Loan Eligibility Prediction System

24-25J-268

Our team

Dilinika pathirathna

IT21164644

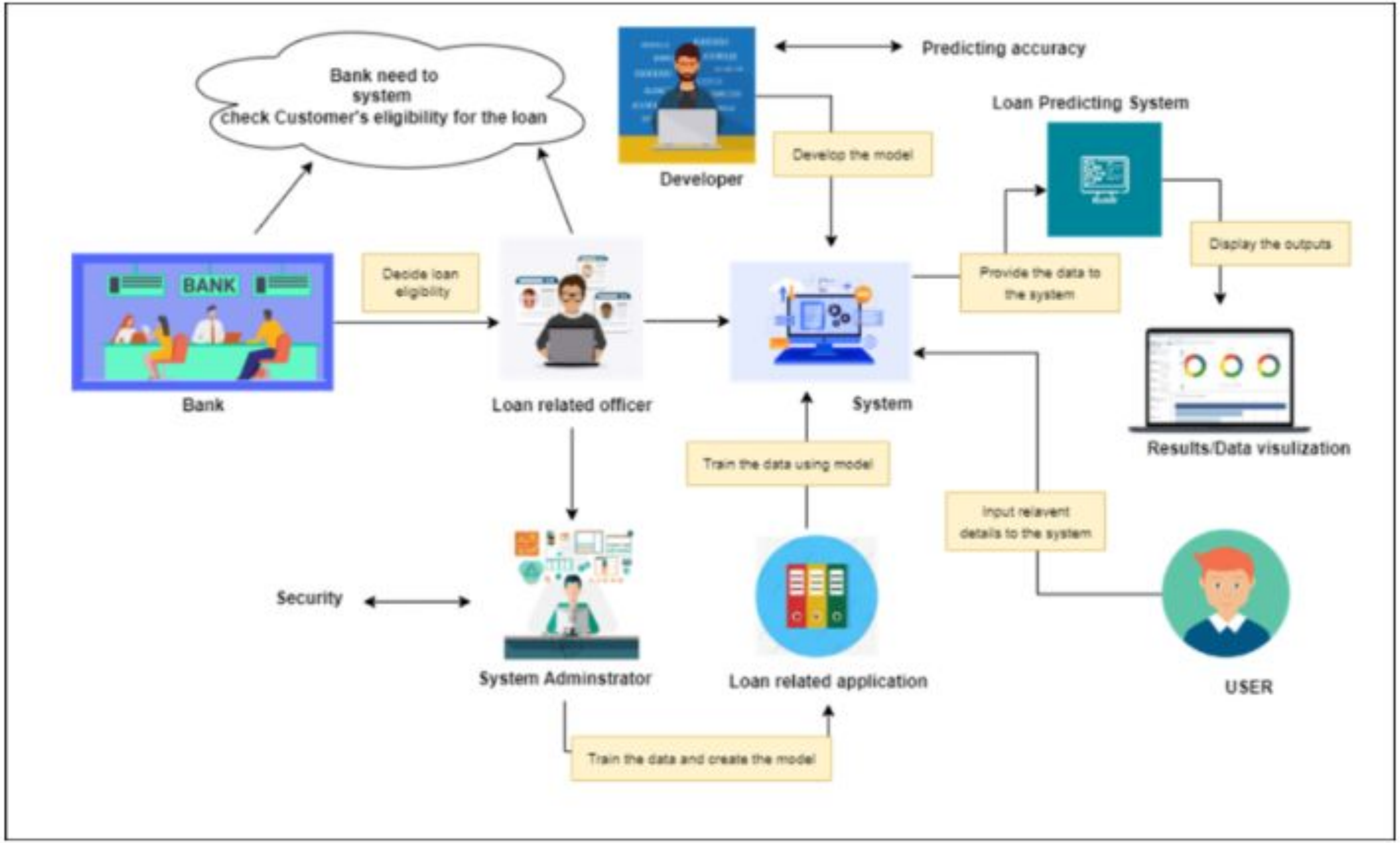
P.R.I.PRAVEAN

IT21191060

Hilma Illiyas

IT21142178

OVERALL SYSTEM DIAGRAM



linika

thirathna

T21164644

Introduction

BACKGROUND

PROBLEM DEFINITION

IMPLEMENTATIONS

OUTCOME AFTER PP1

BACKGROUND

Loan eligibility prediction is crucial in the banking sector as it helps financi
institutions make **faster, data-driven, and accurate** lending decisions. Traditional loan approval processes are often **time-consuming and subjective**, leading to delays and inconsistencies. By using **machine lear techniques like encoding**, banks can:

- **Improve Efficiency** – Automate loan approval, reducing manual w
- **Enhance Accuracy** – Analyze financial data objectively, minimizing human bias.
- **Provide Real-Time Decisions** – Enable customers to get instant eligibility results and loan amount estimations.
- **Reduce Default Risk** – Assess risk more effectively by considering multiple financial factors

LEM DEFINITION

s' current loan eligibility systems typically provide a Yes/No outcome.

do not forecast the loan amount that a customer can comfortably handle.

on restricted static facts (e.g. Income, CRIB), and ignore dynamic elements such as:

Real-time financial behavior

Debt/income ratio

lity to adjust to changing economic situations (for example, inflation and interest rates).

systems are black-box models that lack transparency and explainability.

e is no provision for personalized repayment plans or borrower advice.

ESEARCH GAP

entify three research areas after consideration above existing works,

ne existing works were given status according to whether they were eligible
ot for the bank loan process, didn't implement predicting eligible amount.

ne existing works consider few datasets and above research papers
ider few inputs for the train the model, therefore eligibility predicting
racy became low.

he existing works didn't consider how to change the model and prediction
oach with economic patterns.

LEMENTATIONS

put Unit
user information (personal, financial, and credit history) .

cessing Unit.
s data cleaning, encoding, normalization, and feature selection to prepare input for the model.

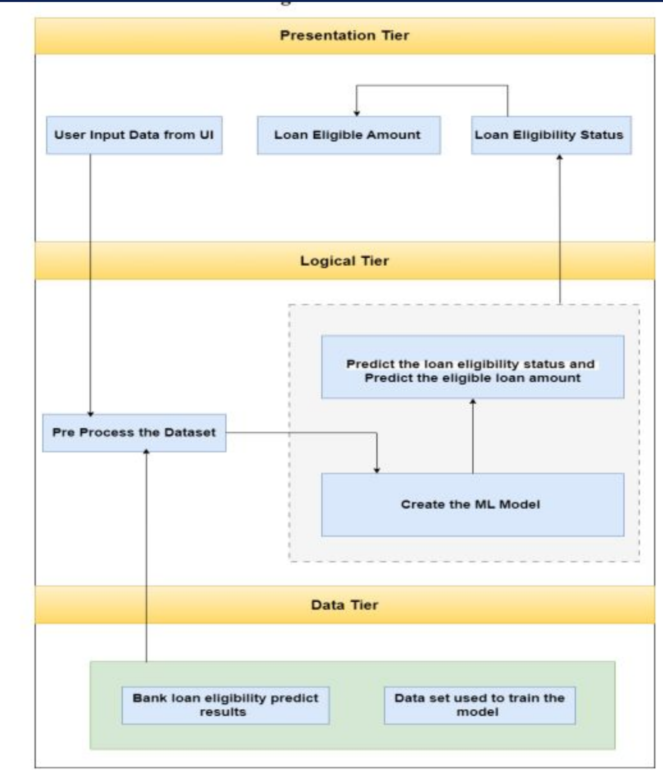
ligibility Prediction. Unit: Uses a trained ML classifier (e.g., Logistic Regression) to determine "Eligible" or
ible".

Limit Estimation Unit.
e, determines a safe loan amount using rules or a regression-based technique.

lculation Unit.
es repayment schedules, monthly EMIs, and interest-based plans.

t Interface Unit.
the forecast result, loan amount, and payback split in an easy-to-understand format (for example,
rd or web UI).

Implementation Overview



Three-Tiered Architecture

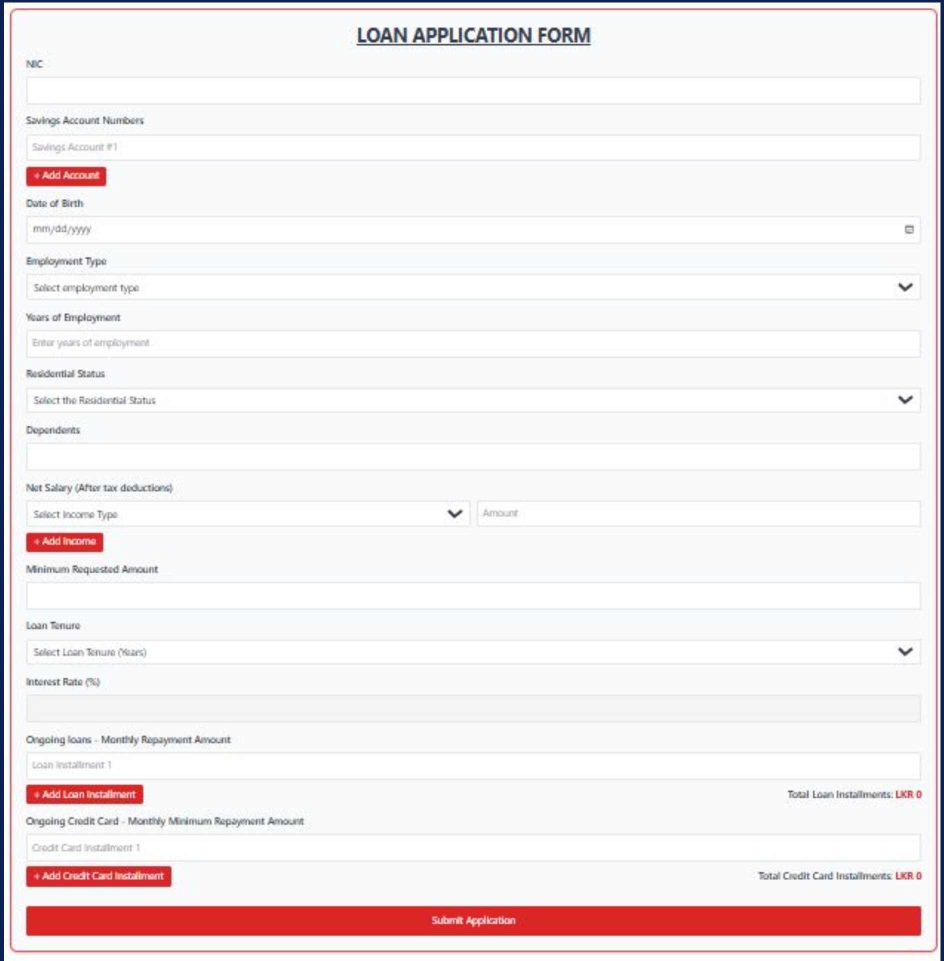
Functional Unit 1 – Data Input & Preprocessing

User provides inputs- income, employment, credit history.

Preprocessing handles:

- Missing values
- Encoding (categorical to numeric)
- Normalization for ML readiness

Data passed to model in clean, structured format.



