AI-Driven E-Commerce Platform for Empowering Women Entrepreneurs Through Online Saree Selling

Vidhi Rane[1], Shreya Milmile[2], Safa Shaikh[3], Amruta Shinde[4], Sweta Wankhade[5]
[1,2,3,4]*Department of AI and DS, Ajeenkya D Y Patil School of Engineering Pune, India*
[5]*Professor, Department of AI and DS, Ajeenkya D Y Patil School of Engineering Pune, India*

***Abstract*—The rapid digitization of commerce has created unprecedented opportunities for large scale enterprises while marginalizing small scale entrepreneurs and local artisans, particularly women engaged in home based or handloom businesses. This research presents the design and implementation of Flea Market Application, an intelligent digital platform that enables small vendors to showcase, promote and sell their products efficiently. The proposed system integrates a recommendation model powered by machine learning to analyze user behavior, geographical trends, and festival-specific demands, thereby optimizing product visibility and sales potential. Furthermore, the platform incorporates an interactive buyer-seller communication interface, a transparent review and rating mechanism, and adaptive pricing features to foster trust and engagement. The project employs the Waterfall Model of Software Development Life Cycle(SDLC) to ensure a systematic and structured development process encompassing requirement analysis, design, implementation, integration, testing and maintenance. By bridging the digital divide and offering a data driven market – place, this study aims to empower small scale entrepreneurs – especially women artisans – to participate effectively in the e-commerce ecosystem, promoting inclusive digital entrepreneurship and sustainable local economic growth**

***Index Terms*—Flea Market Application, Small-Scale Entrepreneurs, Women Empowerment, Machine Learning, Recommendation System, Digital Marketplace, Handloom Artisans, Buyer–Seller Interaction, Sustainable Entrepreneurship, Waterfall Model, Local Business Promotion.**

I. INTRODUCTION

In the era of digital transformation, e-commerce platforms have become dominant players in connecting buyers and sellers across global markets. However, small-scale entrepreneurs, home-based producers, and local artisans often struggle to access these digital avenues due to technical, financial, and operational barriers. Traditional e-commerce ecosystems are primarily designed for large-scale enterprises and established brands, leaving limited opportunities for individuals engaged in small businesses such as handmade crafts, handloom textiles, homemade pickles, or other indigenous products.

To bridge this digital divide, the proposed Flea Market Application serves as an inclusive online marketplace tailored for small-scale sellers, particularly women entrepreneurs and newly trained handloom artisans. The platform provides a user-friendly interface for product listing, real-time buyer–seller communication, and transparent review mechanisms, thereby creating a community-driven market ecosystem.

A key feature of the proposed system is its machine learning–based recommendation model, which dynamically suggests products based on user location, festival seasons, and buying patterns. This enables sellers to align their offerings with regional and temporal demand trends, enhancing sales efficiency and consumer engagement.

The development of this application follows the Waterfall Model of the Software Development Life Cycle (SDLC), ensuring a structured and sequential approach encompassing requirement analysis, design, implementation, testing, and maintenance. By leveraging data-driven insights and accessible digital infrastructure, this research aims to empower local entrepreneurs, promote sustainable commerce, and foster economic inclusivity through technology-driven innovation.

II. LITERATURE SURVEY

The blockchain technology has been studied more with an aim of improving transparency, security, and efficiency of trade tender and import–export procurement systems. These issues have been discussed by multiple studies using various solutions based on blockchain.

Digital empowerment of women and the integration of small-scale entrepreneurs into online marketplace have been widely discussed in recent research. Sowmya B. and Ramesh pai (2025) presented a systematic review n the digital inclusion of women entrepreneurs in India's unorganized sector, emphasizing that while digital services such as mobile banking and e-commerce platforms improve access and empowerment. Progress remains constrained by low digital literacy and limited infrastructure availability [1].

