
- **Exception Handling:** Address errors due to ambiguous text or mixed emotions.

#### 18. Credit Score Classification

- **Use Case Description:** Classify individuals into different credit score bands using historical credit data.
- **User Goals:** Help banks assess creditworthiness.
- **Main Flow:** User inputs customer data → Model classifies into score bands → User receives credit classification.
- **Alternate Flow:** User customizes score bands.
- **Exception Handling:** Manage missing or inconsistent input data.

#### 19. Ads Click-Through Rate Prediction

- **Use Case Description:** Predict the likelihood of users clicking on advertisements based on user data and ad content.
- **User Goals:** Optimize ad placements to improve ROI.
- **Main Flow:** System gathers user interaction data → Model predicts click-through rates → Adjusts ad placements accordingly.
- **Alternate Flow:** User tweaks features like user demographics.
- **Exception Handling:** Address errors due to data sparsity or sudden shifts in user behavior.

#### 20. Consumer Complaint Classification

- **Use Case Description:** Automatically categorize customer complaints into predefined categories.
- **User Goals:** Help customer support teams prioritize and route complaints efficiently.
- **Main Flow:** User inputs complaint text → Model categorizes complaint → Category is displayed.
- **Alternate Flow:** User adjusts the complaint categories.
- **Exception Handling:** Handle ambiguous complaints that don't clearly fit into categories.

#### 21. User Profiling and Segmentation

- **Use Case Description:** Segment users into different profiles based on their behaviors and interactions.
- **User Goals:** Create personalized marketing and user engagement strategies.

- **Main Flow:** System collects user behavior data → Model segments users into profiles → User reviews segments.
- **Alternate Flow:** User adjusts segmentation criteria.
- **Exception Handling:** Manage edge cases where user behavior is inconsistent or unique.

## 22. Credit Scoring & Segmentation

- **Use Case Description:** Combine credit scoring with customer segmentation for more tailored loan products.
- **User Goals:** Offer customized financial products based on user segments.
- **Main Flow:** User inputs customer data → Model generates credit score and segment → Recommendations are displayed.
- **Alternate Flow:** User overrides the segment assignments.
- **Exception Handling:** Handle conflicts between credit scores and segment assignments.

## 23. App User Segmentation

- **Use Case Description:** Segment app users based on their interaction patterns and usage frequency.
- **User Goals:** Improve user engagement through targeted features.
- **Main Flow:** System gathers app usage data → Model segments users → Segmentation results are displayed.
- **Alternate Flow:** User refines segmentation criteria.
- **Exception Handling:** Manage cases where usage data is sparse or inconsistent.

## 24. Credit Card Clustering

- **Use Case Description:** Cluster credit card users based on spending habits and transaction patterns.
- **User Goals:** Identify different user segments for targeted marketing.
- **Main Flow:** User uploads transaction data → Model clusters users → Clusters are displayed.
- **Alternate Flow:** User adjusts clustering parameters.
- **Exception Handling:** Handle outlier users with highly unusual spending.

## 25. Topic Modelling

- **Use Case Description:** Automatically identify topics within a set of documents using NLP techniques.
- **User Goals:** Summarize large volumes of text data for easier understanding.
- **Main Flow:** User inputs documents → Model extracts topics → Topics are displayed.

- **Alternate Flow:** User adjusts the number of topics.
- **Exception Handling:** Address low coherence between extracted topics.

## 26. Clustering Music Genres

- **Use Case Description:** Cluster music tracks based on their audio features (e.g., tempo, rhythm, and pitch) to identify different genres.
- **User Goals:** Help music streaming services organize their library and recommend music.
- **Main Flow:** User inputs audio feature data → Model clusters tracks into genres → Clusters are displayed.
- **Alternate Flow:** User adjusts clustering parameters or adds custom genre tags.
- **Exception Handling:** Handle errors with tracks that don't fit into any existing clusters.