LOAN APPLICATION FORM

NIC

Savings Account Numbers

Savings Account #1

+ Add Account

Date of Birth

mm/dd/yyyy

Employment Type

Select employment type

Years of Employment

Enter years of employment

Residential Status

Select the Residential Status

Dependents

Net Salary (After tax deductions)

Select Income Type

Amount

+ Add Income

Minimum Requested Amount

Loan Tenure

Select Loan Tenure (Years)

Interest Rate (%)

Ongoing loans - Monthly Repayment Amount

Loan Installment 1

+ Add Loan Installment

Total Loan Installments: LKR 0

Ongoing Credit Card - Monthly Minimum Repayment Amount

Credit Card Installment 1

+ Add Credit Card Installment

Total Credit Card Installments: LKR 0

Submit Application

Functional Unit 2 – Eligibility Prediction

- Uses **Logistic Regression** model (binary classification).
- Predicts: **Eligible / Not Eligible** & Eligibility Amount.
- Validated using accuracy, precision, recall.

ACCURACY OF ML MODELS

TIC REGRESSION MODEL : 79

```
y on Test Set: 0.7925
ication Report:
       precision    recall  f1-score   support

  NO       0.82      0.78      0.80       212
 YES       0.77      0.80      0.78       188

racy                           0.79       400
 avg       0.79      0.79      0.79       400
 avg       0.79      0.79      0.79       400
```

2. SUPPORT VECTOR MACHINE MODEL : 67

```
🎯 SVM Accuracy with heavy noise and corrupted labels: 0.675
📊 Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.99      0.76       211
           1       0.95      0.33      0.49       189

    accuracy                           0.68       400
   macro avg       0.79      0.66      0.63       400
weighted avg       0.78      0.68      0.63       400

🔁 Cross-Validation Accuracy (RF with heavy noise and corrupted labels): 0.5577537688442211
```

3. RANDOM FOREST MODEL : 59

```
<ipython-input-15-864267150e7f>:17: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To re
  df['Existing_Debt'] = df['Existing_Debt'].replace(' -   ', np.nan)
🎯 Random Forest Accuracy with noisy labels: 0.59
📊 Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.62      0.61       207
           1       0.58      0.56      0.57       193

    accuracy                           0.59       400
   macro avg       0.59      0.59      0.59       400
weighted avg       0.59      0.59      0.59       400

🔁 Cross-Validation Accuracy (RF with noisy labels): 0.5907638190954774
```

Functional Unit 3 – Credit Limit Estimation

- Activated only if **Eligible = Yes**.
- Uses **rule-based logic**:
  - Income minus liabilities
  - Debt-to-income ratio
- Outputs: **safe, customized loan amount**

DATA SET

Employment_Type	Years_of_Employment	Residential_Status	Dependents	Monthly_Income	Existing_Debt	Debt_to_Income_Ratio	Requested_Amount	Loan_Tenure	Interest_Rate	Max_Allowed_EMI	Loan_Status_Eligibilty
Contract	5	Own House	2	200,000.00	50,000.00	3.614457831	200,000.00	5	12	100,000.00	Yes
Contract	3	Own House	1	250,000.00	200,000.00	1.217532468	200,000.00	5	12	125,000.00	No
Contract	15	Own House	2	180,000.00	115,000.00	1.553062985	30,000.00	4	11	90,000.00	No
Contract	4	Rented House	0	220,000.00	200,000.00	1.093915097	45,000.00	6	13	110,000.00	No
Contract	19	Own House	0	200,000.00	200,000.00	0.9977550511	15,000.00	4	11	100,000.00	No
Contract	1	Rented House	1	467,040.00	460,000.00	1.012333369	15,000.00	1	8	233,520.00	No
Contract	6	Rented House	4	128,759.00	90,501.00	1.407443169	20,000.00	2	9	64,379.50	No
Contract	7	Own House	1	85,217.00	59,562.00	1.40516995	30,000.00	3	10	42,608.50	No
Contract	7	Own House	4	319,849.00	300,000.00	1.061387091	45,000.00	4	11	159,924.50	No
Contract	20	Rented House	3	100,506.00	90,194.00	1.0980982	50,000.00	5	12	50,253.00	No
Contract	16	Rented House	1	115,337.00	105,150.00	1.081622382	60,000.00	6	13	57,668.50	No
Contract	19	Rented House	2	454,932.00	450,000.00	1.007003913	75,000.00	7	14	227,466.00	No
Contract	29	Own House	4	269,092.00	166,850.00	1.587657089	99,000.00	5	12	134,546.00	No
Contract	9	Own House	3	214,000.00	213,585.00	0.9793975637	100,000.00	2	9	107,000.00	No
Contract	19	Rented House	1	130,924.00	99,903.00	1.202207469	300,000.00	4	11	65,462.00	No
Contract	10	Own House	3	140,953.00	101,100.00	1.316087768	200,000.00	4	11	70,476.50	No
Contract	13	Own House	1	254,000.00	253,297.00	0.9968981031	56,000.00	5	12	127,000.00	No
Contract	26	Rented House	0	120,091.00	110,000.00	1.049968306	89,000.00	2	9	60,045.50	No
Contract	7	Rented House	0	68,133.00	40,343.00	1.534768995	45,000.00	1	8	34,066.50	No
Contract	5	Own House	4	25,000.00	22,819.00	1.082164081	12,000.00	7	14	12,500.00	No
Contract	14	Own House	0	293,946.00	280,000.00	1.041376949	85,000.00	5	12	146,973.00	No
Contract	29	Own House	0	82,000.00	79,000.00	1.007257577	49,000.00	2	9	41,000.00	No
Contract	9	Own House	2	309,396.00	308,000.00	0.9612750885	154,000.00	1	8	154,698.00	No
Contract	17	Own House	2	478,466.00	470,000.00	1.008932375	47,000.00	1	8	239,233.00	No
Contract	10	Own House	4	158,686.00	150,000.00	1.049510582	45,000.00	5	12	79,343.00	No
Contract	25	Own House	1	413,339.00	400,000.00	1.029914452	50,000.00	5	12	206,669.50	No
Contract	15	Rented House	3	314,919.00	312,000.00	1.005103409	56,000.00	7	14	157,459.50	No
Contract	18	Own House	4	187,023.00	180,000.00	1.023661741	90,000.00	4	11	93,511.50	No
Contract	16	Own House	4	261,671.00	260,000.00	0.9785141789	300,000.00	6	13	130,835.50	No
Contract	22	Own House	0	229,442.00	250,000.00	0.9150229312	25,000.00	4	11	114,721.00	No
Contract	20	Rented House	2	305,541.00	197,015.00	1.412155386	215,000.00	1	8	152,770.50	No
Contract	4	Own House	1	412,176.00	410,000.00	0.9813013356	204,000.00	2	9	206,088.00	No
Contract	24	Own House	4	266,427.00	265,000.00	0.9938730636	85,000.00	3	10	133,213.50	No
Contract	15	Own House	2	90,000.00	96,000.00	0.9244992296	45,000.00	4	11	45,000.00	No
Contract	17	Own House	0	450,047.00	440,000.00	1.016794945	98,000.00	5	12	225,023.50	No
Contract	17	Rented House	3	440,589.00	440,000.00	0.9988132279	45,000.00	6	13	220,294.50	No

Debt-to-Income Ratio (DTI): The Debt-to-Income (DTI) ratio is a crucial financial metric that measures the borrower's ability to manage their existing debts relative to their income [11]. It was calculated by dividing the sum of the borrower's monthly debt payments (computed as loan_amnt * loan_int_rate / 12) by their monthly income (person_income / 12). A lower DTI ratio indicates that the borrower has lower debt obligations relative to their income, making them more likely to meet their loan repayment obligations.

J. Philip, *Comparative Analysis of Loan Prediction Models with Imbalanced Data and Impact of Loan Eligibility Metrics*, M.Sc. thesis, Dublin Business School, Ireland, 2023. [Online]. Available: https://esource.dbs.ie/server/api/core/bitstreams/732f870c-9671-4e11-9afc-5d2f168fde01/content

Functional Unit 4 – EMI Calculation

- Calculates repayment plan based on:
  - Loan amount
  - Tenure options (e.g., 12-74 months)
  - Interest rate
- Generates: **Monthly EMI**, interest paid, total repayment.

OAN CALCULATIONS

```
const calculateEligibleLoanAmount = (
  maxEMI: number,
  interestRate: number,
  tenure: number,
  MonthlyIncome: number,
  totalCreditCardInstallments: number,
  totalLoanInstallments: number
) => {
  const amountOfIncomes = (MonthlyIncome * 50) / 100;
  const totalDebt =
    amountOfIncomes - (totalCreditCardInstallments + totalLoanInstallments);

  return totalDebt * tenure;
};
```

MONTHLY REPAYMENT CALCULATION

```
const monthlyInstallment =
  (enteredAmount *
    monthlyInterestRate *
    Math.pow(1 + monthlyInterestRate, numberOfPayments)) /
  (Math.pow(1 + monthlyInterestRate, numberOfPayments) - 1);
```

Output & Integration

- User sees:
    - Eligibility decision
    - Loan amount
    - EMI plan (Repayment Plan)
- Delivered via simple **web dashboard** (Flask-based).
- All modules tested independently + as integrated system.

DESIGN FEATURES

Design Feature	Description	Benefit
User-Friendly Interface	Simple, intuitive UI for easy data entry and navigation	Reduces user errors and improves experience
Modular Architecture	Clear separation of components	Facilitates scalability
Robust Validation	Comprehensive input checks before processing	Ensures data accuracy and integrity
Secure Authentication	OTP verification for login and transactions	Enhances system security and prevents fraud
Efficient Prediction Model	Logistic regression for real-time eligibility decisions	Provides transparent and reliable loan approvals
Error Handling	Clear error messages and system recovery mechanisms	Improves reliability and user trust

```
try {
  const userDoc = await getDoc(doc(firestore, "user", storedUserId));
  if (userDoc.exists()) {
    setAccountName(userDoc.data().accountName || "");
    console.log(accountName);
  }
} catch (error) {
  console.error("Error fetching user data:", error);
}
};
fetchUserData();
```

TANDARDS AND BEST PRACTICES

Aspect	Best Practice	Implementation
Financial Calculations	Use of standard loan interest and repayment formulas	Correct and tested calculation formulas in the system
Input Validation	Validations for user input accuracy and completeness	Validation rules for amount, income, and existing debts
Authentication	OTP (One-Time Password) verification for user security	OTP generation and verification integrated during login
Eligibility Prediction	Logistic Regression model for credit eligibility	Model training with prediction results
Security Best Practices	Secure data handling and authentication	Encrypted OTP, secure API calls, and user data protection
Algorithm Transparency	Use of interpretable logistic regression algorithm	Clear explanation of model variables

PROFESSIONAL AND LEGAL SECURITY

Category	Identification	How It Is working in system
Professional	Need for accurate, reliable system development	Followed industry best practices in coding and ML implementation
Legal	Compliance with financial and data laws	Implemented secure OTP verification
Social	Fairness and non-discrimination in lending	Used transparent logistic regression model
Security	Protecting sensitive user data and preventing fraud	Secure API, OTP multi-factor authentication
Ethical	User privacy, consent, and transparency	Clear data usage