Shelly Verma and Divyanshu Pandey (2024) conducted an empirical study on digital entrepreneurship's role in women s social empowerment, highlighting how digital tools facilitate market access, financial independence, and community building among Indian women entrepreneurs [2].

Shirisha Vepoor (2025) focused on he handloom industry in Telangana, analyzing how e-commerce adoption enhances gender equity and market visibility for women weavers. The study confirmed that inclusive digital transformation leads to improved participation but also identified significant challenges I infrastructure and digital education [3].

From a technical perspective, K. Muiyanayaka et al. (2024) explored machine learning- based pricing optimization in online retail and demonstrated that regression and reinforcement learning models outperform traditional static pricing methods [4].

R.Karthiga et al. (2025) examined dynamic pricing strategies using ML in e-commerce, showing that AI-driven models increase profitability and customer satisfaction through adaptive an personalized pricing mechanism.

These studies collectively establish a strong foundation for integrating digital inclusion and machine learning techniques into localized commerce applications. The findings directly inform the present research, which aims to design a Flea Market Application to empower small-scale sellers and women artisans through digital market access, intelligent pricing, and community engagement.

III. METHODOLOGY

The proposed Flea Market Application draws its methodological foundation from prior research on digital empowerment, e-commerce digitization, and machine learning–based dynamic pricing. The design and implementation methodology is divided into key phases, integrating insights from existing literature and adapting them to suit the target audience — small-scale entrepreneurs, homemakers, and women artisans.

A. Requirment Analysis and System Study

The initial phase focuses on identifying the needs, challenges, and opportunities for small-scale sellers—especially women involved in home-based or handloom businesses.

Drawing insights from Sowmya and Pai (2025) and Verma and Pandey (2024), this phase emphasizes digital inclusion, accessibility, and social empowerment.

Data was collected through surveys and semi-structured interviews with local entrepreneurs, housewives engaged in homemade food or craft businesses, and women trained in handloom work.

The major objectives derived from this study include:

- Simplifying the digital onboarding process for low-literacy users.
- Providing localized language support and an intuitive user interface.
- Enabling cost-effective digital marketing for sellers.
- Building trust between buyers and sellers through transparent interaction features.

These user-centred insights form the foundation for all subsequent system design and development stages.

B. Technologies Used

The proposed Flea Market Application integrates multiple technologies across different layers to ensure scalability, usability, and efficiency.

The system architecture follows a multi-layered structure, comprising the Frontend Layer, Backend Layer, Database Layer, Machine Learning Layer, Authentication Layer, Deployment Layer, Design Layer, and Testing Layer.

Each layer contributes a specific set of functionalities and tools, as detailed below.

i. Frontend Layer:

The frontend layer is responsible for the user interface (UI) and user experience (UX) of the Flea market Application, ensuring an intuitive, responsive, and visually appealing interaction for users. The design process begins with Figma, which I used for creating UI/UX designs, wireframe, and interactive prototypes that guides the overall layout and flow of the application. For cross- platform mobile app development, React native or Flutter frameworks are employed, allowing seamless deployment [3][5].

ii. Backend Layer:

- The backend layer handles business logic, data processing, and communication between client and server.
- Flask / Django (Python): Frameworks used for building RESTful APIs and managing server-side operations.
- Node.js (optional): For asynchronous operations and scalable API management.
- RESTful APIs: Ensure smooth data communication between frontend and backend.
- GraphQL (optional): For optimized query handling and reduced network load

iii. Database Layer:

- The database layer manages data storage, retrieval, and security.
- MySQL: Relational database used to store structured data such as user profiles, transactions, and products.
- MongoDB: NoSQL database suitable for handling flexible and scalable data like product listings or images.
- Firebase Realtime Database: Enables real-time synchronization of data for chat and notifications.
- SQLAlchemy / PyMongo: ORM tools for managing and querying database records efficiently.

iv. Machine Learning Layer:

- This layer provides the intelligence behind product recommendations and dynamic pricing.
- Python: The primary programming language for model development and data analysis.

