VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, BELAGAVI

Project Report on

IOT BASED AQUACULTURE MONITORING SYSYTEM

In partial fulfillment of the requirements for the award of the Degree of

Bachelor of Engineering

In

Computer Science and Engineering

By

POOJA C	1AH20CS072
RACHITHA C	1AH20CS079
RISHMA SAI	1AH20CS080
SWAPNA K	1AH20CS107

Under the Guidance of

Mrs. Sandhyarani H G

Assistant Professor, Dept. of CSE

ACSCE, Bangalore

Department of Computer Science and Engineering

2023-2024

ACS COLLEGE OF ENGINEERING
KAMBIPURA, MYSORE ROAD, BANGLORE-74

Department of Computer Science and Engineering

Certificate

This is to certify that the Project Work entitled "IOT BASED AQUACULTURE MONITORING SYSYTEM" is a bonafide work carried out by Ms POOJA C (1AH20CS072), Ms RACHITHA C (1AH20CS079), Ms RISHMA SAI (1AH20CS080), Ms SWAPNA K (1AH20CS107), in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering, of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI during the year 2023-2024. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The Project Report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said degree

Signature of Guide	Signature of HOD	Signature of Principal
Mrs. Sandhyarani H G	Dr. Senthil Kumaran T	Dr. Anandthirtha B Gudi

S No	Name of the Examiners	Signature with Date
1.	_____	_____
2.	_____	_____

DECLARATION

We, Ms. POOJA C (1AH20CS072), Ms. RACHITHA C (1AH20CS079), Ms. RISHMA SAI (1AH20CS080) and M s . SWAPNA K (1AH20CS107), hereby declare that the project work entitled "IOT BASED AQUACULTURE MONITORING SYSTEM" has been independently carried out by us under the guidance of Mrs. Sandhyarani H G, Assistant Professor, Department of Computer Science & Engineering, ACS College of Engineering, Bangalore, in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi. We further declare that we have not submitted this report either in part of in full to any other university for the reward of any degree.

POOJA C	1AH20CS072
RACHITHA	1AH20CS079
RISHMA SAI	1AH20CS080
SWAPNA K	1AH20CS107

Place: Bangalore Date:

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to the ACS College of Engineering, Bengaluru for providing me an opportunity to carry out my project report.

I express my deep regards to our honorable Chairman Dr. A C SHANMUGAM, for providing me an opportunity to fulfil my ambition in this prestige institute.

I would like to express my immense gratitude to Dr. ANANDTHIRTHA B GUDI, Principal, ACS College of Engineering, Bengaluru, for his timely help and inspiration during the tenure of the course.

I express my sincere regards and thanks To Dr. T SENTHIL KUMARAN, Professor and HOD, Computer Science and Engineering, ACSCE, Bengaluru for the encouragement and support throughout encouragement the work.

With profound sense of gratitude, I acknowledge the guidance, support and encouragement to my guide Mrs. SANDHYARANI H G, Assistant Professor, Dept. of Computer Science and Engineering, ACSCE, Bengaluru.

I hereby like to thank our Project Coordinator Dr. ASHOK KUMAR, Professor, Dept. of Computer Science and Engineering, for the encouragement and support.

We also thank all the Teaching and Non-teaching staff of Computer Science and Engineering Department, who have helped us to complete the project in time.

Project Team members:

		POOJA C	1AH20CS072
RACHITHA C	1AH20CS079	R RISHMA SAI	1AH20CS080
		SWAPNA K	1AH20CS107

ABSTRACT

Movie recommendation systems have transformed the entertainment industry by providing users with personalized movie suggestions. Traditional approaches, such as collaborative filtering and content-based filtering, face challenges like the cold start problem, data sparsity, and bias towards popular content. This project explores a hybrid recommendation system leveraging machine learning and deep learning techniques to overcome these limitations. The system analyses user preferences, ratings, and content features to generate personalized recommendations.

A combination of techniques, including content-based filtering, collaborative filtering, and deep learning models such as neural networks, is implemented to enhance recommendation accuracy. Additionally, feature extraction from movie metadata, genre classification, and sentiment analysis from reviews are incorporated to improve prediction performance. The system also explores multimodal learning by integrating visual and speech features to classify movie genres and improve recommendations. This research aims to enhance user experience, reduce bias, and ensure computational efficiency.

Keywords: Movie Recommendation, Machine Learning, Deep Learning, Content-Based Filtering, Collaborative Filtering, Hybrid Model, Sentiment Analysis, Personalization.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Movie recommendation systems have become an essential part of the entertainment industry. With the increasing number of movies released each year, users often struggle to find films that match their preferences. Our project aims to develop a movie recommender system using the TMDB dataset, leveraging machine learning and deep learning techniques to suggest personalized movie recommendations.

Recommender systems are used extensively in various domains such as e-commerce, music streaming, and video streaming services. The primary goal of a movie recommender system is to analyze user preferences, past interactions, and movie content to provide relevant suggestions. Popular streaming services like Netflix, Amazon Prime, and Disney+ heavily rely on such systems to enhance user engagement and increase retention.

The movie recommendation system proposed in this project utilizes a hybrid approach, combining content-based filtering, collaborative filtering, and deep learning models. The system takes into account factors such as movie genres, actors, directors, user ratings, and reviews to generate meaningful recommendations. By integrating various filtering techniques, our model aims to provide better accuracy and a more personalized experience for users.

1.2 Areas of Project

- Machine Learning and Deep Learning**:** The project employs advanced ML algorithms and neural networks to enhance the recommendation quality.
- Natural Language Processing (NLP)**:** User reviews and descriptions are processed using NLP techniques to extract sentiments and additional features.

- Data Analytics and Visualization: The system involves analyzing and visualizing user interactions and movie metadata to derive meaningful insights.
- Web Development (if deployed as a web-based system)**:** The system can be implemented as a web-based application with an intuitive UI to enhance user accessibility.

1.3 Challenges and Complexities

- Cold Start Problem**:** The system may face difficulties in recommending movies to new users with no prior interactions.
- Data Sparsity**:** A large portion of users may have rated only a few movies, making it challenging to generate accurate predictions.
- Bias Toward Popular Movies**:** Many recommendation algorithms tend to favor widely popular movies, neglecting lesser-known gems.
- Computational Complexity**:** Processing a large-scale dataset with deep learning models requires significant computational power.

1.4 Movtivation

With the rise of streaming platforms like Netflix and Amazon Prime, personalized movie recommendations have become a necessity. This project aims to improve the accuracy and effectiveness of recommendation systems by using hybrid models that combine content-based filtering, collaborative filtering, and deep learning techniques.

Traditional recommendation techniques often fail to capture user preferences comprehensively, leading to irrelevant suggestions. By leveraging modern AI techniques, this project seeks to enhance user engagement, improve recommendation diversity, and minimize redundancy in suggestions. Personalized recommendations not only help users discover new content but also increase overall satisfaction and platform retention rates.

1.5 Objectives

- To develop a personalized movie recommendation system.
- To integrate machine learning techniques for better recommendation accuracy.
- To analyze and visualize user preferences.
- To enhance the user experience by minimizing irrelevant suggestions.
- To implement a scalable system capable of handling large amounts of movie data.
- To improve diversity in recommendations, avoiding over-reliance on popular movies.

1.6 Problem Statement

Developing an effective recommendation system is challenging due to issues like data sparsity, cold start problems, and bias in existing models. Our system will address these challenges using a hybrid approach combining multiple recommendation techniques.

Users often struggle to navigate the vast collection of available movies on streaming platforms. Traditional search mechanisms are inefficient in helping users discover content that aligns with their preferences. By developing an intelligent recommender system, we aim to enhance user experience by automatically suggesting movies that match their tastes, viewing history, and engagement patterns.

1.7 Description of Dataset

- Dataset Source: The Movie Database (TMDB)
- Features Included: Movie title, genres, ratings, user reviews, cast, crew, and metadata.
- Data Size: Over 50,000 movies and user interactions.
- Key Attributes:
    - Movie ID

    - Title
    - Genre
    - Cast & Crew
    - User Ratings
    - Reviews and Sentiments
- Data Preprocessing:
    - Handling missing values
    - Normalization of user ratings
    - Feature extraction from movie metadata
    - Text processing for reviews

1.8 Applications

- Streaming platforms (Netflix, Hulu, Amazon Prime, Disney+)**:** Personalized movie recommendations improve user engagement.
- Online movie rental services**:** Suggesting relevant movies based on rental history.
- E-commerce platforms recommending movies**:** Platforms like Google Play Movies and Apple iTunes use such systems to boost sales.
- Educational platforms**:** Recommending educational or documentary films based on learning preferences.