NON-FUNCTIONAL REQUIREMENTS

Non-Functional Requirement	
Performance	Fast response and real-time eligibility prediction
Reliability	Consistent system operation with minimal errors
Usability	Easy and intuitive user interface
Security	Protection of user data and secure authentication
Scalability	Ability to handle increased users and features in the future
Maintainability	Easy to update, debug, and enhance

P.R.I.PRAVEA N

T2[illegible]91060

Introductio
n

BACKGROUND

RESEARCH PROBLEM

OBJECTIVES

OUTCOME AFTER PP1

Provide Financial Literacy And Appropriate Financing Practices Among Customers

BACKGROUND

- What is financial counselling ?
- Why we are focusing on this topic
- Why we are doing to Provide financial literacy and appropriate financing practices

OBJECTIVES

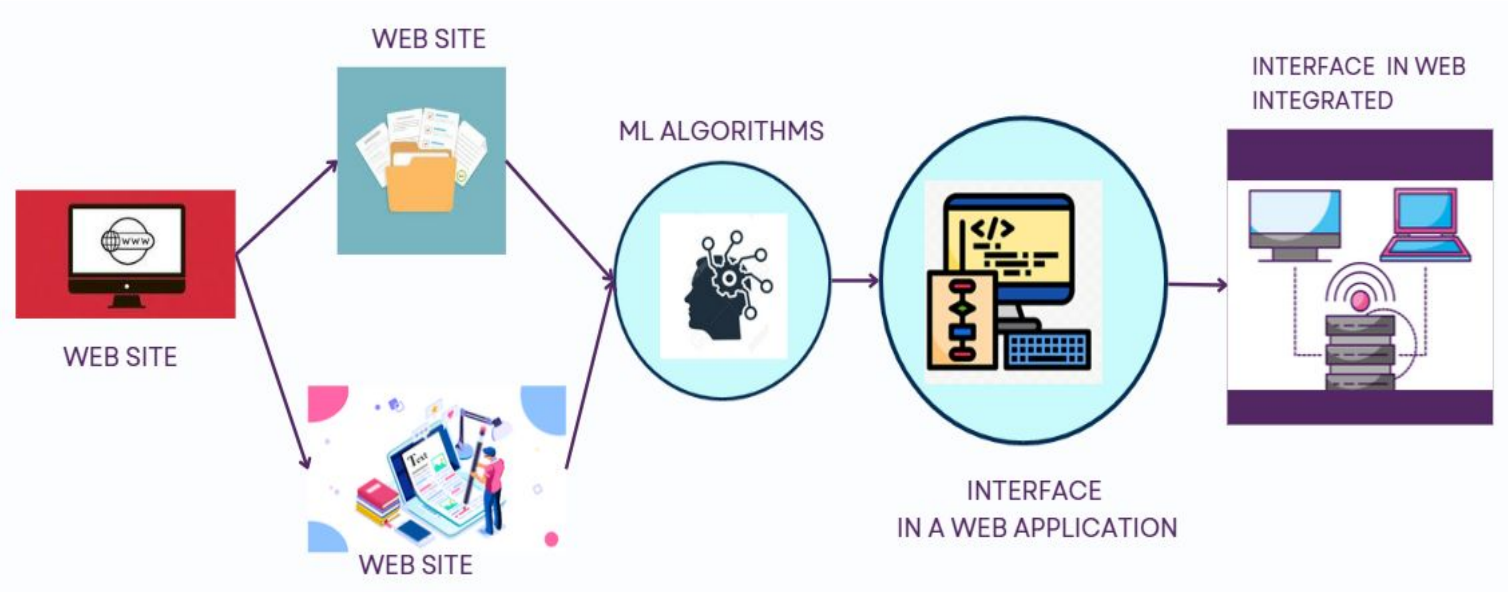
Provides suitable prediction to fulfill the gap of the Provide financial literacy and appropriate financing to the customers

SUB MAIN

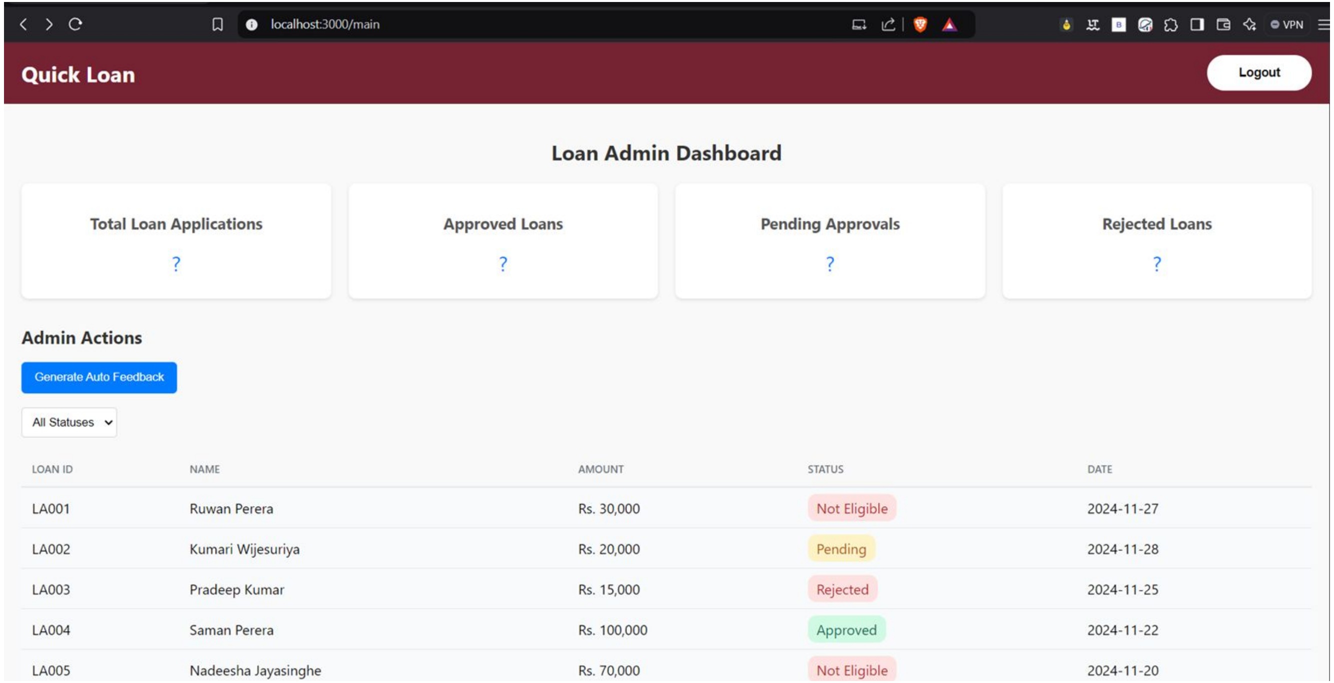
Integrate KYC Information to Customize Financial Advice