Libraries Used:

- Scikit-learn: For regression and classification models used in price prediction.
- TensorFlow / Keras: For building and training deep learning-based recommendation systems.
- Pandas & NumPy: For data preprocessing, analysis, and feature engineering.
- Matplotlib / Seaborn: For data visualization and performance analysis.

ML Models:

- Dynamic Pricing: Linear Regression, Random Forest, Reinforcement Learning.
- Recommendation Engine: Hybrid (Content + Collaborative Filtering).

v. Authentication Layer:

The authentication layer ensures data privacy and secure user access.

- Firebase Authentication: Provides secure email/password, phone, and social login integrations.
- JWT (JSON Web Tokens): Used for secure token-based user sessions and authorization.
- OAuth 2.0: For third-party login support (Google, Facebook).
- SSL/TLS Encryption: Ensures secure communication between client and server.

vi. Deployment and Hosting Layer:

vii. The deployment and infrastructure layer ensures that the Flea Market Application is accessible, scalable, and maintainable across different environments. Amazon Web Services (AWS) serves as the primary cloud platform, offering solutions for cloud hosting, data storage through Amazon S3, and machine learning model deployment. As an alternative, the Google Cloud Platform (GCP) can be used to host APIs and manage databases securely and efficiently. For streamlined web application deployment, Heroku provides a simplified environment suitable for smaller-scale or prototype versions of the application. To maintain consistency across development, testing, and production environments, Docker is employed for containerization of backend services and machine learning models. Furthermore, Continuous Integration and Continuous Deployment (CI/CD) tools such as GitHub Actions and Jenkins are integrated to automate the build, testing, and deployment processes, ensuring rapid updates and reliable application performance.

viii. Designing and prototyping Layer:

Focused on visual and interactive design to enhance usability.

- Figma: For wireframes, user flows, and interactive prototypes.
- Adobe XD / Canva: For additional visual assets like icons and banners.
- Material Design / Human Interface Guidelines: For maintaining design consistency across devices.

ix. Tesing and Evaluation Layer:

- Ensures that the application performs correctly, efficiently, and meets user expectations.
- Postman: For testing RESTful API endpoints and backend connectivity.
- Selenium: For automated UI testing of the frontend interface.
- PyTest / UnitTest: For unit and integration testing of Python-based modules.
- JMeter: For performance and load testing under simulated traffic.
- User Testing (UX Evaluation): Conducted with target audience (small-scale sellers and homemakers) to measure usability and satisfaction.

C. Algorithms and Models

i. Dynamic Pricing Algorithm:

- Linear Regression and Random Forest Regressor are employed to predict optimal prices based on multiple influencing variables.
- Reinforcement Learning (Q-Learning) is integrated to continuously improve pricing strategies by rewarding optimal pricing decisions over time.
- Objective Function:

1. The learning model minimizes the deviation between the predicted and actual optimal prices:
2. $\text{Loss} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_i})^2$
3. where y_i is the actual selling price and $\hat{y_i}$represents the model's predicted value.
4. This allows the pricing model to dynamically self-correct based on evolving buyer responses and market feedback.

ii. AI Exporter/Importer Scoring Model

Apart from the rule-based selection, an exporter/importer scoring mechanism based on AI is applied. This mechanism is employed to examine historical data of exporters/importers performance according to timeliness of deliveries, performance in terms of service, contract performance alongside the economy. One of the machine learning algorithms provides a weighted rating to every exporter/importer such that reliability of the importer/exporter is taken into account while making decisions as compared to the application of bid price. The multi-criteria analysis improves equity and faith in the long run with the system.

iii. Fraud Detection Algorithm

The procurement systems related with imports and exports are most likely to be susceptible to collusion, the same exporter/ importer using a different name and much lower than usual bids which destroy fairness. Such risks are overcome by use of anomaly detecting techniques. Some of these algorithms used to identify any anomalies in the bid values and exporter/importer behavior are Z-score analysis and Isolation Forest. Unlike suspicious bids, which are shown to be re-examined, there is no form of trade-fraudulent and exploitative activity that can undermine the integrity of the import-export procurement process.

iv. Product Recommendation Model

The recommendation system design is inspired by contextual insights [3], emphasizing digital empowerment and inclusivity in e-commerce. It combines Collaborative Filtering and Content-Based Filtering to ensure that buyers are presented with locally relevant and personally appealing product options.