1.9 Summary

This chapter introduced the need for a movie recommendation system, the challenges it addresses, and an overview of the TMDB dataset. The project's objectives, problem statement, and areas of application were also discussed, highlighting the significance of personalized recommendations in modern digital platforms.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Survey 1

LITERATURE SURVEY

Title: Movie Recommendation Systems Using Actor-Based Matrix Computations in South Korea

Authors: Syjung Hwang, Eunil Park

Publication: IEEE – 2022

Description:

This study proposes an actor-based recommendation model that enhances traditional movie recommendation techniques by leveraging actor-filmography data. Instead of relying on collaborative filtering or genre-based approaches, the system computes rank correlation between actors and movie genres to provide more personalized recommendations. The research utilizes datasets from Korean Film Council, Naver Movie, and Korea Box Office Information System, analyzing over 4,450 actors and 509 movies. A survey-based evaluation demonstrates that users who prefer actor-based recommendations find this system more effective than traditional genre-based approaches.

Advantages:

Improves personalization by considering actor preferences rather than just movie genres.

Effectively handles the cold start problem by associating new users with actors they like.

Uses real audience preferences from box office and online platforms.

Can be integrated with hybrid models for improved accuracy.

Limitations:

Does not account for directors, cinematography, or storyline preferences.

Limited dataset scope (only South Korean movies from 2010–2019).

Requires large-scale actor-filmography data processing, increasing computational cost.

Ignores social media trends and peer influence in movie recommendations.

Survey 2

LITERATURE SURVEY

Title: CollRec: Pre-Trained Language Models and Knowledge Graphs Collaborate to Enhance Conversational Recommendation Systems

Authors: Shuang Liu, Zhizhuo Ao, Peng Chen, Simon Kolmanić

Publication: 2024

Description:

This paper introduces CollRec, a conversational recommendation system that integrates pre-trained language models (PLMs) with knowledge graphs (KGs) to improve movie recommendations. The system first extracts knowledge graphs from user conversations using T5-based transformers, then enhances recommendation accuracy by linking movie entities. The model is trained on datasets like ReDial, WebNLG 2020, and Reddit-Movie, using evaluation metrics such as precision, recall, BLEU, and Rouge scores. Results show that the CollRec model outperforms traditional text-based recommender systems.

Advantages:

Improves recommendation accuracy using knowledge graph-enhanced PLMs.

Understands user conversations better, making recommendations more interactive.

Produces fluent, AI-generated responses, enhancing user experience.

Scalable across multiple domains, not just movies.

Limitations:

Computationally expensive, requiring high-end GPUs for real-time processing.

Dependent on knowledge graphs, which need constant updates for accuracy.

Cold start problem for less frequently mentioned movies.

Survey 3

LITERATURE SURVEY

Title: Exploiting Aesthetic Features in Visual Contents for Movie Recommendation

Authors: Xiaojie Chen, Pengpeng Zhao, Yanchi Liu, Lei Zhao, Junhua Fang, Victor S. Sheng, Zhiming Cui

Publication: 2019

Description:

This paper presents a visual content-based movie recommendation system that incorporates aesthetic features from movie posters and still frames. The model, called Aesthetic-aware Unified Visual Content Matrix Factorization (UVMF-AES), utilizes Convolutional Neural Networks (CNNs) such as VGG16 and OWACNN to extract spatial and aesthetic features. By combining content-based filtering with deep learning techniques, the system enhances the recommendation accuracy for visually rich movies. The research is conducted on MovieLens2011 and IMDb datasets, evaluating performance with RMSE, Precision@N, and Recall@N.

Advantages:

Higher recommendation accuracy using deep learning-based aesthetic analysis.

Reduces the data sparsity issue by leveraging visual content instead of ratings.

Uses CNNs (VGG16, OWACNN) to improve feature extraction.

Limitations:

Computationally expensive, requiring deep learning models and high-end hardware.

Limited to visual features, ignoring plot, storyline, and user reviews.

Cold start problem for movies that lack rich visual data (e.g., documentaries).

2.2 Background Study

Recommendation systems can be broadly categorized into three types:

- Content-Based Filtering: This method suggests movies based on metadata such as genre, director, and actors. It uses techniques like TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity to compare movies based on textual descriptions.
- Collaborative Filtering: This approach suggests movies based on user interactions, preferences, and behaviors. It relies on techniques such as user-

based filtering and item-based filtering. A well-known method in collaborative filtering is matrix factorization, which includes Singular Value Decomposition (SVD) and Alternating Least Squares (ALS).

- Hybrid Models**:** Hybrid recommendation systems combine multiple techniques to improve accuracy and personalization.

2.3 Existing Model

- **Memory-Based Collaborative Filtering:** This approach utilizes a similarity matrix to recommend items based on the nearest neighbors of a user or item. It can be further divided into **user-based** and **item-based** filtering. User-based filtering recommends movies based on the preferences of similar users, whereas item-based filtering suggests movies that are similar to those previously rated highly by the user.
- **Model-Based Collaborative Filtering:** This technique applies machine learning models to predict user preferences. Matrix factorization techniques such as Singular Value Decomposition (SVD) and Non-Negative Matrix Factorization (NMF) are commonly used. More advanced models employ deep learning architectures such as autoencoders and recurrent neural networks (RNNs) to improve accuracy.
- **Deep Learning-Based Approaches:** Deep learning has revolutionized recommendation systems by allowing the extraction of complex patterns from large datasets. Convolutional Neural Networks (CNNs) are used for image-based recommendations, while transformers and attention mechanisms have improved the ability to understand sequential interactions in user behavior.

2.4 Summary

The literature survey on movie recommendation systems explores various advanced techniques to enhance recommendation accuracy and user experience. One study introduces an actor-based matrix computation approach, leveraging actor preferences and historical roles to improve recommendations, effectively addressing the cold-start

problem but requiring high computational power. Another research focuses on conversational AI with knowledge graphs (CollRec), enabling interactive, real-time movie suggestions by integrating natural language processing and external knowledge bases, though it demands frequent updates and computational resources. The third study proposes an aesthetic feature-based model (UVMF-AES), utilizing deep learning to extract visual features from movie posters and trailers, improving recommendations for visually appealing films but lacking storyline-based insights.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 INTRODUCTION

Requirement analysis is a crucial step in the development of any system. It helps define the scope, technical constraints, and essential components required for a successful implementation. This chapter focuses on understanding the necessary hardware and software components, functional and non-functional requirements, and performance expectations of the movie recommendation system.

3.2 SYSTEM REQUIREMENTS

3.2.1 Hardware Requirements

SI.NO	Components	Specification
1	Processor	Intel i5/i7 or AMD equivalent with multiple cores
2	RAM	Minimum 8GB (16GB recommended for deep learning)
3	Storage	At least 50GB free space
4	GPU	NVIDIA GPU with CUDA support for deep learning
5	Network	Stable internet connection for API requests

Table 3.2.1 Hardware Requirements

3.2.2 Software Requirements

Sl. No.	Components	Specification
1	Programming Language	Python 3, micro Python.
2	Development Environment	memory model and peripheral access
3	Database	PostgreSQL, MySQL, or MongoDB
4	Machine Learning Libraries	Scikit-learn, Surprise Library
5	Deep Learning Frameworks	TensorFlow, Keras
6	Data Processing Libraries	Pandas, NumPy

Table 3.2.2 Software Requirements

3.3 Summary

This chapter outlined the essential system requirements, both hardware and software, for developing the movie recommendation system. Additionally, it detailed the functional and non-functional requirements that the system must fulfill to ensure efficiency, accuracy, and a seamless user experience. In the next chapter, we will discuss the proposed system architecture and its various components.

CHAPTER 4
PROPOSED SYSTEM

4.1 Introduction

With the evolution of data-driven decision-making, traditional recommendation systems have evolved to incorporate more sophisticated techniques. The proposed movie recommendation system aims to enhance accuracy and personalization by leveraging hybrid models that combine content-based filtering, collaborative filtering, and deep learning techniques. This system integrates user preferences, movie metadata, and contextual factors to generate better recommendations.

The hybrid approach helps overcome common challenges such as the cold start problem, data sparsity, and popularity bias. By employing machine learning algorithms, deep learning models, and natural language processing (NLP), our system delivers highly personalized and accurate recommendations.

4.2 Proposed Model

The proposed system is designed to generate personalized movie recommendations by integrating multiple recommendation techniques. The system consists of the following major components:

- Content-Based Filtering**:** Recommends movies based on similarities in metadata such as genre, cast, and director.
- Collaborative Filtering: Suggests movies based on user behavior, analyzing patterns in ratings and interactions.
- Hybrid Approach: Combines content-based and collaborative filtering to enhance recommendation accuracy.
- Deep Learning Models: Utilizes neural networks for sentiment analysis on user reviews, improving personalized recommendations.
- Web Interface: A user-friendly interface where users can log in, rate movies, and receive recommendations.