Facilitate Financial Goal Setting and Strategic Planning

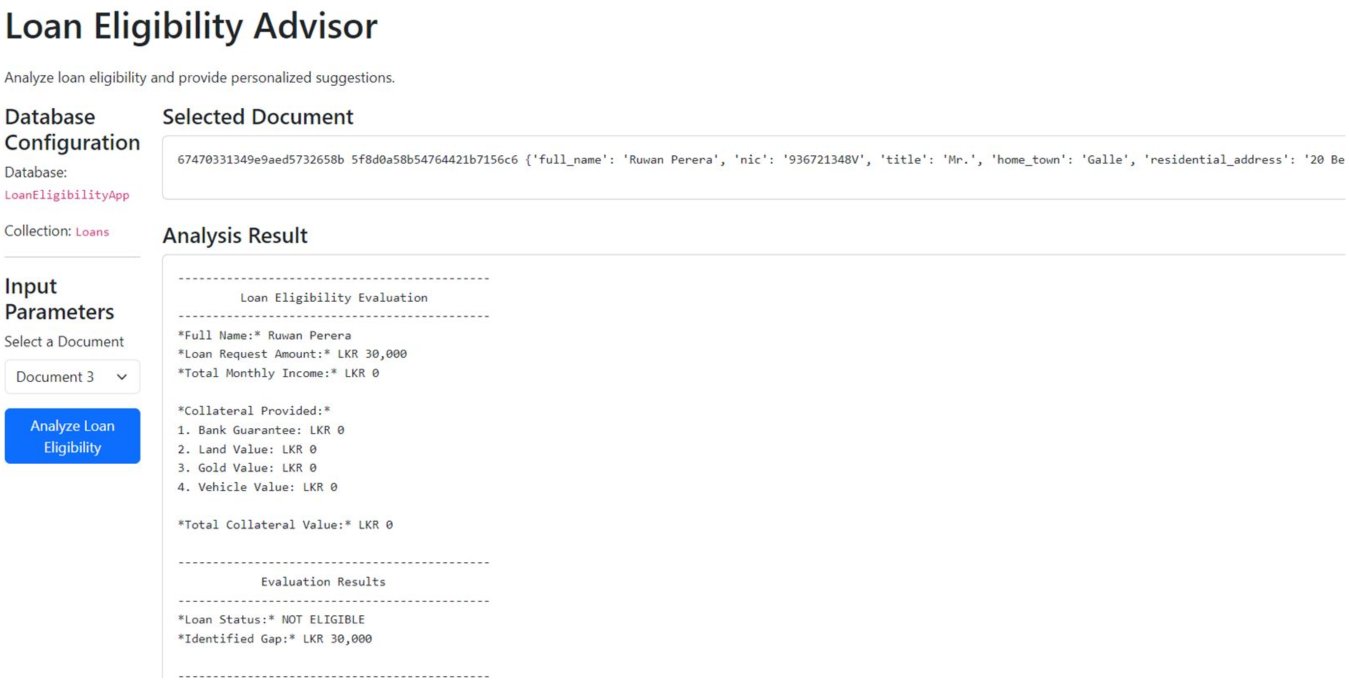
A Picture Of Methodology



Done Upto PP1 with LLM

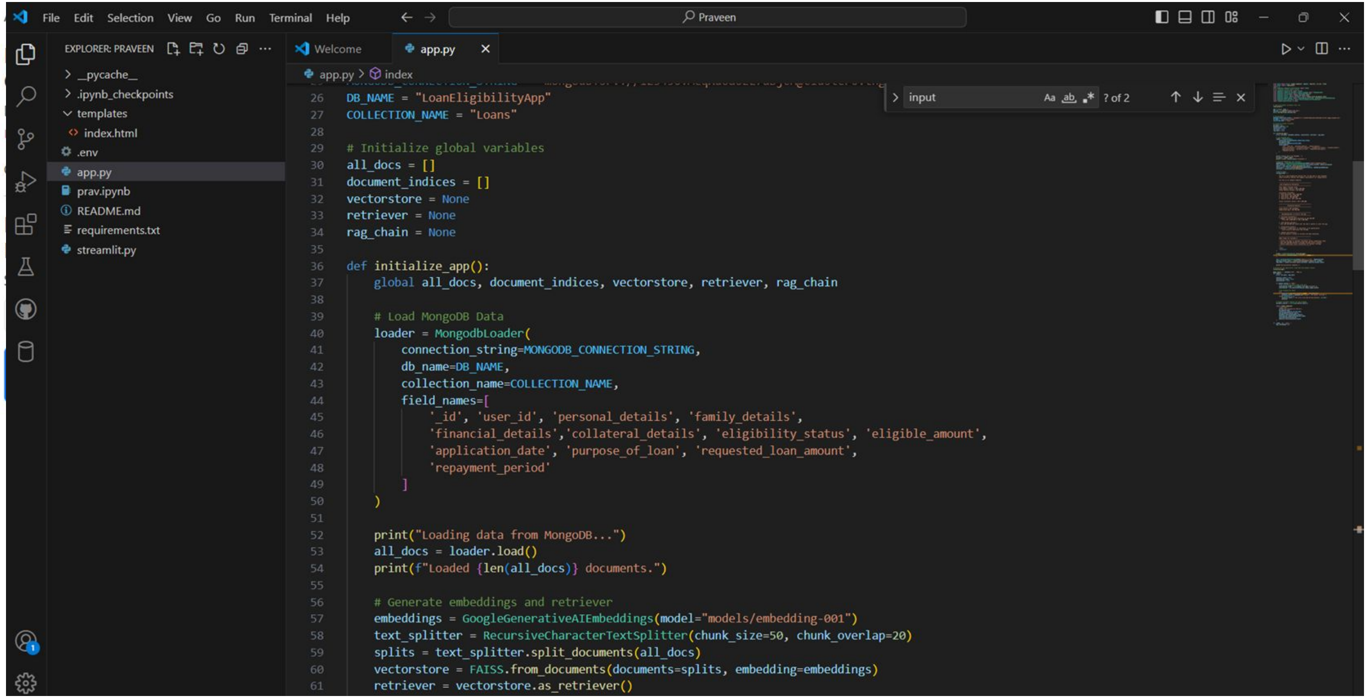


Done Upto PP1 with LLM



Done Upto PP1 with LLM

Loan Eligibility Advisor



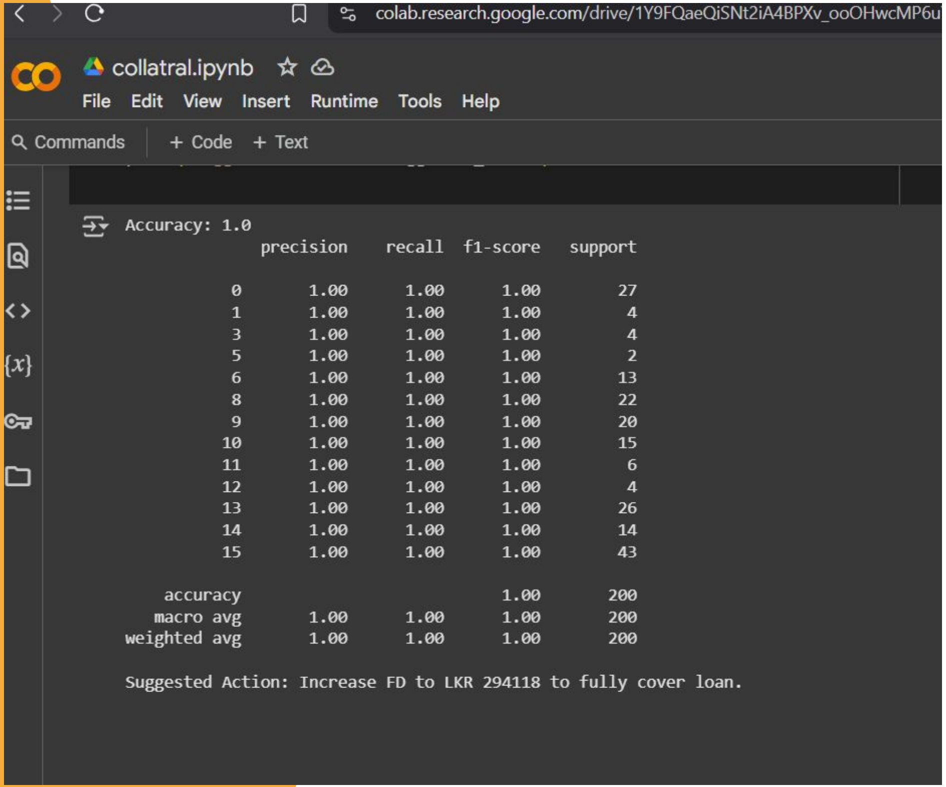
Collateral Info and Allocation

Collateral Type	Loan Eligibility
Fixed Deposit	85%
Insurance Policy (Surrender Value)	70%
EPF/Pension Fund (Loan Against Savings)	75%
Govt. Bonds & T-Bills	90%
Inventory/Stock (Small Business)	50%

Dataset

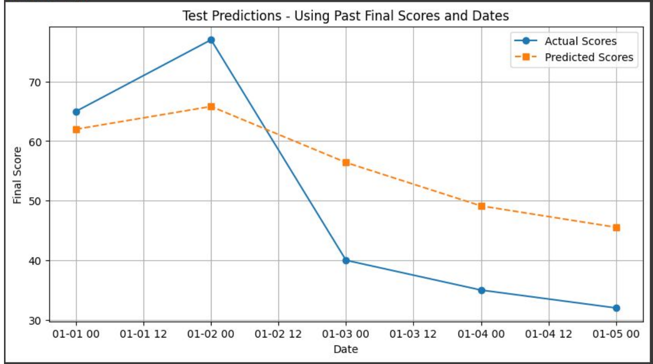
ID	Collateral Type	Asset Value (LKR)	Loan Eligibility (%)	Max Loan Amount (LKR)	Requested Loan Amount (LKR)	Gap (LKR)	Suggestion 1
1	Government Bonds & Treasury Bills (T-Bills)	100000	80	80000	100000	20000	If a T-Bill is maturing soon, liquidate a portion to cover the shortfall.
2	Government Bonds & Treasury Bills (T-Bills)	150000	80	120000	200000	80000	If a T-Bill is maturing soon, liquidate a portion to cover the shortfall.
3	Inventory or Stock (for Small Businesses)	200000	50	100000	250000	150000	Use part of business revenue to finance the shortfall.
4	Fixed Deposit	100000	85	85000	200000	115000	Increase FD to LKR 235294 to fully cover loan.
5	Insurance	250000	70	175000	100000	0	Request Insurance as Collateral Type for 70%
6	Insurance	150000	70	105000	150000	45000	Pledge additional assets (Fixed Deposit, EPF, Bonds) to cover shortfall.
7	Fixed Deposit	100000	85	85000	250000	165000	Increase FD to LKR 294118 to fully cover loan.
8	Employee Provident Fund (EPF)	100000	75	75000	250000	175000	Request lender to disburse partial loan first and apply for a top-up later.
9	Government Bonds & Treasury Bills (T-Bills)	200000	80	160000	250000	90000	If a T-Bill is maturing soon, liquidate a portion to cover the shortfall.
10	Insurance	150000	70	105000	100000	0	Request Insurance as Collateral Type for 70%
11	Fixed Deposit	250000	85	212500	200000	0	Request Fixed Deposit as Collateral Type for 85%
12	Fixed Deposit	100000	85	85000	200000	115000	Increase FD to LKR 235294 to fully cover loan.
13	Fixed Deposit	150000	85	127500	150000	22500	Increase FD to LKR 176471 to fully cover loan.
14	Fixed Deposit	100000	85	85000	250000	165000	Increase FD to LKR 294118 to fully cover loan.
15	Government Bonds & Treasury Bills (T-Bills)	250000	80	200000	150000	0	Request Government Bonds & Treasury Bills (T-Bills) for 80%
16	Government Bonds & Treasury Bills (T-Bills)	200000	80	160000	150000	0	Request Government Bonds & Treasury Bills (T-Bills) for 80%
17	Government Bonds & Treasury Bills (T-Bills)	250000	80	200000	150000	0	Request Government Bonds & Treasury Bills (T-Bills) for 80%
18	Inventory or Stock (for Small Businesses)	250000	50	125000	250000	125000	Use part of business revenue to finance the shortfall.
19	Fixed Deposit	150000	85	127500	200000	72500	Increase FD to LKR 235294 to fully cover loan.
20	Employee Provident Fund (EPF)	150000	75	112500	150000	37500	Request lender to disburse partial loan first and apply for a top-up later.
21	Fixed Deposit	250000	85	212500	250000	37500	Increase FD to LKR 294118 to fully cover loan.
22	Insurance	150000	70	105000	200000	95000	Pledge additional assets (Fixed Deposit, EPF, Bonds) to cover shortfall.
23	Insurance	150000	70	105000	150000	45000	Pledge additional assets (Fixed Deposit, EPF, Bonds) to cover shortfall.
24	Inventory or Stock (for Small Businesses)	250000	50	125000	100000	0	Request Inventory or Stock (for Small Businesses) as Collateral Type for 50%
25	Inventory or Stock (for Small Businesses)	250000	50	125000	150000	25000	Use part of business revenue to finance the shortfall.
26	Fixed Deposit	250000	85	212500	200000	0	Request Fixed Deposit as Collateral Type for 85%