## 27. Music Recommendation System

- **Use Case Description:** Recommend songs to users based on their listening history and preferences.
- **User Goals:** Improve user engagement by providing personalized music suggestions.
- **Main Flow:** User listens to songs → Model analyzes preferences → Recommendations are generated.
- **Alternate Flow:** User provides feedback to refine recommendations.
- **Exception Handling:** Handle sparse data for new users or unusual preferences.

## 28. Fashion Recommendations using Image Features

- **Use Case Description:** Suggest fashion items to users based on the analysis of image features like color, style, and texture.
- **User Goals:** Enhance user shopping experience with personalized clothing suggestions.
- **Main Flow:** User uploads fashion item images → Model extracts features and recommends similar items → Recommendations are displayed.
- **Alternate Flow:** User adjusts recommendation criteria like price range.
- **Exception Handling:** Handle poor quality images or missing metadata.

## 29. End-to-End Chatbot

- **Use Case Description:** Build a chatbot that handles user queries and interactions from start to finish using NLP.
- **User Goals:** Automate customer service and support.
- **Main Flow:** User asks a question → Chatbot processes the query → Generates and returns an appropriate response.

- **Alternate Flow:** User refines or rephrases the query for better results.
- **Exception Handling:** Handle ambiguous questions or out-of-scope queries.

### 30. YouTube Video Chaptering

- **Use Case Description:** Automatically generate chapters for YouTube videos using speech-to-text and NLP.
- **User Goals:** Improve video navigation and user engagement.
- **Main Flow:** User uploads video → System transcribes audio → Model identifies chapters → Chapters are displayed.
- **Alternate Flow:** User manually adjusts chapter timings.
- **Exception Handling:** Handle poor audio quality or incorrect transcription.

### 31. Text Generation Model

- **Use Case Description:** Use a language model to generate text based on a given prompt.
- **User Goals:** Aid content creation and brainstorming processes.
- **Main Flow:** User inputs a text prompt → Model generates a continuation → Text is displayed.
- **Alternate Flow:** User adjusts generation parameters like tone or length.
- **Exception Handling:** Handle cases where the generated text is nonsensical or repetitive.

### 32. Next Word Prediction Model

- **Use Case Description:** Predict the next word in a sequence based on preceding words using NLP models.
- **User Goals:** Enhance typing speed and accuracy in applications like text editors.
- **Main Flow:** User types a sequence → Model suggests the next word → User accepts or edits the suggestion.
- **Alternate Flow:** User toggles the prediction feature on/off.
- **Exception Handling:** Handle cases where context is too ambiguous for accurate predictions.

### 33. Generative AI Model From Scratch

- **Use Case Description:** Build a custom generative AI model for image or text generation.
- **User Goals:** Create content that is unique to a specific brand or purpose.
- **Main Flow:** User provides training data → Model learns from data → Generates new content.
- **Alternate Flow:** User adjusts model parameters for different generation styles.
- **Exception Handling:** Handle overfitting or insufficient training data.

### 34. Text Summarization Model with LLMs

- **Use Case Description:** Summarize lengthy documents into concise summaries using large language models.
- **User Goals:** Save time by quickly understanding the essence of long texts.
- **Main Flow:** User uploads a document → Model generates a summary → Summary is displayed.
- **Alternate Flow:** User adjusts the desired summary length.
- **Exception Handling:** Handle errors with extremely short or poorly structured input text.

### 35. Code Generation Model with LLMs

- **Use Case Description:** Generate code snippets based on user inputs like natural language descriptions.
- **User Goals:** Assist developers in automating repetitive coding tasks.
- **Main Flow:** User inputs code requirements → Model generates code snippet → User reviews the code.
- **Alternate Flow:** User adjusts input requirements for different code.
- **Exception Handling:** Handle cases where the generated code is syntactically correct but does not meet user needs.

### 36. Text Generation with LLMs

- **Use Case Description:** Generate coherent and contextually appropriate text using large language models.
- **User Goals:** Facilitate creative writing or automate responses in chatbots.
- **Main Flow:** User provides a prompt → Model generates a text response → User reviews the output.
- **Alternate Flow:** User refines prompts for different outputs.
- **Exception Handling:** Handle repetitive or off-topic generation.