4.2.1 System Architecture

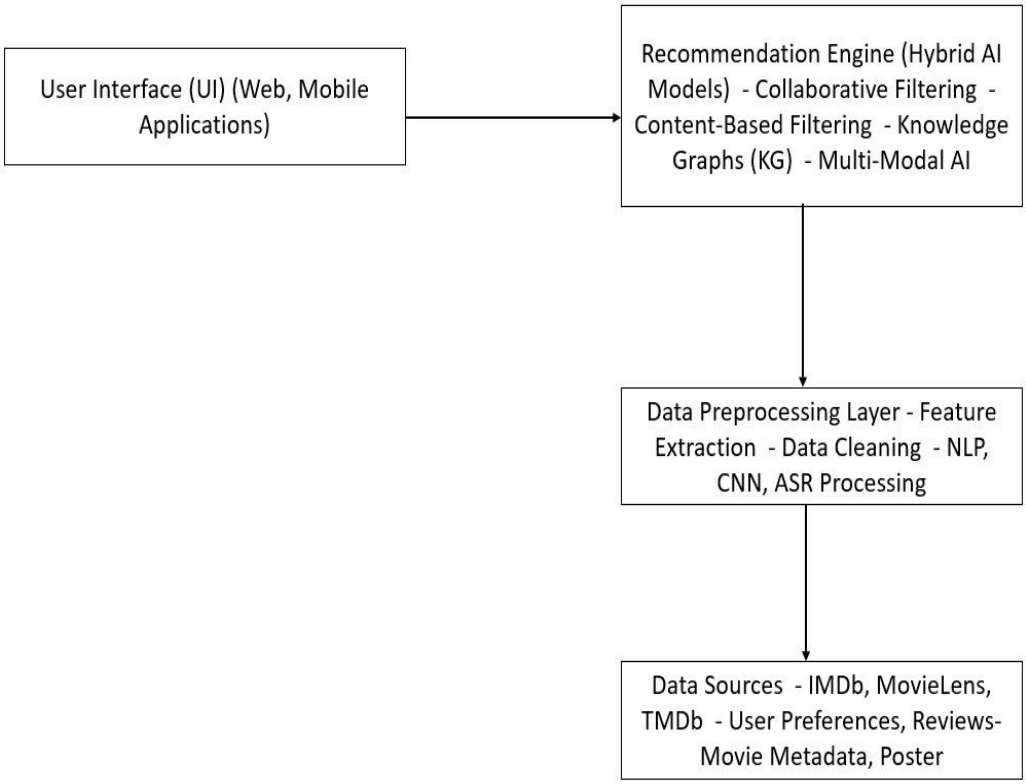

Fig 4.2.1 System Architecture

The Fig 4.2.1 The movie recommendation system follows a structured pipeline consisting of multiple layers, ensuring efficient data processing, feature extraction, and personalized recommendations. Below is the detailed breakdown of the architecture:

User Interface (UI):

- The UI can be implemented as a web or mobile application.
- Users interact with the system through the UI by searching for movies, rating films, on
  exploring recommendations.

Recommendation Engine:

The core component of the system, responsible for generating personalized movie recommendations.

It integrates multiple AI-based techniques such as:

- Collaborative Filtering: Recommends movies based on user preferences and similarities with other users.
- Content-Based Filtering: Suggests movies similar to those previously watched by analyzing metadata (genre, actors, directors, etc.).
- Knowledge Graphs (KG): Establishes relationships between entities (movies, actors, directors, etc.) to improve recommendations.

Data Preprocessing Layer:

- This layer ensures data quality before it is used by the recommendation engine.
- Feature Extraction: Identifies key features such as genres, cast, and reviews.
- Data Cleaning: Removes duplicate, irrelevant, or inconsistent data.
- NLP (Natural Language Processing): Extracts sentiment and keywords from user reviews.
- CNN (Convolutional Neural Networks): Analyzes visual data like movie posters.
- ASR (Automatic Speech Recognition): Processes audio data (if applicable) for voice-based recommendations.

Data Sources:

- The system utilizes multiple external and internal databases to gather information:
- IMDb, MovieLens, TMDb: Datasets providing movie metadata, user reviews, and ratings.
- User Preferences: Tracks individual user interactions, watch history, and ratings.
- Movie Metadata: Includes details such as genre, director, actors, and descriptions.
- Posters and Visual Data: Utilized for content-based recommendations via deep learning models.

4.3 Detailed Description of Sub Models

A movie recommendation system consists of three main sub-models that contribute to generating personalized recommendations for users. These sub-models include Collaborative Filtering (CF), Content-Based Filtering, and Hybrid Models. Below is a detailed explanation of each approach:

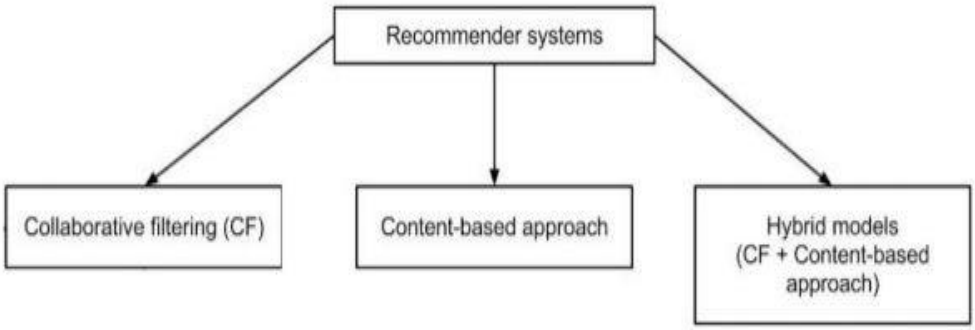

Fig 4.3.1 Recommender systems

4.3.2 Pre-Processing Module

The **pre-processing module** is responsible for cleaning and preparing the data before further analysis. It ensures that the data used for training and prediction is structured, relevant, and noise-free.

Steps in Pre-Processing:

- Data Cleaning: Removing duplicate entries, missing values, and inconsistent data.
- Text Preprocessing: Tokenization, stemming, stopword removal (for movie descriptions, reviews, etc.).
- Normalization**:** Standardizing numerical features such as ratings.
- Handling Missing Values**:** Using interpolation or collaborative techniques to fill missing user ratings or metadata.

4.3.3 Classification

1. Collaborative Filtering (CF)

Collaborative Filtering is one of the most widely used techniques in recommendation systems. It works on the principle of user-item interaction patterns.

2. Content-Based Filtering

This approach recommends movies based on their attributes (e.g., genre, actors, directors, descriptions) and matches them to the user's past preferences.

3. Hybrid Models (CF + Content-Based)

Hybrid models combine Collaborative Filtering and Content-Based approaches to leverage the strengths of both techniques.

4.3.4 Training and Validation

Dataset Splitting:

Training Set (70-80%): Used to train the model.

Validation Set (10-15%): Fine-tunes model hyperparameters.

Test Set (10-15%): Evaluates final performance.

Model Training:

Uses machine learning or deep learning models (e.g., Matrix Factorization, Neural Networks, Transformer Models).

Optimizes parameters using Gradient Descent, Adam Optimizer, etc.

Validation & Performance Metrics:

Accuracy Metrics: RMSE (Root Mean Square Error), MAE (Mean Absolute Error).

Ranking Metrics: Precision@K, Recall@K, F1-score.

A/B Testing: Comparing model performance with a baseline.