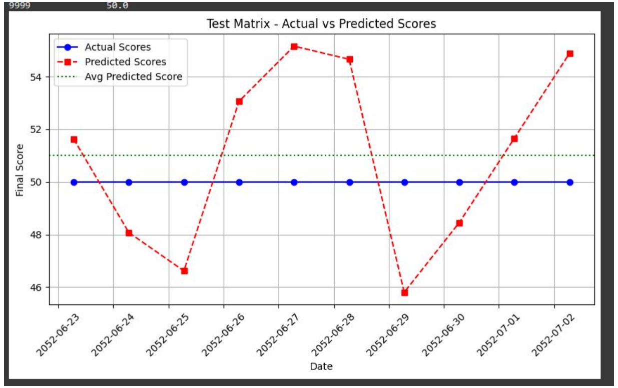
Model Evaluation on Test Data



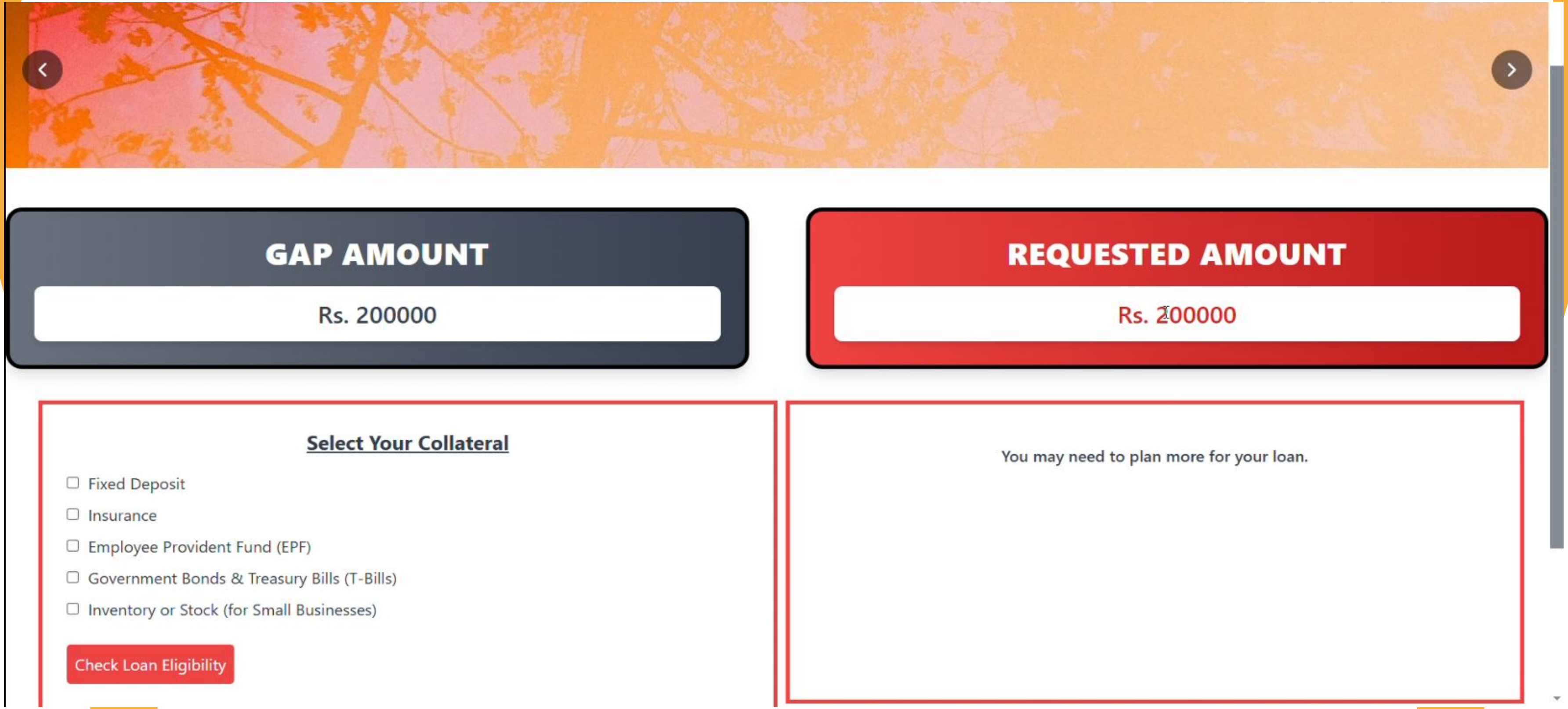

```
Accuracy: 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        27
           1       1.00      1.00      1.00         4
           3       1.00      1.00      1.00         4
           5       1.00      1.00      1.00         2
           6       1.00      1.00      1.00        13
           8       1.00      1.00      1.00        22
           9       1.00      1.00      1.00        20
          10       1.00      1.00      1.00        15
          11       1.00      1.00      1.00         6
          12       1.00      1.00      1.00         4
          13       1.00      1.00      1.00        26
          14       1.00      1.00      1.00        14
          15       1.00      1.00      1.00        43

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200

Suggested Action: Increase FD to LKR 294118 to fully cover loan.
```

```
Loading saved Prophet model...
Loading saved XGBoost model...
Making predictions with Prophet...
Making predictions with XGBoost...

Model Evaluation on Test Data:
MAE: 11.65
MSE: 157.21
RMSE: 12.54
R2 Score: 0.51
```





Done Upto PP2 with ML



GAP AMOUNT

Rs. 200000

REQUESTED AMOUNT

Rs. 200000

Select Your Collateral

- [x] Fixed Deposit

100000

- [ ] Insurance
- [ ] Employee Provident Fund (EPF)
- [x] Government Bonds & Treasury Bills (T-Bills)

150000

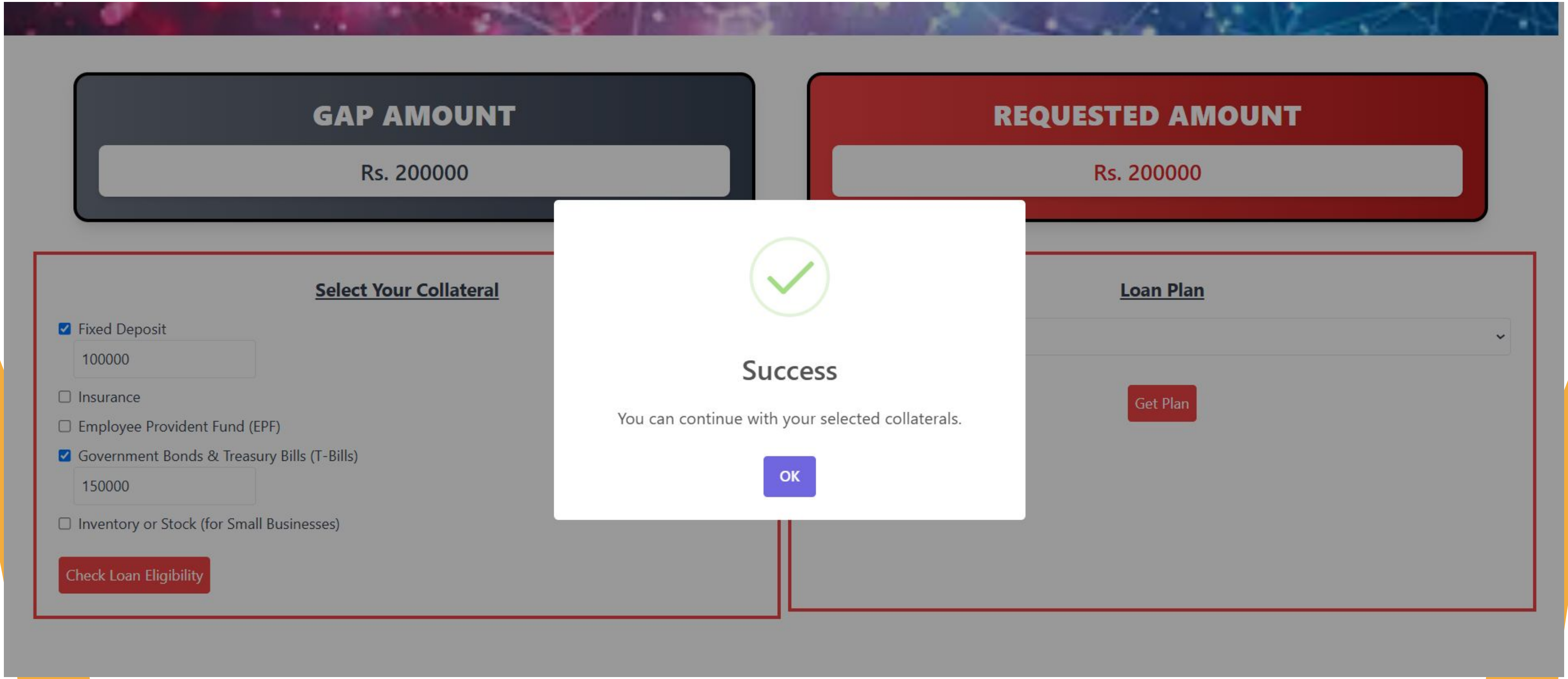
- [ ] Inventory or Stock (for Small Businesses)

Check Loan Eligibility

Loan Plan

Select Collateral

Get Plan

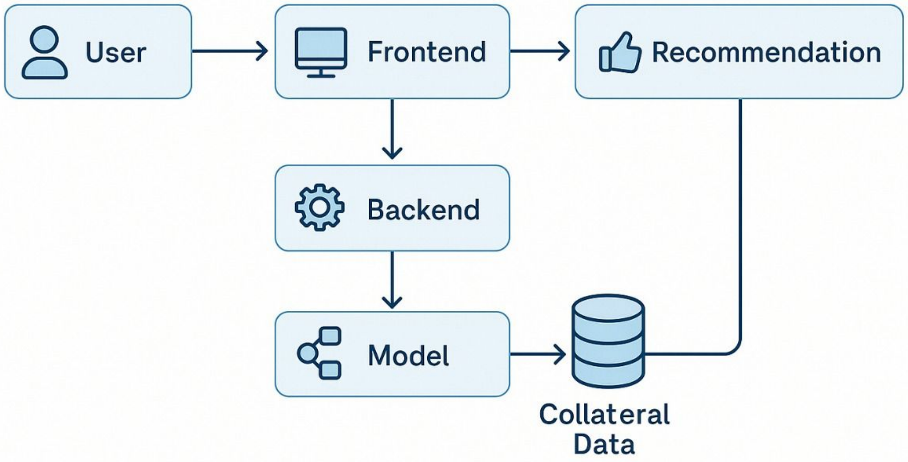

GAP AMOUNT
Rs. 200000
REQUESTED AMOUNT
Rs. 200000
Select Your Collateral
Fixed Deposit
100000
Insurance
Employee Provident Fund (EPF)
Government Bonds & Treasury Bills (T-Bills)
150000
Inventory or Stock (for Small Businesses)
Check Loan Eligibility
Success
You can continue with your selected collaterals.
OK
Loan Plan
Get Plan

Final RP

Collateral Info and Allocation

Collateral Type	Loan Eligibility
Fixed Deposit	90%
Land	70%
Govt. Bonds & T-Bills	90%
EPF/Pension Fund (Loan Against Savings))	70%