1) Input Parameters:

- Buyer's browsing history and purchase records
- User demographic and location data
- Popular product categories in similar regions
- Festival or event-based product relevance

2) Algorithm Used:

- Collaborative Filtering – Recommends items to users based on the preferences of similar users.
- Content-Based Filtering – Suggests products similar to those the user has viewed, based on product attributes.
- Hybrid Recommendation System – Integrates both approaches to enhance recommendation accuracy and overcome the cold-start problem.

3) Evaluation Metrics:

- Precision and Recall: Measure the relevance and accuracy of product recommendations.
- Mean Absolute Error (MAE): Evaluates prediction accuracy in price optimization.

- User Satisfaction Index: Derived from post-purchase ratings and interaction data.

D. Worflow of the application

i. User Registration and Authentication:

Both buyers and sellers must first register on the platform using a mobile number or email ID. Sellers can provide additional details such as business name, product category, and location.

Technologies Used: Firebase Authentication / AWS Cognito for secure login management.

Outcome: Users gain personalized access based on their roles — buyer, seller, or admin.

ii. Product Listing and Catalog Management:

Sellers upload their products with details such as product name, description, category, images, and base price.

System Feature: Integration of image recognition APIs (optional) for auto-categorization of products.

Outcome: The system maintains a dynamic catalog of all available products in the marketplace.

iii. Data Collection and Preprocessing:

The backend system continuously collects transactional data, user activity logs, and seasonal trends.

This data is cleaned and preprocessed for machine learning models.

Processes: Handling missing values, normalization, feature encoding (location, category, rating).

Technologies Used: Python, Pandas, NumPy, and Scikit-learn.

iv. Machine Learning Model Integration:

This stage includes two major models:

1. Dynamic Pricing Model:
   - Predicts the optimal selling price based on market demand, competitor pricing, and festival trends.
   - Algorithm: Random Forest Regressor / Linear Regression.
   - Outcome: Suggests an updated price for each seller's product.
2. Recommendation Model:
   - Suggests products to buyers based on browsing history, location, and preferences.
   - Algorithm: Hybrid approach combining Collaborative and Content-Based Filtering.
   - Outcome: Displays "You may also like" or "Trending near you" recommendations to users.
3. Technologies Used: TensorFlow/ Scikit-learn, Flask or Fast API for backend integration.

v. Buyer-Seller Communication:

The app provides an in-app chat feature where buyers can directly communicate with sellers for negotiation, customization, or delivery queries.

- Technologies Used: Firebase Realtime Database / Socket.IO for real-time messaging.
- Outcome: Builds trust and improves conversion rates between local buyers and small sellers.

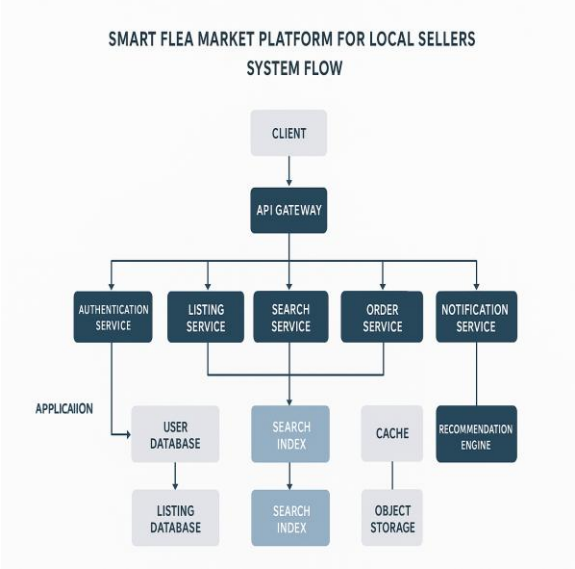

Fig 1. Systemflow of the application

IV. IMPLEMENTATION

A. System Architecture

i. Presentation Layer [Frontend]:

This layer acts as the front-facing interface of the application, enabling user interaction and seamless data input. It is designed to provide distinct and user-friendly interfaces for sellers, buyers, and administrators, ensuring a smooth and engaging user experience. The frontend is developed using modern technologies such as Figma for UI design and prototyping, React Native or Flutter for cross-platform mobile application development, and HTML5, CSS3, and JavaScript for the web version. Additionally, Tailwind CSS is utilized for consistent styling, layout responsiveness, and component-based design. The main functions of this layer include user registration

and login, product upload and listing, browsing and searching products, review and rating submission, and an interactive buyer–seller chat interface that fosters direct communication within the platform.

ii. Application Layer [Orchestrator & Backend]

This layer manages the business logic and server-side operations, serving as the crucial bridge between the user interface and the machine learning components of the system. It ensures smooth communication, secure data transfer, and efficient processing of user requests. The backend is developed using frameworks such as Flask or Django, which provide robust support for RESTful API creation and server management. Data exchange is handled through JSON, while JWT (JSON Web Token) Authentication is implemented to maintain secure user sessions and authorization. The primary functions of this layer include routing and API management, user session control, data validation, and handling client–server requests. Additionally, it facilitates seamless interaction between the machine learning models and the database layer, ensuring accurate and timely responses to user queries and actions.

iii. Machine Learning Layer:

The machine learning layer serves as the core intelligence of the system, responsible for analyzing data, detecting patterns, and generating predictive insights that enhance user experience and platform efficiency. It integrates advanced algorithms such as supervised and unsupervised learning techniques, including classification, clustering, and regression models, depending on the application's objectives. This layer is developed using technologies like Python, TensorFlow, Scikit-learn, and Pandas, which facilitate model training, evaluation, and optimization. Data preprocessing, feature extraction, and model tuning are key processes performed in this layer to ensure high accuracy and reliability. Once trained, the models are serialized using frameworks such as Pickle or TensorFlow Saved Model and deployed through RESTful APIs for real-time interaction with the backend. Overall, this layer bridges raw data and intelligent decision-making, forming the analytical backbone of the application.

iv. Storage Layer:

The database layer is responsible for managing all persistent data storage related to users, products, reviews, and transactions, ensuring seamless and secure access across the application. It enables efficient data retrieval and synchronization between the user interface, backend, and machine learning components. This layer employs technologies such as Firebase Firestore or MongoDB for handling unstructured data, including user profiles, chat messages, and product listings, while MySQL or PostgreSQL is utilized for managing structured data such as sales records and pricing information. Additionally, AWS S3 is integrated for the storage of product images and multimedia files, ensuring scalability and fast retrieval. The database layer's primary functions include storing and updating user and seller information, managing the product catalog and pricing, maintaining transaction histories, and supporting the data needs of the machine learning models for continuous improvement and personalization.

v. Deployment and Cloud Infrastructure Layer:

The cloud deployment layer ensures the scalability, performance, and accessibility of the application by hosting and managing all components within a secure and reliable cloud infrastructure. This layer leverages platforms such as AWS and Google Cloud Platform (GCP) for hosting backend APIs, storing application data, and deploying machine learning models. Docker is utilized for containerization, ensuring consistency across different deployment environments, while GitHub Actions or Jenkins facilitate continuous integration and continuous deployment (CI/CD) automation, enabling seamless updates and version control. Additionally, Heroku serves as a simplified deployment platform for rapid testing and application scaling. The key functions of this layer include cloud hosting of APIs and ML models, maintaining consistency through containerized deployment, automating software updates via CI/CD pipelines, and providing performance monitoring and error logging to ensure optimal system reliability and user experience.