4.4 Data Flow Diagram

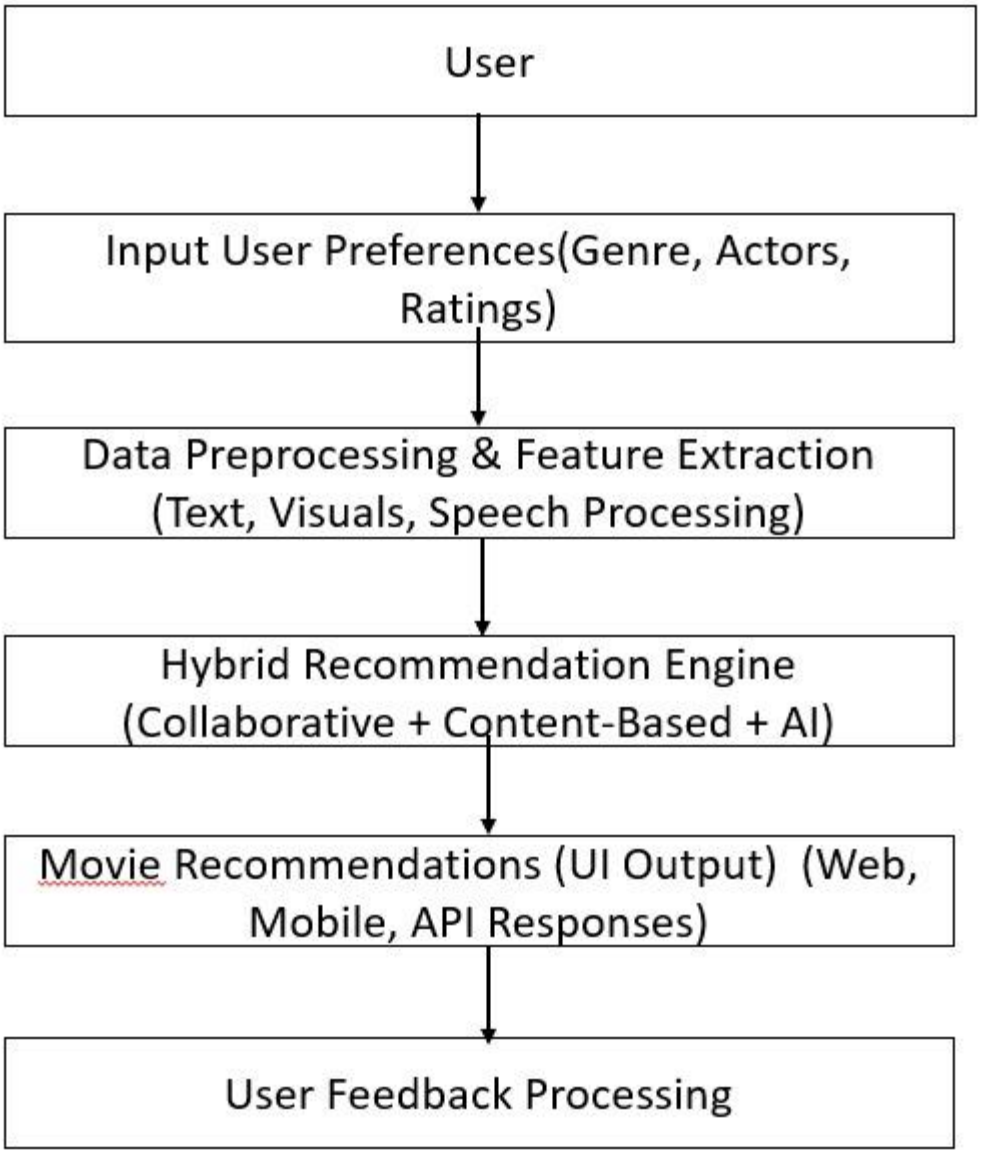

Fig 4.4.1.1 Data Flow Diagram

The Fig 4.4.1 represents Data Flow Diagram (DFD), it is a visual representation that illustrates the flow of data within a system or process. It provides a clear and concise overview of how data moves through different components, processes, and external entities within the system.

4.4.1 Level 0 DFD

Overview:

- The Level 0 DFD provides a high-level view of the system, showing how the user interacts with the system and the main output.

- It focuses on the input (user preferences) and the final output (recommendations).

Processes in Level 0 DFD:

- User Inputs Preferences – The user provides preferences such as genre, actors, and ratings.
- System Processes Data – The system processes the input and applies recommendation algorithms.
- Generate Movie Recommendations – The system provides personalized movie recommendations as output.

Entities & Data Flow:

User → [Movie Recommendation System] → Recommendations

4.4.2 Level 1 DFD

Processes in Level 1 DFD:

- Receive User Preferences: The system takes input from the user.
- Data Preprocessing & Feature Extraction: The system processes user data, extracts features (text, visuals, speech), and cleans the dataset.
- Hybrid Recommendation Engine: Uses collaborative filtering, content-based filtering, and AI techniques to predict relevant movies.
- Generate Recommendations: The system generates a list of recommended movies.
- User Feedback Processing: The system collects feedback to improve future recommendations.

4.4.3 Level 2 DFD

Processes in Level 2 DFD:

User Inputs Preferences:

- User provides movie preferences (Genre, Actors, Ratings, etc.).
- User data is stored in the User Profile Database.

Data Preprocessing & Feature Extraction:

- Text Processing (Reviews, Summaries)
- Image Processing (Movie Posters)
- Speech Processing (Trailer Audio Analysis)
- Cleaned and extracted features are stored in the Processed Data Repository.

Hybrid Recommendation Engine:

- Collaborative Filtering: Compares user behavior with similar users.
- Content-Based Filtering: Recommends movies based on features.

4.5 Sequence Diagram

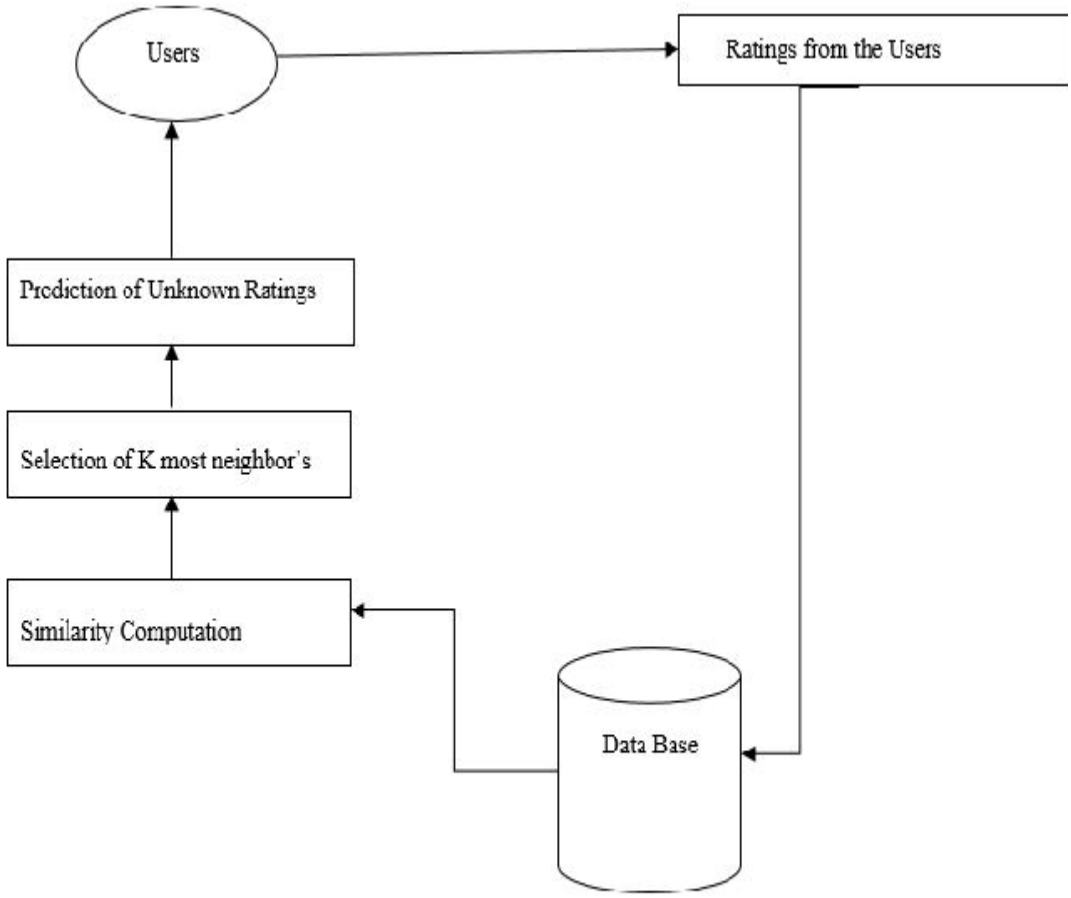

Fig. 1 Architecture of Movie recommender System using Item Based Collaborative Filtering technique.

Fig 4.5.1 Sequence diagram

The fig 5.3 represents represents the flow of processes in a **movie recommendation system** based on **item-based collaborative filtering**. Below is a step-by-step explanation of the architecture:

User Interaction & Ratings Collection

Users provide ratings for movies they have watched.

These ratings are stored in a **database** for further processing.

Similarity Computation

- The system retrieves movie rating data from the database.
- It computes similarity between different movies based on user ratings.
- This step helps in identifying movies that are closely related in terms of user preferences.