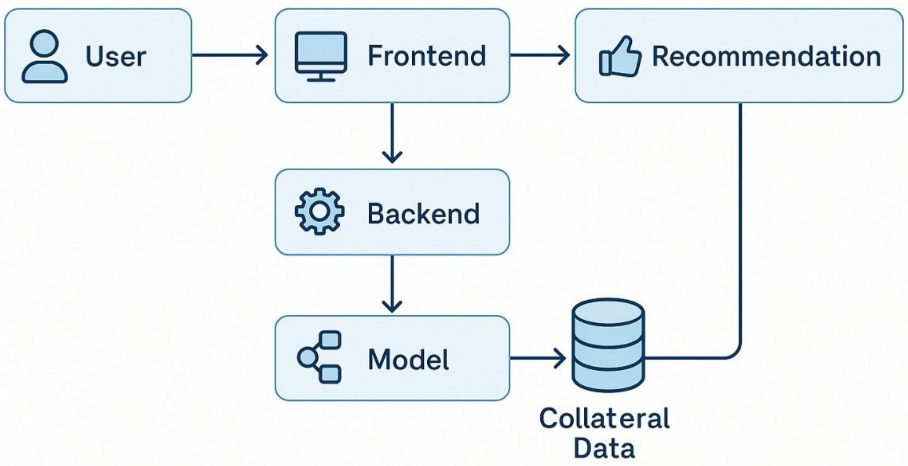
Flowchart System Architecture Diagrams



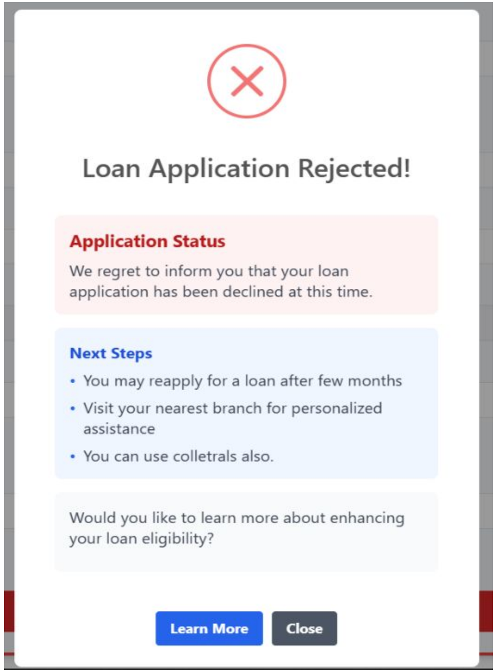
New Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	monthly_income	requested_loan_amount	fd_available	fd_value	fd_withdrawable_now	epf_available	epf_value	epf_withdrawable_now	gov_bonds_available	gov_bonds_value	gov_bonds_withdrawable_nov	land_available	land_value	land_withdrawable_now	risk_appetite	output_fd_percent	output_epf_percent	output_bond_percen	output_land_percent
2	141958	292353.1437	FALSE	0	0	FALSE	0	0	FALSE	0	0	TRUE	571430	292353.1437	Low	0	0	0	1
3	105305	86798.1303	TRUE	159503	120971.6896	TRUE	133855	73159.02155	FALSE	0	0	TRUE	179981	118253.8509	High	0.285261403	0.714738597	0	0
4	194073	179194.1201	TRUE	58555	46787.51679	TRUE	116530	79457.62129	TRUE	210551	179194.1201	FALSE	0	0	Medium	0	0	1	0
5	98953	206428.3829	TRUE	233165	206428.3829	FALSE	0	0	FALSE	0	0	FALSE	0	0	High	1	0	0	0
6	31534	25871.45436	TRUE	34300	25871.45436	TRUE	19268	11895.10429	FALSE	0	0	FALSE	0	0	Low	1	0	0	0
7	159643	101384.5449	TRUE	130151	101384.5449	FALSE	0	0	TRUE	143767	100795.6796	FALSE	0	0	High	1	0	0	0
8	59504	101091.9807	FALSE	0	0	FALSE	0	0	TRUE	114488	101091.9807	TRUE	844824	527727.4433	Medium	0	0	1	0
9	72256	248411.8825	TRUE	283109	248411.8825	FALSE	0	0	FALSE	0	0	FALSE	0	0	Low	1	0	0	0
10	159182	140009.5424	FALSE	0	0	FALSE	0	0	TRUE	131626	105899.2463	TRUE	268126	140009.5424	Low	0	0	0	1
11	70015	149355.0267	TRUE	212982	166568.2351	TRUE	259002	140019.6527	FALSE	0	0	TRUE	374767	232270.5029	High	0.351633614	0.648366386	0	0
12	194013	49474.33166	TRUE	171152	127608.7118	TRUE	147848	98112.36942	TRUE	65016	49474.33166	FALSE	0	0	Medium	0	0	1	0
13	41959	61555.68875	TRUE	175838	129004.9921	TRUE	71087	42919.35379	TRUE	78840	61555.68875	TRUE	887437	616197.5752	Medium	0	0	1	0
14	85318	480576.971	FALSE	0	0	TRUE	235281	141294.6647	FALSE	0	0	TRUE	948613	480576.971	Low	0	0	0	1
15	75680	170895.6786	TRUE	204806	170895.6786	TRUE	108506	62274.94528	TRUE	297486	256024.7667	FALSE	0	0	Low	1	0	0	0
16	132547	545734.8512	FALSE	0	0	FALSE	0	0	TRUE	94896	73972.7989	TRUE	793974	545734.8512	Low	0	0	0	1
17	127512	103700.2294	TRUE	127796	112016.6788	TRUE	184088	102311.3739	TRUE	183714	155273.4443	FALSE	0	0	High	0.143102712	0.856897288	0	0
18	199426	222445.7575	TRUE	129121	102925.2867	FALSE	0	0	TRUE	267186	221869.1842	TRUE	686584	367449.4927	Low	0.548168115	0	0	0.451831885
19	39830	126537.9916	TRUE	160810	126537.9916	FALSE	0	0	TRUE	326189	270417.4241	FALSE	0	0	High	1	0	0	0
20	127455	175454.0639	TRUE	381760	270561.2352	FALSE	0	0	TRUE	197563	175454.0639	TRUE	355628	197697.897	Medium	0	0	1	0
21	198274	408704.9284	TRUE	469451	408704.9284	FALSE	0	0	TRUE	75726	57638.0425	FALSE	0	0	High	1	0	0	0
22	138015	150999.7495	TRUE	455101	327415.7239	TRUE	82991	52635.27432	FALSE	0	0	FALSE	0	0	High	0.357974795	0.642025205	0	0
23	130627	27462.77835	TRUE	97922	75164.83235	TRUE	42097	21839.77154	TRUE	152038	126861.3019	TRUE	168834	100138.7552	High	0.105447734	0.894552266	0	0
24	42677	194301.0247	FALSE	0	0	TRUE	123632	82286.22014	TRUE	261451	194301.0247	TRUE	426836	252453.4465	Medium	0	0	1	0
25	183165	32467.77503	TRUE	355757	251951.9113	TRUE	83847	46465.19435	TRUE	38251	32467.77503	FALSE	0	0	Medium	0	0	1	0
26	87215	184998.5249	FALSE	0	0	TRUE	61293	41133.24912	TRUE	201475	148097.4332	TRUE	356063	184998.5249	Low	0	0	0	1
27	114758	324842.456	TRUE	370260	324842.456	TRUE	136131	78264.77797	TRUE	33289	29516.93225	FALSE	0	0	Low	1	0	0	0
28	31938	40106.66408	TRUE	48304	40106.66408	FALSE	0	0	FALSE	0	0	FALSE	0	0	High	1	0	0	0
29	171836	22380.19634	TRUE	37350	26676.7792	TRUE	35351	20400.33119	TRUE	347497	266921.1032	FALSE	0	0	High	0.315443568	0.684556432	0	0
30	58756	142175.8221	TRUE	80640	68969.98823	TRUE	229798	157676.2344	FALSE	0	0	TRUE	111476	61057.56471	High	0.17473868	0.82526132	0	0
31	132856	344622.4801	TRUE	453733	344622.4801	FALSE	0	0	FALSE	0	0	FALSE	0	0	High	1	0	0	0
32	198352	405859.2377	TRUE	485853	405859.2377	FALSE	0	0	TRUE	326242	260538.5353	TRUE	189407	111691.2442	High	1	0	0	0
33	146174	184450.8688	TRUE	213339	178694.0269	FALSE	0	0	TRUE	223945	184450.8688	TRUE	258124	172348.4017	Medium	0	0	1	0
34	191334	121854.9586	TRUE	258683	209968.0373	TRUE	128874	70847.81534	FALSE	0	0	FALSE	0	0	High	0.366640755	0.633359245	0	0
35	64482	407194.7053	TRUE	416332	359943.1906	TRUE	49298	30400.78268	TRUE	71629	55873.81312	TRUE	805195	448608.2887	Low	0.467078752	0	0	0.532921248
36	134821	439155.5847	TRUE	496261	439155.5847	FALSE	0	0	TRUE	217611	155539.4888	FALSE	0	0	High	1	0	0	0
37	181426	146212.0306	TRUE	455952	376393.6723	TRUE	209938	142806.4621	TRUE	165576	146212.0306	TRUE	936089	645978.4471	Medium	0	0	1	0
38	145899	266136.322	TRUE	356809	266136.322	TRUE	137016	75631.45808	TRUE	188677	133293.3553	FALSE	0	0	Low	1	0	0	0
39	77043	55929.93313	TRUE	66178	55929.93313	TRUE	60990	41707.11558	TRUE	330285	288409.0823	FALSE	0	0	Low	1	0	0	0