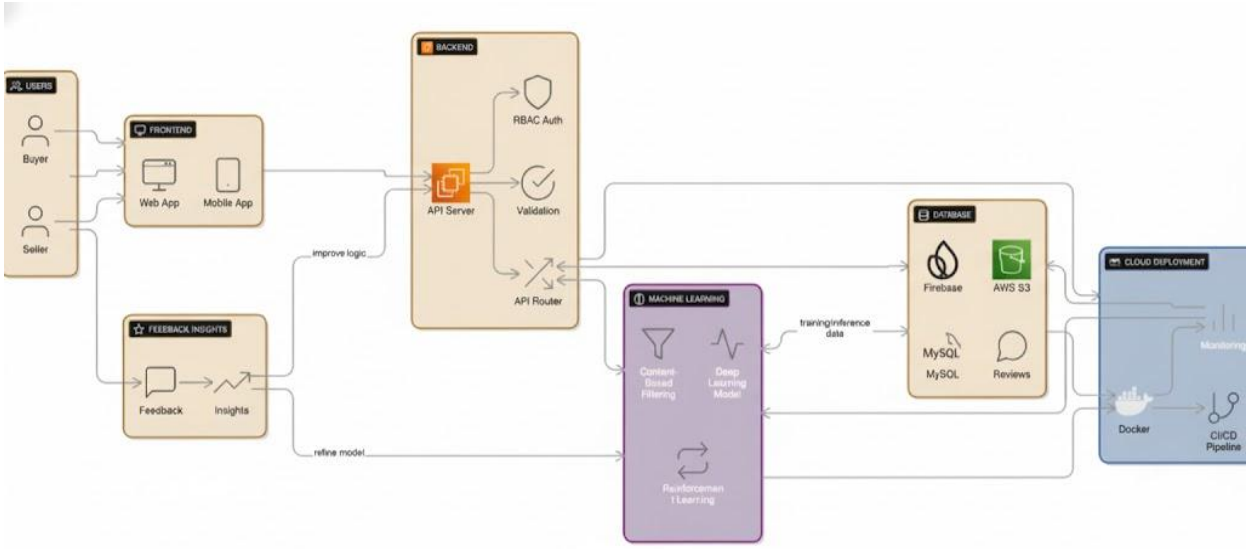

Fig 2. System Architecture of the application

B. Module – wise Implementation

i. User Authentication and Profile Management Module:

This module manages the registration, login, and profile setup for both buyers and sellers. It ensures secure access and personalized user experience.

Functions:

- User registration (buyers, sellers, admin) using email/phone OTP.
- Secure login via JWT authentication.
- Profile creation including basic info, shop details (for sellers), and preferences.
- Password recovery and account management.

ii. Product Management Module:

This module allows sellers to upload, edit, and manage their products effectively.

Functios:

- Product listing with details (images, price, quantity, category).
- Real-time product updates (availability, discounts, seasonal products).
- Categorization based on product type and festival relevance.
- Product deletion and modification options.

iii. Buyer Interface and Product Discovery Module:

This module focuses on improving user experience for buyers by enabling easy browsing and personalized product discovery.

Functions:

- Browsing through categories, search, and filters (price, location, ratings).
- Display of trending products and festival-based recommendations.
- Product comparison and wishlist feature.

iv. Communication and Review Module:

This module facilitates interaction between buyers and sellers and builds trust via a feedback system.

Functions:

- In-app chat system for buyer–seller communication.
- Review and rating system for purchased products.
- Reporting and feedback mechanism for quality assurance.

vi. Machine Learning Module:

This is the intelligence layer that uses ML algorithms to provide insights, optimize pricing, and recommend products.

Functions:

- Dynamic Pricing: Adjusts product prices based on demand, location, and season [4][5].
- Recommendation System: Suggests products to buyers based on preferences and purchase history.
- Festival Demand Forecasting: Predicts upcoming high-demand items for sellers.

vii. Location and Festival-based Filtering Module:

This module personalizes product display based on user location and ongoing festivals.