Selection of K Nearest Neighbors

- Based on similarity scores, the system selects the **K most similar movies** (nearest neighbors) for recommendation.
- This ensures that movies with similar audience preferences are grouped together.

Prediction of Unknown Ratings

- The system predicts the ratings for movies that the user has not yet rated.
- It uses information from similar movies and their ratings by other users.

Recommendation to Users

- Finally, the system recommends movies to users based on predicted ratings.
- Users receive recommendations based on their previous interactions and ratings.

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

The implementation of the IoT-based Aquaculture Monitoring System represents a comprehensive endeavour that seamlessly integrates cutting-edge technologies to revolutionize aquaculture operations. This system leverages the power of the Internet of Things (IoT), sensor networks, cloud computing, and machine learning algorithms, creating a robust and intelligent solution for monitoring, and optimizing aquaculture environments.

The implementation process commences with the strategic deployment and integration of specialized sensors capable of measuring critical water quality parameters such as temperature, pH, dissolved oxygen, turbidity, and conductivity. These sensors are meticulously integrated with the IoT network, ensuring secure and reliable data transmission to the central processing unit. Concurrently, a robust IoT network infrastructure is established, facilitating efficient data transmission from the sensor nodes to the cloud-based platform while implementing secure communication protocols and gateways to safeguard data privacy and integrity.

At the core of the system lies a scalable and secure cloud-based platform, responsible for data processing, storage, and analysis. This platform seamlessly integrates machine learning models and predictive algorithms, enabling sophisticated analysis of incoming data and generating actionable insights. The development and training of these machine learning models leverage historical data, enabling the identification of patterns, detection of anomalies, and forecasting of potential water quality changes.

The system's implementation also involves the design and development of a user-friendly interface accessible through web and mobile applications. Intuitive dashboards and visualizations are meticulously crafted to facilitate real-time monitoring of water quality parameters and system status, empowering aquaculturists with a comprehensive understanding of their operations. Furthermore, the integration of alert and notification systems ensures that aquaculturists are promptly informed of potential issues or deviations from optimal conditions, enabling timely interventions.

Scalability and flexibility are paramount considerations in the implementation process, ensuring that the IoT-based Aquaculture Monitoring System can seamlessly integrate with existing aquaculture management systems or third-party applications, while accommodating future growth and expansion of aquaculture operations. Rigorous testing of the system components, including sensor functionality, data transmission, cloud processing, and user interface, is conducted to ensure optimal performance and reliability.

The successful deployment of the system in the target aquaculture facilities is accompanied by the establishment of comprehensive maintenance protocols and procedures, encompassing system updates, sensor calibration, and troubleshooting measures. Throughout the implementation process, adherence to best practices, industry standards, and regulatory requirements is paramount, ensuring the system's reliability, security, and compliance.

By leveraging the synergistic integration of IoT, sensor networks, cloud computing, and machine learning algorithms, the IoT-based Aquaculture Monitoring System empowers aquaculturists with real-time data, predictive analytics, and intelligent decision support, enabling them to optimize environmental conditions, enhance operational efficiency, and promote sustainable practices in their aquaculture operations.

5.2 Hardware Implementation

Water quality monitoring is crucial for various applications, including environmental protection, public health, and industrial processes. Ensuring the safety and suitability of water sources requires continuous monitoring of various parameters that can affect water quality. This project aims to develop a comprehensive water quality monitoring system that can measure and analyze multiple parameters in real-time.

The proposed system utilizes an array of sensors to measure different water quality parameters, including temperature, conductivity, pH, and turbidity. These parameters provide valuable insights into the physical, chemical, and biological characteristics of the water. By monitoring these parameters, it is possible to detect potential contaminants, assess water suitability for various applications, and take necessary actions to maintain or improve water quality.

The hardware setup for this project consists of the following main components:

- Sensors (Temperature, pH, Conductivity, Turbidity)

- Esp32
- Power Supply
- Battery

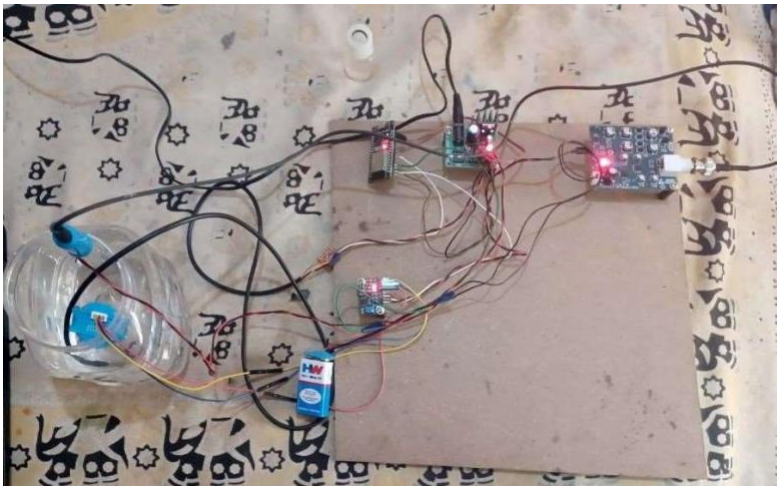

Fig 5.2.1 Hardware Setup for Real-Time Aquaculture Monitoring System

The Fig 6.1 shows a hardware setup for Aquaculture monitoring system. The setup consists of various sensors connected to a microcontroller board (likely an ESP32) through wires. The sensors include temperature, conductivity, pH, and turbidity sensors, which are designed to measure different parameters of water quality.

The temperature sensor measures the water temperature, the conductivity sensor measures the ability of water to conduct electrical current (an indication of dissolved solids), the pH sensor measures the acidity or basicity of the water, and the turbidity sensor measures the cloudiness or haziness of the water caused by suspended particles.

The data collected by these sensors is sent to the ESP32 microcontroller board, which can process and transmit the data wirelessly or to a connected device for further analysis or display. The setup also includes a power supply unit to provide the necessary power to the components.

This hardware setup allows for real-time monitoring of various water quality parameters, which can be useful in various applications such as environmental monitoring, water treatment, aquaculture, and industrial processes where water quality is critical.

5.3 Software Implementation

The IoT-based Aquaculture Monitoring System features a comprehensive web application designed to provide aquaculturists with a centralized platform for monitoring and managing their operations. Developed using modern web technologies, including HTML and CSS the web application offers a responsive and intuitive user experience across various devices and platforms.

The web application presents a holistic overview of aquaculture, encompassing its diverse purposes, species cultivated, sustainable seafood production goals, economic impacts, environmental considerations, and the potential to address food security and poverty. It highlights the multifaceted nature of aquaculture, spanning commercial, recreational, and conservation objectives, while emphasizing responsible practices and the role of cutting edge technologies like recirculating and integrated multi-trophic systems.

A key feature of the web application is the real-time sensor data visualization interface which displays live data from various sensors monitoring critical parameters such as temperature, pH, turbidity, and conductivity. This interface leverages interactive charts and graphs, enabling users to continuously monitor the aquatic environment and quickly identify any deviations from optimal conditions.

Fig 5.3.1 Result page of water status in aquaculture farm

Furthermore, the web application incorporates a water quality assessment feature (Fig 6.2), which presents the key water quality parameters measured in the aquaculture farm. Based on these values, the application provides an overall assessment of whether the water conditions are suitable or not for the aquatic species. This feature empowers users

to take timely actions to maintain optimal conditions, ensuring the health and productivity of their aquaculture operations.

Adhering to modern design principles, the web application prioritizes user experience and accessibility. The interface features responsive layouts, intuitive navigation, and visually appealing dashboards, ensuring a seamless and user-friendly interaction. Additionally, the application incorporates accessibility features to cater to users with varying abilities.

The web application seamlessly integrates with other modules of the IoT-based Aquaculture Monitoring System, such as the Data Collection and Communication Module, the Data Processing and Analysis Module, and the Alerts and Notifications Module. This integration enables real-time data visualization, advanced analytics, and timely alerts and notifications to be presented through the web application, providing users with a comprehensive and centralized platform for monitoring and managing their aquaculture
operations.