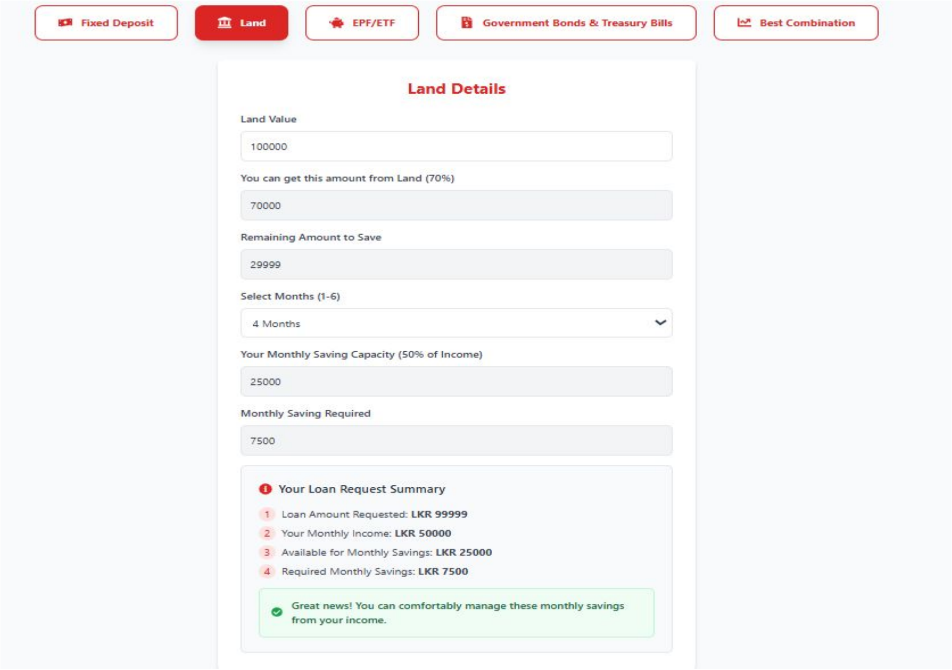
Flowchart System Architecture Diagrams



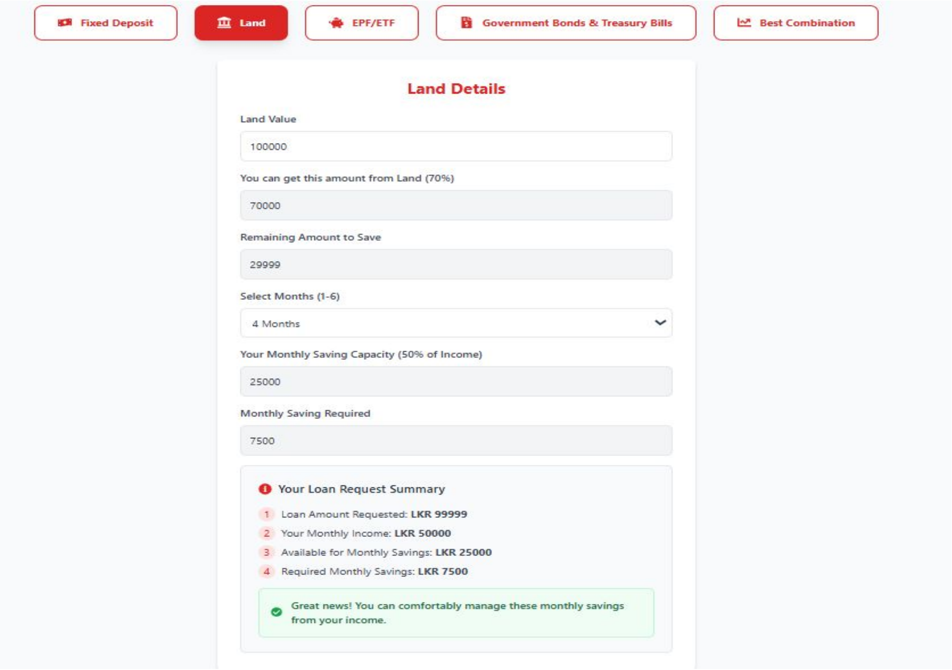
Proof of Module Running



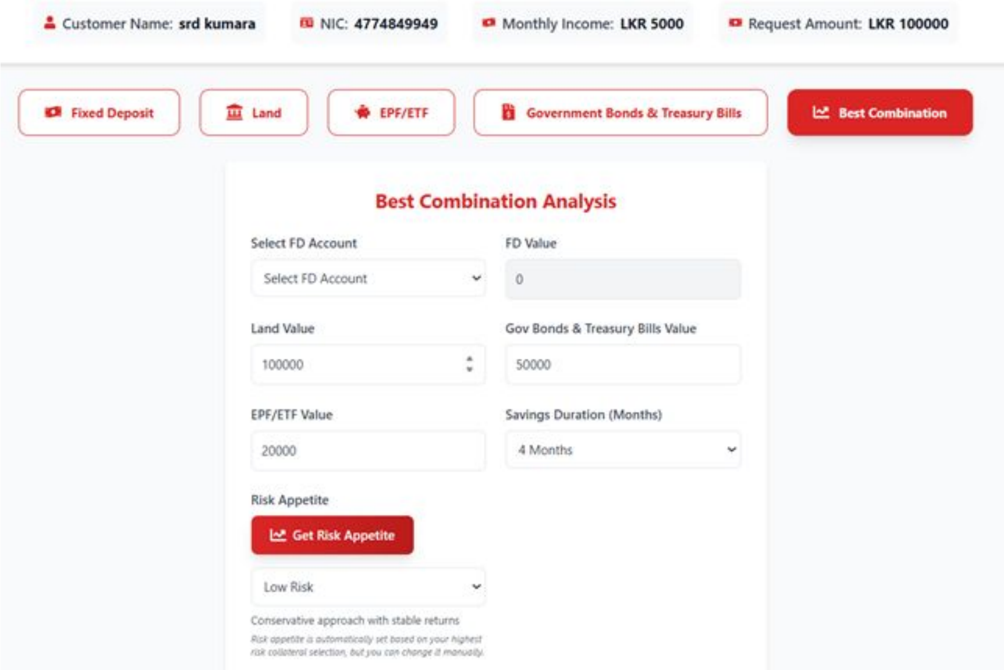
Proof of Module Running



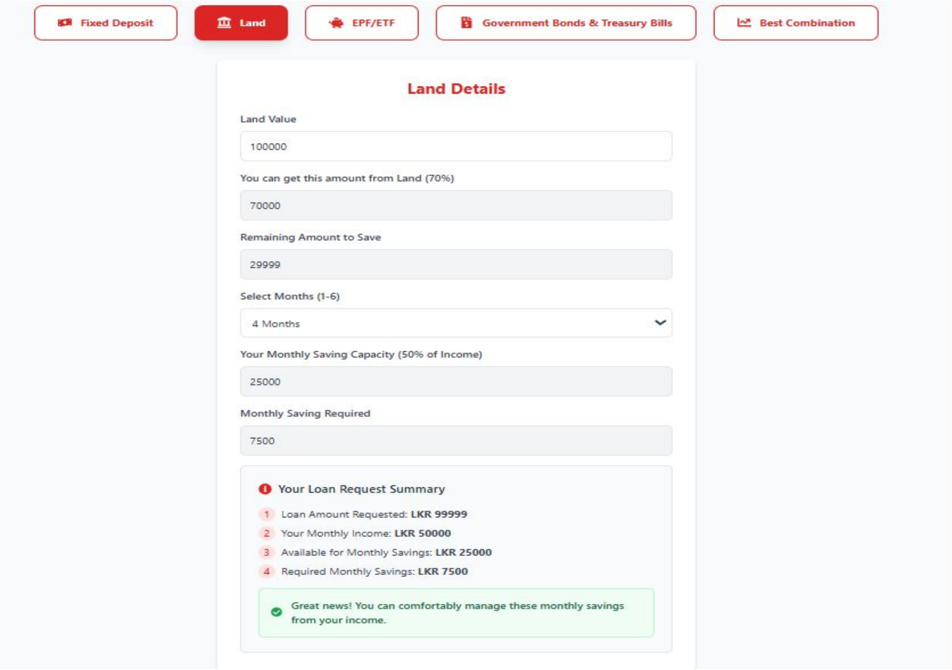
Proof of Module Running



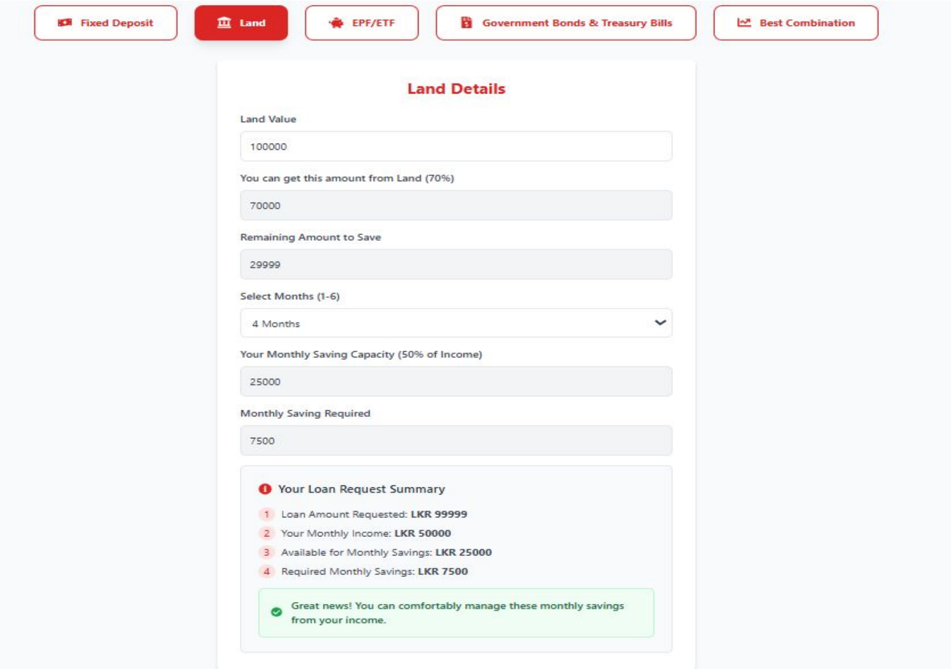
Proof of Module Running



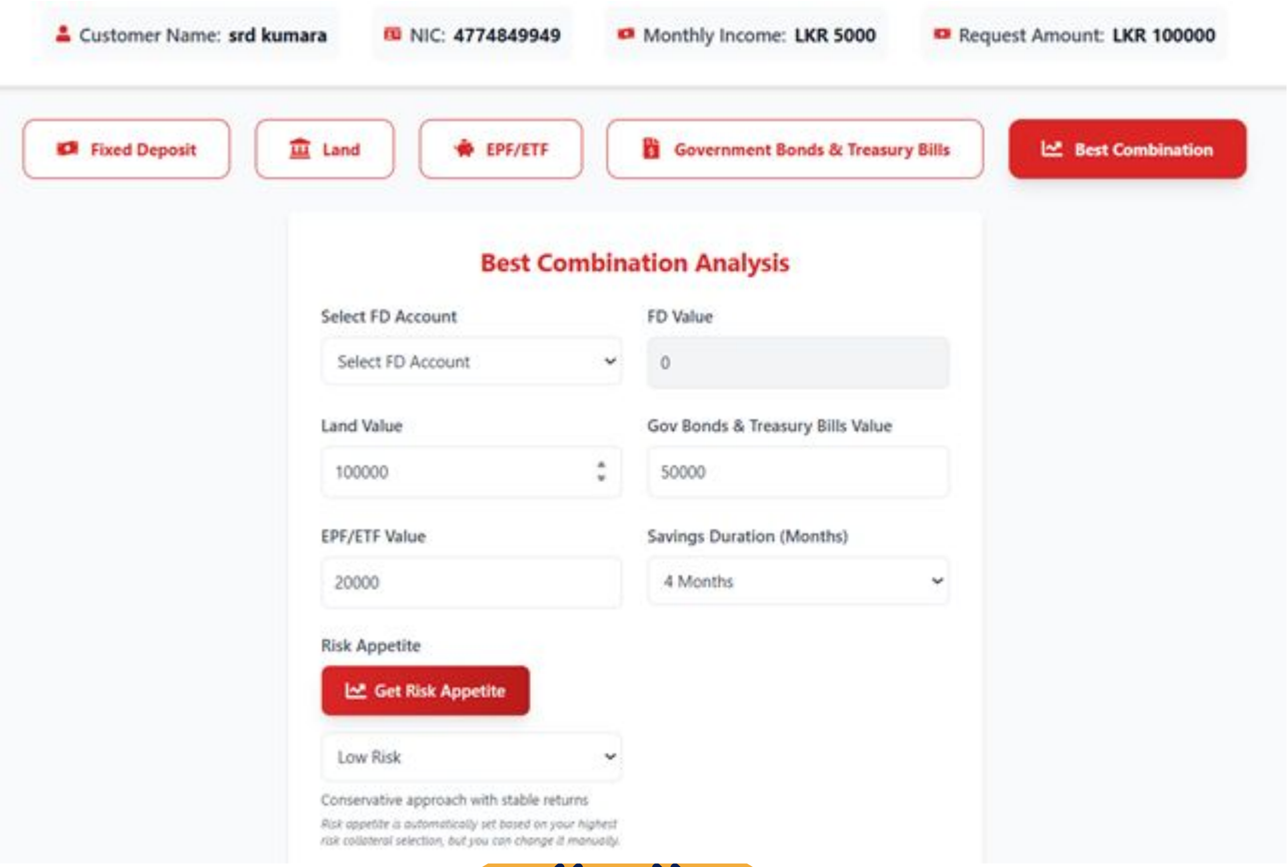
Proof of Module Running



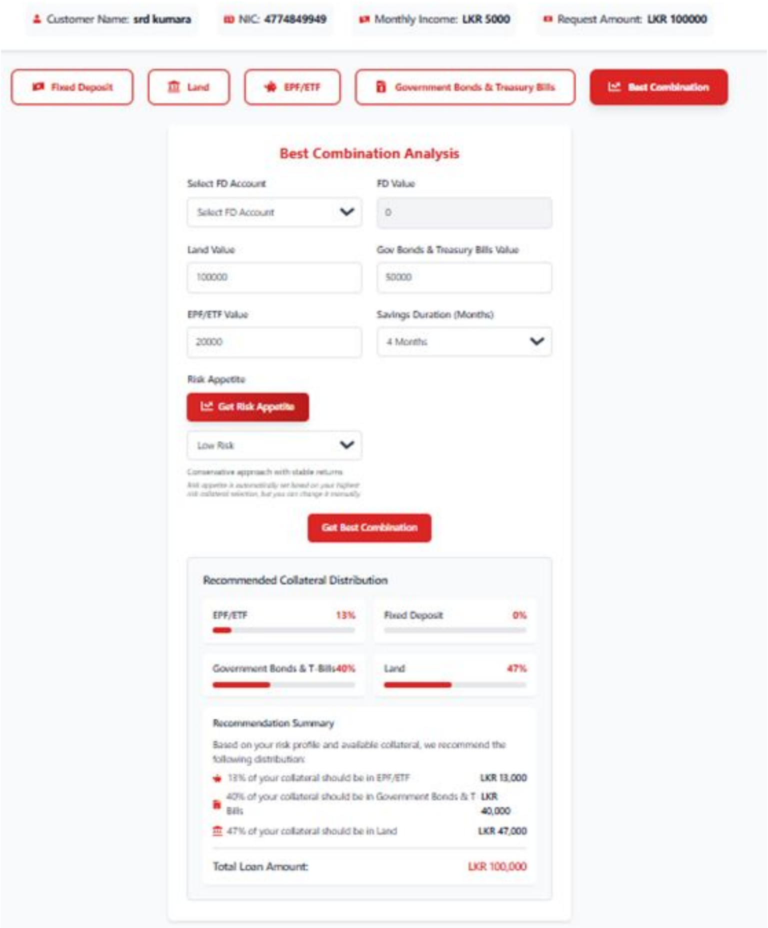
Proof of Module Running



Proof of Module Running



Proof of Module Running



Collateral based Alternative Financing Suggestion Model

```
print('Mean squared error using Neural Network: ', best_val_loss)
print('Mean absolute error Using Neural Network: ', best_val_mae)
print('R2  Using Neural Network: ', best_val_r2)
```

```
Mean squared error using Neural Network:  0.003955541652333482
Mean absolute error Using Neural Network:  [0.03393894 0.03057644 0.02362445 0.03611839]
R2  Using Neural Network:  [0.91026809 0.95693616 0.96456356 0.96810703]
```

Final RP BACKEND

```jsx
const ReasonPage = () => {
        <label className="block text-gray-700 font-medium mb-2">
          Land Value
        </label>
        <input
          type="number"
          value={landValue}
          onChange={(e) => handleLandValueChange(e.target.value)}
          className="w-full p-3 border rounded-lg focus:ring-2 focus:ring-blue-400"
          placeholder="Enter land value"
        />
      </div>

      <div>
        <label className="block text-gray-700 font-medium mb-2">
          You can get this amount from Land (70%)
        </label>
        <input
          type="text"
          value={landLoanAmount}
          readOnly
          className="w-full p-3 border rounded-lg bg-gray-100"
        />
      </div>

      <div>
        <label className="block text-gray-700 font-medium mb-2">
          Remaining Amount to Save
        </label>
        <input
          type="text"
          value={landRemainingAmount}
          readOnly
          className="w-full p-3 border rounded-lg bg-gray-100"
        />
      </div>
```

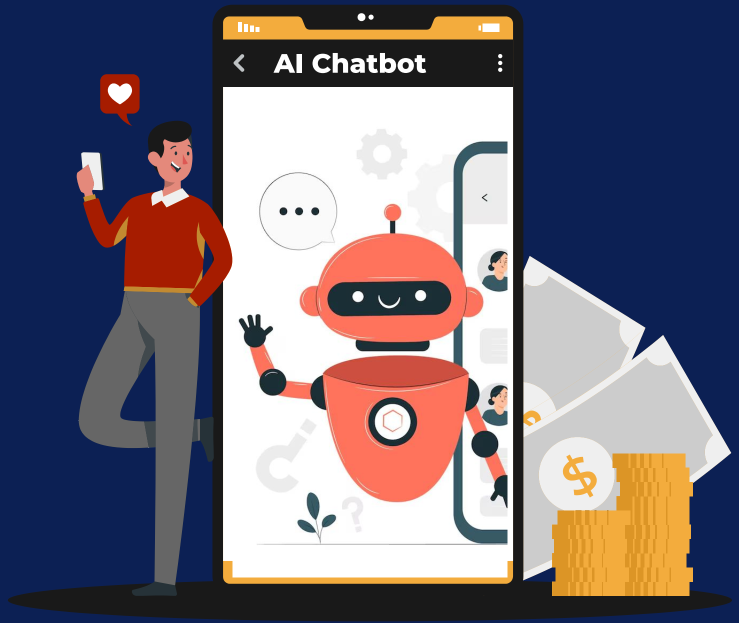
```jsx
import FaMoneyBillWave
  >
    <FaMoneyBillWave className="mr-3 text-xl" />
    Fixed Deposit
  </button>
  <button
    onClick={() => setActiveTab("Land")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "Land"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
    <FaLandmark className="mr-3 text-xl" />
    Land
  </button>
  <button
    onClick={() => setActiveTab("EPF_ETF")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "EPF_ETF"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
    <FaPiggyBank className="mr-3 text-xl" />
    EPF/ETF
  </button>
  <button