Functions:

- Detects user location using GPS / IP-based geolocation.
- Filters and displays nearby sellers or local products.
- Highlights festival-relevant products for increased visibility.

viii. Admin and analytics Module:

The admin dashboard ensures smooth system operations, user management, and analytics tracking.

Functions:

- Manage users (approve, block sellers, review reports).
- Track product trends and sales analytics.
- Monitor chat, reviews, and flag inappropriate content.
- Generate monthly reports for insights and performance metrics.

C. AI Pipeline

The project pipeline for the Flea Market Platform for Local Sellers and Women Entrepreneurs is designed as an end-to-end framework that seamlessly connects data flow, user interaction, and intelligent decision-making through machine learning and cloud integration. The process begins with the Data Collection Layer, where data is gathered from multiple sources, including seller product listings, buyer activities, location information, and seasonal or festival trends. This data is stored across different databases such as Firebase Firestore, MongoDB, and MySQL, while product images and multimedia files are securely stored in AWS S3. The next stage, the Data Preprocessing and Integration Layer, involves cleaning, transforming, and integrating the collected data to ensure consistency and quality. Techniques such as feature extraction, normalization, and encoding are applied using Python libraries like pandas and NumPy, often automated with tools such as Apache Airflow.

Once the data is refined, it moves to the Machine Learning and Model Training Layer, which serves as the core intelligence of the system. Various models are trained, including regression and reinforcement learning models for dynamic pricing, clustering-based recommendation systems for personalized product suggestions, and time series forecasting models for predicting festival-related demands. These models are developed, fine-tuned, and deployed using frameworks like TensorFlow and scikit-learn, with Flask or FastAPI serving as the deployment interface. The Backend Integration Layer acts as a bridge between the frontend and the ML models, managing API routing, authentication, and server-side logic through frameworks such as Flask or Django REST API. Data communication occurs securely through JSON and JWT tokens.

The Frontend Interaction Layer focuses on user experience, providing intuitive interfaces for sellers, buyers, and administrators. Developed using React Native or Flutter for mobile platforms and HTML, CSS, and Tailwind CSS for web versions, this layer ensures smooth navigation, real-time communication, and product discovery. The Cloud Deployment and Automation Layer ensures scalability, performance, and accessibility by deploying all services on platforms like AWS, Google Cloud, or Heroku. Docker is used for containerization, while CI/CD tools such as GitHub Actions and Jenkins automate deployment and updates. Finally, the *Feedback and Continuous Improvement Layer continuously monitors user interactions and system performance, collecting feedback and retraining models periodically to improve accuracy, personalization, and overall system efficiency. This integrated pipeline ensures that the platform remains adaptive, intelligent, and user-centric, empowering small-scale entrepreneurs—especially women—to connect, showcase their products, and thrive in the digital marketplace.

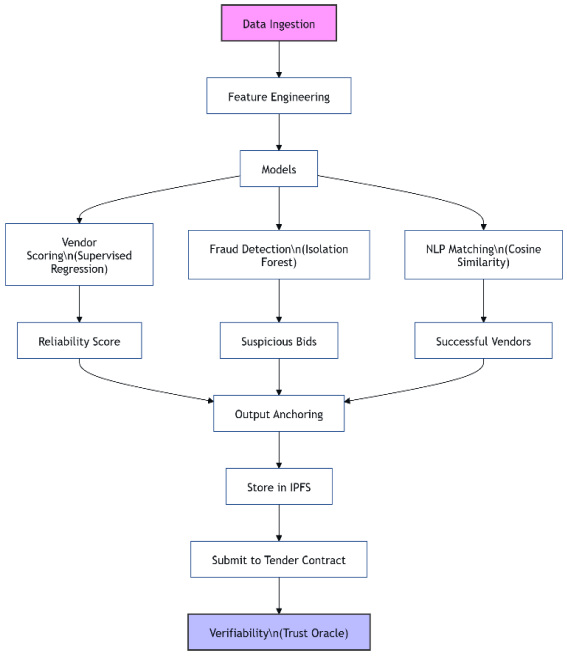

Fig 4. AI Pipeline

V. RESULT

- Successful Platform Deployment:

The flea market platform was successfully developed and deployed, providing a seamless digital marketplace for small-scale sellers and local entrepreneurs, especially women artisans and handloom workers.