//Code for the web application to operate the system

Home.html

```
{% extends 'main.html' %}

{% block content %}

{% if message %}

      <div class="alert alert-danger">{{ message }}</div>

   {% endif %}

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta      name="viewport"      content="width=device-width, initial-scale=1,
       shrinktofit=no">

    <meta name="description" content="">
```

```
    <meta name="author" content="">
    <title>Aquaculture Monitoring System</title>
    <link rel="canonical" href="https://getbootstrap.com/docs/4.0/examples/carousel/">
    <!-- Bootstrap core CSS -->
    <link href="../../dist/css/bootstrap.min.css" rel="stylesheet">
   <style>
#acc{
color: blue;
      }
     #acc:hover{
color: brown;       }
body
      {
       background-color: rgb(131, 223, 240);
      }       h1{         font-family: 'Times New
Roman', Times, serif;
      }       h2{
color: lightyellow;
      }
p{            font-family: 'Times New Roman',
Times,
serif;        font-size: large;         color: black;
      }
```

```
  </style>
 </head>
 <body >
  <main role="main">
   <section class="jumbotron p-3 p-md-3 text-black rounded bg-light text-center">
    <div class="container">
     <h1 class="jumbotron-heading"><b>AQUACULTURE MONITORING
SYSTEM</b></h1>
    </div>
   </section>
    <h2>About Aquaculture</h2>
    <p><b>
   </div>
  </div>
      </section>
  </main>

  <!-- Bootstrap core JavaScript -->
  <!-- Placed at the end of the document so the pages load faster -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aZGJwFDMVNA/GpGFF93hXpG5
K
kN" crossorigin="anonymous"></script>
  <script>window.jQuery || document.write('<script
src="../../assets/js/vendor/jqueryslim.min.js"><\/script>')</script>
  <script src="../../assets/js/vendor/popper.min.js"></script>
```

```
    <script src="../../dist/js/bootstrap.min.js"></script>

    <!-- Just to make our placeholder images work. Don't actually copy the next line! -->
<script src="../../assets/js/vendor/holder.min.js"></script>

  </body>

</html>        {%

endblock %}
```

Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>Aquaculture Monitoring System</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <meta name="og:title" content="Aquaculture Monitoring System">

  <meta name="og:image" content="static/logo1.png">

  <meta name="Keywords" content="Flask, Machine Learning, Deep Learning, Artificial Intelligence, AI, ML,DL, Web Development">

  <meta name="description" content="A Machine Learning and Deep Learning based webapp for Aquaculture monitoring system.">

  <title>Aquaculture Monitoring System</title>

  <link rel="icon" href="{{ url_for('static', filename = 'logo1.png') }}" type="image/icon type">

  <p> Aquaculture Monitoring System </p>
```

```
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T
"
crossorigin="anonymous">

   <link                    href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/fontawesome.min.css" rel="stylesheet"/>

   <link                                                    rel="canonical"
href="https://getbootstrap.com/docs/4.0/examples/stickyfooter/">

   <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8a
bt
TE1Pi6jizo" crossorigin="anonymous"></script>

   <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W
1
" crossorigin="anonymous"></script>

   <script
      src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

  <style>

   #home, #form1, #form2, #form3,
#form4{      display: none;      border: 1px
solid black;       padding: 25px 50px;
width: 70%;      margin:
```

```
auto;
    }
    #home.active,       #form1.active,  #form2.active,  #form3.active,  #form4.active{
display: block;
}
#form1  button,  #form2  button{       backgroundcolor:
greenyellow;
}
#form1 button:hover, #form2 button:hover{    background-
color: green;
}
#form1     a,     #form2     a{
color: blue;
}
#form1     a:hover,     #form2     a:hover{
color: brown;
} body
{
  background-image:
url("https://t3.ftcdn.net/jpg/02/92/90/56/360_F_29290566_yFUJNJPngYeRNlrRL4hAp
H WxuYyRY4kN.jpg");  background-size: 100% 100vh;    background-repeat: no-repeat;
}
.container{    padding-
```

```
left: 35%;
}
  </style>
</head>
<body>
        <nav class="navbar navbar-expand-sm navbar-dark fixed-top bg-dark" style="background-color: black !important;">
    <a style="text-decoration: none; color: white" href="{{ url_for('home') }}">Home</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" datatarget="#navbarNav" aria-controls="navbarNav" aria-expanded="false" arialabel="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
        <!-- <li class="nav-item active">
          <a href="{{ url_for('home') }}" class="nav-link">Home</a>
        </li> -->
         <li class="nav-item active">
           <a class="nav-link" href="{{ url_for('index') }}">Log In</a>
         </li>
      </ul>
    </div>
  </nav>
```

```
<br><br><br>
{% if msg  %}
        <div class="jumbotron ">
        <center><p class="lead">{{msg}}</p></center>
        </div>
    </div>
    </div>
      {% endif %}
<div class="container">
  <form  method="post" action="{{ url_for('userlog')}}" id="form1" class="active">
    <div class="col-lg-12">
     <center>
       <img class="rounded-circle" src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcRDzdC8tyLdAaX5o6jygK9y6ruUdNcAyNqLj
Z
TW1XuN6w&s" alt="Generic placeholder image" width="100" height="100">
    <h2>User Login</h2>
  </center>
</div>
    <div class="form-group">
     <label for="username">Username:</label>
     <input type="text" class="form-control" id="username" placeholder="Enter
username" name="name">
    </div>
    <div class="form-group">
```

```
      <label for="password">Password:</label>
      <input type="password" class="form-control" id="password" placeholder="Enter password" name="password">
    </div>
    <button type="submit" class="btn btn-default">Submit</button></br></br>
    <p>Click here to <a data-value="#form2" onclick="toggleform(event)">Sign up</a></p>
  <!-- </center> -->
  </form>
  <form method="post" action="{{ url_for('userreg')}}" id="form2">
    <div class="col-lg-12">
      <center>
        <img class="rounded-circle" src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRDzdC8tyLdAaX5o6jygK9y6ruUdNcAyNqLjZ
TW1XuN6w&s" alt="Generic placeholder image" width="100" height="100">
    <h2>User Registration</h2>
  </center>
 </div>
    <h2></h2>
    <div class="form-group">
      <label for="username">Username:</label>
      <input type="text" class="form-control" id="username" placeholder="Enter username" name="name">
    </div>
```

```
<div class="form-group">
  <label for="email">Email:</label>
  <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
</div>
<div class="form-group">
  <label for="phone">Mobile No.:</label>
  <input type="tel" class="form-control" id="phone" placeholder="Enter mobile number" name="phone" pattern="[0-9]{10}">
</div>
<div class="form-group">
  <label for="password">Password:</label>
  <input type="password" class="form-control" id="password" placeholder="Enter password" name="password"  minlength="5">
</div>
<button type="submit" class="btn btn-default">Submit</button></br></br>
<p>Click here to <a data-value="#form1" onclick="toggleform(event)">Sign in</a></p>  </form>
</div> <script>  function toggleform(e) {
var Id =(e.target.getAttribute('data-value'))
console.log(Id)  let Items= ['#form1',
'#form2'];  Items.map(function(item) {
if(Id === item ) {  console.log("ggg")
  $(item).addClass("active");
```

```
        }
else {
            $(item).removeClass("active");
        }
    })
}
</script>
</body>
</html>
```

Fetal.html

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="og:title" content="Aquaculture Monitoring System">
  <meta name="og:image" content="static/logo1.png">
  <meta name="Keywords" content="Flask, Machine Learning, Deep Learning, Artificial Intelligence, AI, ML,DL, Web Development">
  <meta name="description" content="A Machine Learning and Deep Learning based webapp for Aquaculture monitoring system.">
  <title>Aquaculture Monitoring System</title>
  <link rel="icon" href="{{ url_for('static', filename = 'logo1.png') }}" type="image/icon type">
  <p> Aquaculture Monitoring System </p>
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9
J voRxT2MZw1T"
crossorigin="anonymous">
<link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/fontawesome.min.css" rel="stylesheet"/>
<link rel="canonical"
href="https://getbootstrap.com/docs/4.0/examples/stickyfooter/"> <script
src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8a
bt
TE1Pi6jizo" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W
1
" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
<style> body
{
background-image:
url('https://i.pinimg.com/originals/80/50/e1/8050e13f1236e94dae432f055efb850f.jpg');
background-size: 100% 100vh; background-repeat: no-repeat;
}
```

```
  </style>
</head>
<body>
        <nav class="navbar navbar-expand-sm navbar-dark fixed-top bg-dark" style="background-color: black !important;">
      <a style="text-decoration: none; color: white" href="{{ url_for('home') }}">Home</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" datatarget="#navbarNav" aria-controls="navbarNav" aria-expanded="false" arialabel="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
       </button>
       <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ml-auto">
            <li class="nav-item active">
            <a class="nav-link" href="{{ url_for('logout') }}">Logout</a>
           </li>
        </ul>
       </div>
     </nav>
        <br>
<br>
<main>
        <div class="container-fluid" style="margin-bottom: 10px;">
```

```
<div class="row"  style="margin-bottom: 45px;">
   <div class="col-md-2"></div>
   <div class="col-md-8">
      <div class="col-lg-12">
         <center>
           <img class="rounded-circle" src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSMCEgWwO4B45LagliI0dNNQ1S5nbpCoCqq DfAVwZIQm95fCyz-JDYroJoppAFD7nAgC4w&usqp=CAU" alt="Generic placeholder image" width="80" height="80">
      </center>
      <center><h1 style="color: red;"><b>Aquaculture Monitoring System</b></h1></center>
      <div class="card card-body" style="border: 1px solid black;">
      <form class="form-horizontal" action="{{ url_for('predictPage') }}" method="POST">
            <div class="row">
               <!-- ^ Turbidity, Conductivity, ph, temprrature -->
               <div class="col-md-12">
                  <div class="form-group">
                     <input style="border: 1px solid black;" class="form-control" type="text" name="a_id" placeholder="Enter id" required step="any", value="{{name}}">
                  </div>
               </div>
            </div>
```

```
<div class="row">
  <div class="col-md-6">
    <div class="form-group">
      <input style="border: 1px solid black;" class="form-control" type="number" name="Turbidity" required placeholder="Turbidity" value="{{turb}}" step="any">
    </div>
  </div>
  <div class="col-md-6">
    <div class="form-group">
      <input style="border: 1px solid black;" class="form-control" type="number" name="Conductivity" required placeholder= "Conductivity" value="{{con}}" step="any">
    </div>
  </div>
</div>
<div class="row">
  <div class="col-md-6">
    <div class="form-group">
      <input style="border: 1px solid black;" class="form-control" type="number" name="Temperature"  required placeholder="Temperature" step="any" value="{{temp}}">
    </div>
  </div>
  <div class="col-md-6">
```

```
                <div class="form-group">
                    <input style="border: 1px solid black;" class="form-control" type="number" name="Ph" required placeholder="Ph" step="any" value="{{ph}}">
                </div>
              </div>
            </div>
            <input type="submit" class="btn btn-info btn-block" value="Predict" style="background-color: navy" step="any">
          </form>
        </div>
    </div>
    <div class="col-md-2"></div>
</div>
</main>
</body>
</html>
```