### 37. Website Traffic Analysis & Forecasting

- **Use Case Description:** Predict website traffic patterns based on historical data.
- **User Goals:** Plan marketing strategies and server capacity.
- **Main Flow:** User uploads traffic data → Model predicts future traffic trends → Forecasts are displayed.
- **Alternate Flow:** User adjusts forecast horizon (e.g., daily, weekly).
- **Exception Handling:** Manage gaps in historical data or sudden traffic surges.

### 38. Ads CTR Forecasting

- **Use Case Description:** Predict the click-through rate (CTR) of ads over time using time-series analysis.
- **User Goals:** Optimize ad spending and improve conversion rates.
- **Main Flow:** User uploads ad performance data → Model forecasts future CTR → Forecasts are used for campaign adjustments.
- **Alternate Flow:** User adjusts the time range for forecasting.
- **Exception Handling:** Handle missing data or rapid changes in ad performance.

### 39. Multivariate Time Series Forecasting

- **Use Case Description:** Predict future trends using multiple time-dependent variables.
- **User Goals:** Improve accuracy in forecasts involving complex interdependencies.
- **Main Flow:** User uploads multivariate data → Model forecasts future trends → Results are displayed.
- **Alternate Flow:** User selects specific variables for forecasting.
- **Exception Handling:** Handle collinearity between variables or sudden shifts in trends.

### 40. Demand Forecasting & Inventory Optimization

- **Use Case Description:** Predict product demand and optimize inventory levels.
- **User Goals:** Reduce stockouts and overstock situations.
- **Main Flow:** User uploads historical sales data → Model predicts demand → System adjusts inventory levels.
- **Alternate Flow:** User sets custom safety stock levels.
- **Exception Handling:** Handle inaccurate predictions due to seasonal fluctuations.

### 41. Netflix Subscriptions Forecasting

- **Use Case Description:** Predict subscription trends for Netflix or similar streaming services.
- **User Goals:** Understand subscriber growth and retention patterns.
- **Main Flow:** User inputs historical subscription data → Model predicts future trends → Forecasts are displayed.
- **Alternate Flow:** User adjusts the time period or demographic segments.
- **Exception Handling:** Manage anomalies like sudden subscriber spikes due to new releases.

### 42. Currency Exchange Rate Forecasting

- **Use Case Description:** Predict future exchange rates using time-series models.



- **User Goals:** Aid in financial decision-making and risk management.
- **Main Flow:** User inputs historical exchange rate data → Model predicts future rates → Forecasts are displayed.
- **Alternate Flow:** User adjusts forecasting parameters.
- **Exception Handling:** Address market shocks or geopolitical events affecting currency.

#### 43. Instagram Reach Forecasting

- **Use Case Description:** Predict the reach of Instagram posts over time.
- **User Goals:** Optimize posting times and content strategies.
- **Main Flow:** User uploads engagement data → Model forecasts reach trends → User receives recommendations.
- **Alternate Flow:** User refines time ranges or audience segments.
- **Exception Handling:** Handle rapid shifts in platform algorithms affecting reach.

#### 44. Time Series Forecasting with ARIMA

- **Use Case Description:** Use ARIMA models for forecasting time series data like stock prices or weather patterns.
- **User Goals:** Gain accurate time series predictions for strategic planning.
- **Main Flow:** User inputs time-series data → Model forecasts future values → Forecasts are displayed.
- **Alternate Flow:** User adjusts ARIMA parameters (p, d, q).
- **Exception Handling:** Handle errors with non-stationary data or incorrect parameter selection.

#### 45. Weather Forecasting

- **Use Case Description:** Predict weather conditions using historical weather data.
- **User Goals:** Improve preparedness for adverse weather conditions.
- **Main Flow:** User inputs historical weather data → Model predicts future conditions → Forecasts are displayed.
- **Alternate Flow:** User selects specific weather parameters (e.g., temperature, rainfall).
- **Exception Handling:** Handle sudden weather anomalies or missing data.