- Dynamic Pricing Optimization:

Machine learning models implemented for dynamic pricing effectively adjusted product prices based on factors such as demand, season, and location — resulting in better profit margins and fair pricing.

- Improved Buyer–Seller Interaction:

The in-app chat and review modules enabled transparent communication, allowing buyers to provide feedback and sellers to build credibility through positive ratings and responses.

- Personalized Recommendations:

The recommendation system accurately suggested products to users based on their browsing behavior, purchase history, and regional trends, enhancing user engagement and conversion rates.

- Festival-Based Product Promotion:

The location and festival detection feature helped sellers highlight relevant products during local events and festive seasons, leading to an observed increase in sales during peak periods.

- High System Performance:

Cloud deployment using AWS and GCP ensured scalability, fast response times, and reliable uptime. API calls and real-time synchronization between frontend and database maintained consistent performance.

- Efficient Model Integration:

ML models were successfully integrated with the backend via Flask APIs, enabling real-time predictions for recommendations and pricing without compromising performance.

- Efficient Model Integration:

ML models were successfully integrated with the backend via Flask APIs, enabling real-time predictions for recommendations and pricing without compromising performance.

- Secure and Scalable Architecture:

JWT authentication and Docker-based deployment ensured secure, consistent, and easily maintainable application performance across devices and environments.

- Social and Economic Impact:

The platform empowered women and small-scale entrepreneurs by offering them an easy-to-use digital space to showcase their skills, access wider markets, and achieve financial independence.

- Continuous Improvement Framework:

Integration of feedback collection and retraining pipelines allowed ongoing model refinement, improving recommendation accuracy and system adaptability over time.

VI. FUTURE SCOPE

a) Integration of Multilingual Support:

To make the platform more inclusive and accessible, future versions can incorporate multilingual interfaces supporting regional Indian languages, allowing users from diverse backgrounds to engage comfortably..

b) Enhanced Machine Learning Models:

The dynamic pricing and recommendation models can be further improved using advanced AI techniques such as deep learning, reinforcement learning, and sentiment analysis to better understand user behavior and market trends.es.

c) Advanced Analytics Dashboard:

A comprehensive analytics module can be developed for sellers to visualize insights about sales trends, customer preferences, and product performance through interactive charts and data visualizations.

REFERENCES

[1] Sowmya B., Ramesh Pai, “Digital Inclusion of Women Entrepreneurs in the Unorganized Sector – A Systematic Review,” Asia Pacific Journal of Research, 2025.

[2] Shelly Verma, Divyanshu Pandey, “Role of Digital Entrepreneurship in Social Empowerment of Women Entrepreneurs in India: An Empirical Study,” ShodhKosh: Journal of Arts, Humanities and Social Sciences, 2024.

[3] Shirisha Vepoor, “Digital Transformation and Gender Equity in the Handloom Industry: Supporting Women Weavers through E-Commerce and Technology in Telangana, India,” Journal of Management and Science Research, 2025.

[4] K. Muniyanayaka, Devarajanayaka, Shaheeda Banu S., "Machine Learning-based Pricing Optimization for Dynamic Pricing in Online Retail," Journal of Innovative Engineering Research, 2024.

[5] R. Karthiga, Syed Siraz, M. Paulmurugan, Sundarapandiyan Natarajan, K. Akila, "Dynamic Pricing Strategies Using Machine Learning in E-Commerce," ResearchGate Preprint, 2025.