Predict.html

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="og:title" content="Aquaculture Monitoring System">
  <meta name="og:image" content="static/logo1.png">
```

```
<meta name="Keywords" content="Flask, Machine Learning, Deep Learning, Artificial Intelligence, AI, ML,DL, Web Development">

<meta name="description" content="A Machine Learning and Deep Learning based webapp for Aquaculture monitoring system.">

<title>Aquaculture Monitoring System</title>

<link rel="icon" href="{{ url_for('static', filename = 'logo1.png') }}" type="image/icon type">

<p> Aquaculture Monitoring System </p>

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/fontawesome.min.css" rel="stylesheet"/>

  <link rel="canonical" href="https://getbootstrap.com/docs/4.0/examples/stickyfooter/">

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abt TE1Pi6jizo" crossorigin="anonymous"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1 " crossorigin="anonymous"></script>
```

```
<script
    src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>

<main>

  <div class="row"  style="margin-bottom: 200px;">

    <div class="col-md-3"></div>

    <div class="col-md-12">

     <center><h1 class="display-4"> <b>Water in Aquaculture Farm is  <br>
{{status}}
!!!!</b></h1></center>

      <div class="row">

        <div class="col-md-2"></div>

        <div class="col-md-8"><a href="{{ url_for('fetalPage')}}" class="btn btn-block
btn-primary" style="background-color: navy">Recheck</a></div>

        <div class="col-md-4"></div>

      </div>

    <div class="container my-3">

      <div class="row text-center">

        <!-- temperature data card -->

        <div class="col-xs-12 col-6 col-lg-3">

            <div class="card custon-shadow mt-3 pkfloat">

              <h5 class="card-title border-bottom p-1 text-light bg-primary rounded"
style="background-color: blue !important;">TEMPERATURE</h5>

                <span class="card-text"><br>
```

```
                    <span class="h2">
                        {{tempe}}
                    </span>
                    <span class="text-
primary">°C</span> <img
src="/static/thermometer.svg" width="55"
style="margin-left: 30px; margin-bottom: 15px">
                    </span>
                </div>
            </a>
        </div>
        <!-- turbidity card -->
        <div class="col-xs-12 col-6 col-lg-3">
            <div class="card custon-shadow mt-3 pkfloat">
                <h5 class="card-title border-bottom p-1 text-light bg-primary rounded"
style="background-color: blue !important;">TURBIDITY</h5>
                <span class="card-text"><br>
                    <span class="h2">{{turb}}</span>
                    <span class="text-primary">NTU</span>
<img src="/static/turbidity.svg" width="55" style="margin-
left: 30px; margin-bottom: 15px ">
</span>

                </div>
```

```
            </a>
        </div>
        <!-- ph card -->
        <div class="col-xs-12 col-6 col-lg-3">
            <div class="card custon-shadow mt-3 pkfloat">
                <h5 class="card-title border-bottom p-1 text-light bg-primary rounded" style="background-color: blue !important;">pH</h5>
                <span class="card-text"><br>
                    <span class="h2">{{ph}} </span>
                    <img src="/static/ph.svg" width="55" style="margin-left: 30px; marginbottom: 15px ">
                </span>
            </div>
        </a>
    </div>
    <!-- conductivity card -->
    <div class="col-xs-12 col-6 col-lg-3">
        <div class="card custon-shadow mt-3 pkfloat">
            <h5 class="card-title border-bottom p-1 text-light bg-primary rounded" style="background-color: blue !important;">Conductivity</h5>
            <span class="card-text"><br>
                <span class="h2">{{con}}</span>
```

```
                    <span class="text-primary">ohms</span>
<img src="/static/watermeter.png" width="55"
style="margin-left: 30px; margin-bottom: 15px ">
                </span>
                <p class="tx-12"><span class="tx-
success">{{percentage_conductivity_change}}%
                    ({{conductivity_change}})</span> past hour</p> -->
            </div>
          </a>
        </div>
      </div>
      <br>
        <hr class="my-4">
        <center><p style="color:black ;font-weight: bold;font-size:
larger;">INCONSISTENCY</p>
        <p style="color:black; font-size: large ;font-weight: bold;"> {{tur}}</p>
        <p style="color:black;font-size: large ;font-weight: bold;">{{cond}}</p>
        <p style="color:red;font-size: large ;font-weight: bold;">{{temp}}</p>
        <p style="color:black;font-size: large ;font-weight:
bold;">{{p}}</p></center>
        <p class="lead">
        <a class="btn btn-primary btn-lg"
    href="https://www.was.org/" role="button">Learn more </a>
        </p>
        </div>
```

```
        </div>
    </div>
</main>
</body>
</html>
```

5.4 Summary

The aquaculture monitoring system is designed to monitor and assess water conditions in fish farms or aquaculture facilities. It employs various sensors, such as temperature, pH, turbidity, and conductivity sensors, to collect real-time data from the water bodies. The collected data is processed and analysed using specialized algorithms and prediction models to evaluate the suitability of the water conditions for fish growth. The system provides users with a web-based interface to view the water condition status, receive alerts for unfavourable conditions, and access recommendations for corrective actions. By automating the monitoring process and providing timely insights, the system aims to optimize fish productivity, ensure sustainable practices, and enable efficient management of aquaculture operations.

CHAPTER 6

RESULTS

6.1 Introduction

An IoT-based aquaculture monitoring system integrates sensors to track water quality, temperature, and pH levels in real-time. Data is transmitted to a central platform, enabling automated alerts and adjustments to maintain optimal conditions, enhancing fish health, growth, and reducing operational costs. 6.2 Test Cases

ID	Description	Expected Output	Actual Output	Remarks
1	Data Collection	The data is collected from sensors i.e. Temperature, pH, Conductivity, and turbidity	Same as expected	Pass
2	Transfer Data	Collected data is sent to cloud	Same as expected	Pass
3	Process Data	Collected data is processed based on trained model	Same as expected	Pass
4	Predict Output	Model predicts if the status of water is in good condition or not	Same as expected	Pass
5	Send alert	If water condition is not good, user receives the message	Same as expected	Pass

Table 6.2.1 Test cases for Aquaculture Monitoring System

Unit Testing: Individual units or modules of the software, such as sensor data acquisition, data processing, prediction algorithms, and user interface components, can be tested independently to verify their correct functionality and behaviour.

Integration Testing: After unit testing, the different modules and components can be integrated and tested together to ensure proper communication and data flow between them, as well as to identify any interface-related issues or compatibility problems.

System Testing: The complete aquaculture monitoring system, including hardware (sensors) and software components, can be tested as a whole to validate its end-toend functionality, performance, and compliance with the specified requirements.

6.3 Working Cases

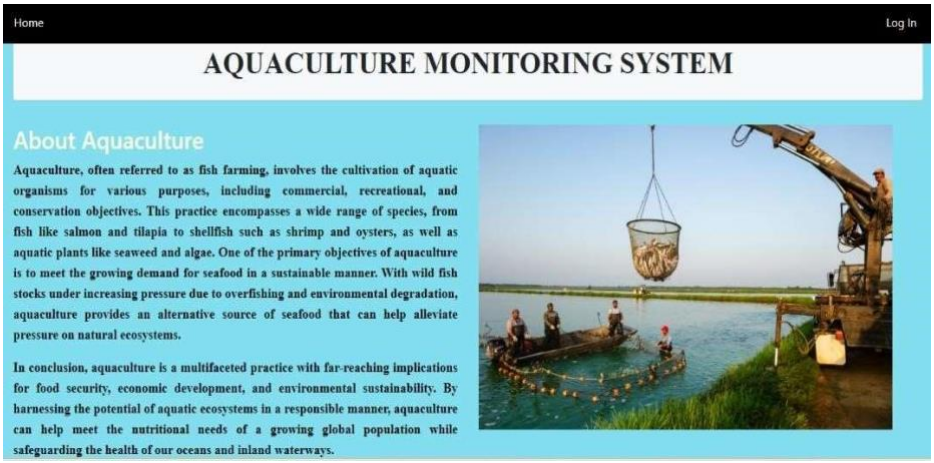

Fig 6.3.1 Home Page of Aquaculture Monitoring System Web Page

Fig 6.3.1 provides a comprehensive overview of aquaculture, covering its diverse purposes, species cultivated, sustainable seafood production goals, economic impacts, environmental considerations, and potential for addressing food security and poverty. It highlights aquaculture's multifaceted nature, spanning commercial, recreational, and conservation objectives while underscoring responsible practices and the role of technologies like recirculating and integrated multi-trophic systems.

Provides a comprehensive overview of aquaculture, covering its diverse purposes, species cultivated, sustainable seafood production goals, economic impacts, environmental considerations, and potential for addressing food security and poverty. It highlights aquaculture's multifaceted nature, spanning commercial, recreational, and conservation objectives while underscoring responsible practices and the role of technologies.

Fig 6.3.2 Sensor Data Display

The Fig 6.3.2 displays a user interface for an "Aquaculture Monitoring System." The interface allows users to enter values into input fields, likely representing data collected from sensors monitoring an aquaculture operation. After entering the values, the user can click the "Predict" button, presumably to generate an analysis or prediction based on the inputted sensor data. The interface has a calming beach background image behind the data entry section.

Fig 6.3.3 Water Quality Monitoring Results

The Fig 6.3.3 displays the key water quality parameters (temperature, turbidity, pH, and conductivity) measured in an aquaculture farm. It provides an overall assessment of whether the water conditions are suitable or not based on these values, allowing users to

quickly monitor and take necessary actions to maintain optimal conditions for the aquatic species.

6.4 Comparative Analysis

[1]	Online Survival Prediction System	Live Fish Waterless Transportation System	Area Under the Curve (AUC), False Positive Rate	Proactive survival prediction, Reduces transport losses	Dependency on accurate physiological stress metrics
[2]	Machine Learning Algorithms, Data Mining	IOT, Data Analytics, Cloud Computing	Accuracy, Precision, Root Mean Squared Error	Effective Resource Allocation, Early Detection of Water Leaks and Wastage	Technical Expertise, Initial Investment, and Infrastructure
[3]	Deep Learning Method	Web Development Cloud Computing	Response Time, Throughput, Error rate, Resource Utilization	Accurate Classification of marine water quality	Complexity and learning curve associated with using DNN.
[4]	Data Processing, Backpropagation Neural Network	Apache Hadoop, Apache Spark, Microsoft Azure	Accuracy, Precision, Root Mean Squared Error	Scalability, Parallel Processing, Accuracy	Computational complexity and Resource Requirements
[5]	Sensor Data Fusion, Machine Learning	Arduino, Raspberry Pi, AWS IOT	Sensor Data, System Efficiency, Crop Yield	Real-time Monitoring, Remote Accessibility	Nutrient Levels, Water Parameters

6.5 Summary

The IoT-based aquaculture monitoring system was developed to address the challenges of maintaining optimal conditions in aquaculture environments. The system integrates various sensors, an Arduino microcontroller, and IoT technology to provide

real-time monitoring and management of key water parameters, ensuring the health and growth of aquatic organisms.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

The aquaculture monitoring system presents a comprehensive and automated solution for monitoring water conditions in fish farms and aquaculture facilities. By integrating multiple sensors, data processing algorithms, and predictive models, the system provides real-time insights into the water quality parameters critical for fish growth and productivity. The userfriendly web interface facilitates easy access to water condition status, alerts, and recommendations, enabling aquaculture operators to make informed decisions and take timely actions to maintain optimal water conditions.

Through extensive testing and validation, the system has demonstrated its reliability, accuracy, and robustness in various operational scenarios. The successful implementation and deployment of the aquaculture monitoring system have the potential to significantly improve fish yield, reduce operational costs, and promote sustainable aquaculture practices.

7.2 Future Enhancement

While the current aquaculture monitoring system offers a comprehensive solution, there are several potential enhancements that could further improve its functionality and expand its

capabilities:

1. Automated control and feedback mechanisms: Integrating the system with automated control systems could enable real-time adjustments and interventions based on the monitored water conditions, minimizing the need for manual interventions.
2. Expansion to monitor additional parameters: The system could be extended to include sensors and monitoring capabilities for additional water quality parameters, such as dissolved oxygen levels, nutrient concentrations, or the presence of specific contaminants.

3. Mobile application development: Developing a dedicated mobile application could provide aquaculture operators with real-time access to water condition data and alerts,

   enabling them to monitor and manage their operations from anywhere.
4. Integration with external data sources: Incorporating external data sources, such as weather forecasts, environmental data, or market trends, could provide additional context and insights for optimizing aquaculture operations and decision-making processes.
5. Integrating with blockchain technology: Explore the use of blockchain for secure data storage, traceability, and transparency, enabling stakeholders to track the origin and history of aquaculture products.

These future enhancements can be prioritized and implemented based on the specific needs, resources, and technological advancements in the aquaculture industry, allowing the monitoring system to continually evolve and provide cutting-edge solutions for sustainable and efficient aquaculture operations.

REFERENCES

[1] K. P. Rasheed Abdul Haq and V. P. Harigovindan "Water Quality Prediction For Smart Aquaculture Using Hybrid Deep Learning Models" 2020 IEEE.

[2] Liang Kuang, Pei Shi, Chi Hua, Beijing Chen, And Hui Zhu "An Enhanced Extreme Learning Machine for Dissolved Oxygen Prediction in Wireless Sensor Networks" 2020 IEEE.

[3] Dashe Li, Jiajun Sun, Huanhai Yang , And Xueying Wang "An Enhanced Naive Bayes Model For Dissolved Oxygen Forecasting In Shellfish Aquaculture" 2020 IEEE.

[4] Wenjun Liu, Shuangyin Liu, Shahbaz Gul Hassan, Yingying Cao, Longqin Xu, Dachun Feng, Liang Cao, Weijun Chen, Yaocong Chen, Jianjun Guo, Tonglai Liu and Hang Zhang “A Novel Hybrid Model to Predict Dissolved Oxygen for Efficient Water Quality in Intensive Aquaculture" 2023 IEEE.

[5] Yongjun Zhang, Yufu Ning, Xiaoshuan Zhang, Branko Glamuzina and Shaohua Xing "Multi-Sensors-Based Physiological Stress Monitoring and Online Survival Prediction System for Live Fish Waterless Transportation" 2020 IEEE.

[6] Shouqi Cao, Lixin Zhou, And Zheng Zhang "Prediction of Dissolved Oxygen Content in Aquaculture Based on Clustering and Improved Elm" 2021 IEEE.

[7] Gissella Bejarano, Mayank Jain, Arti Ramesh, Anand Seetharam, Aditya Mishra “Predictive Analytics for Smart Water Management in Developing Regions" 2018 IEEE.

[8] Carlin C.F. Chu, S.C. Yuen, Y.K. Wong "Deep Neural Network for Marine Water Quality Classification with The Consideration of Coastal Current Circulation Effect" 2017 IEEE.

[9] Wen Yan, Li Min and Ye Yu “MapReduce-Based Bp Neural Network Classification of Aquaculture Water Quality" 2020 IEEE.

[10] Praveen C Menon "IoT Enabled Aquaponics with Wireless Sensor Smart Monitoring" 2020 IEEE