    onClick={() => setActiveTab("T-Bills")}
    className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
      activeTab === "T-Bills"
        ? "text-white bg-red-600 shadow-lg"
        : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
    }`}
  >
      Government Bonds & Treasury Bills
    </button>
    <button
      onClick={() => setActiveTab("Best_Combination")}
      className={`flex items-center p-4 px-8 font-bold rounded-xl transition-all duration-200 ${
        activeTab === "Best_Combination"
          ? "text-white bg-red-600 shadow-lg"
          : "text-red-600 bg-white border-2 border-red-600 hover:bg-red-50"
      }`}
    >
      <FaChartLine className="mr-3 text-xl" />
      Best Combination
    </button>
  </div>

  {activeTab === "Best_Combination" && (
    <div className="max-w-2xl mx-auto bg-white p-8 rounded-lg shadow-md">
      <h2 className="text-2xl font-bold mb-6 text-center text-red-600">
        Best Combination Analysis
      </h2>

      <div className="space-y-6">
        <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
          <div>
            <label className="block text-gray-700 font-medium mb-2">
              Select FD Account
            </label>
            <select
              value={selectedFdAccount}
              onChange={(e) => {
                handleFdAccountSelect(e.target.value);
                setBestCombination((prev) => ({
                  ...prev,
                  fd_value: Number(fdAmount),
                }));
```

AI
CHATBOT

M I F HILMA

IT21142178

Introduction



BACKGROUND

RESEARCH QUESTION

RESEARCH GAP

MAIN & SUB OBJECTIVES

Background

01 Banks handle high volumes of customer inquiries daily.

02 Traditional customer is time-consuming and costly.

03 Customers expect instant responses and 24/7 availability.

04 AI chatbots help automate responses, reduce costs, and improve efficiency.

05 Many banks have chatbots, but they are rule-based with limited AI capabilities.

Research Problem

- The existing customer service systems in banks are inefficient for handling large-scale queries

- Many banking chatbots provide generic responses

- Customers face delays in getting loan details, repayment plans, and eligibility criteria.

- Bank staff spend time searching for customer data instead of getting instant AI-driven insights.

Research Gap

01 Do not offer personalized interactions for general customers, existing loan customers, and bank staff.

02 Many bank chatbots (including Seylan's) are basic rule-based systems.

03 Lack advanced AI/NLP capabilities to understand banking queries.

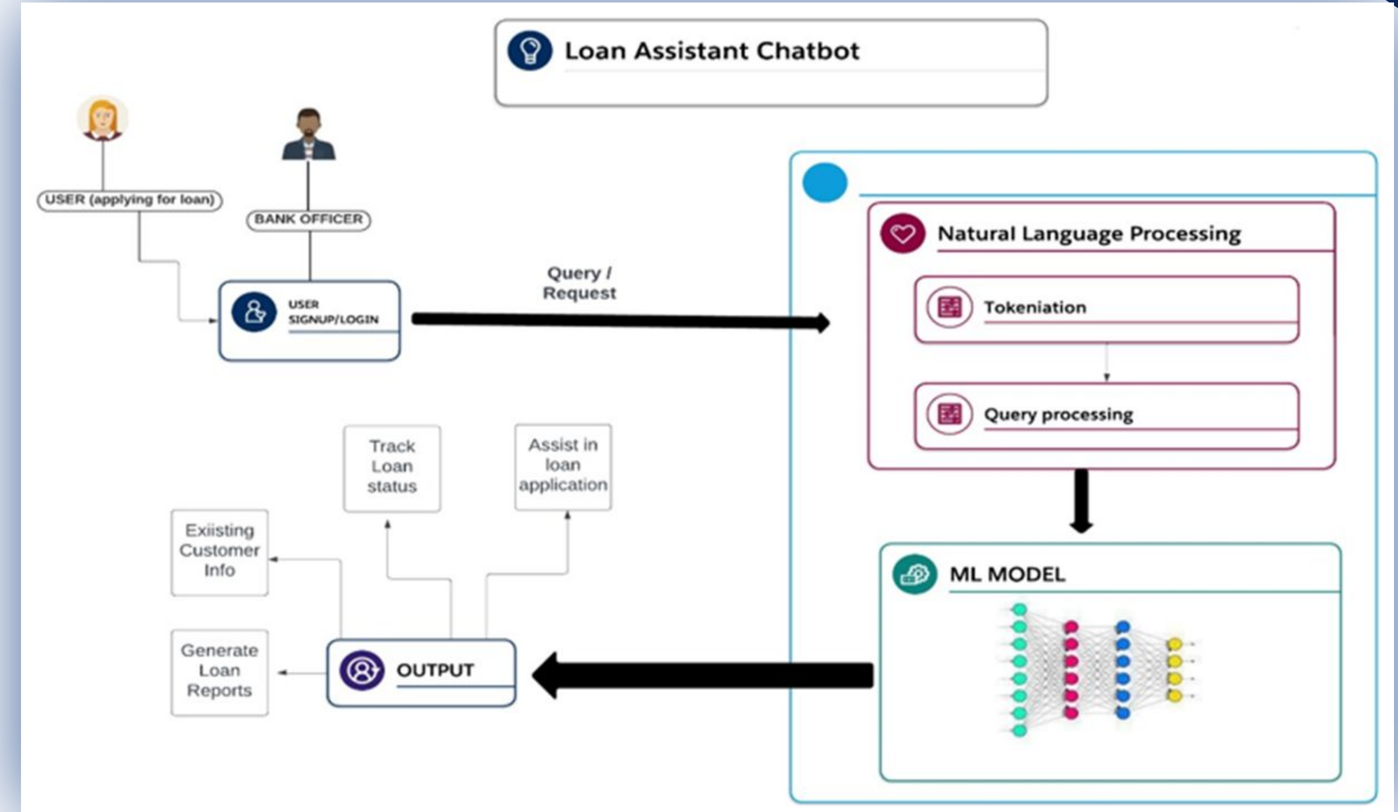
Research Objectives

Design a chatbot that serves general customers, existing customers, and bank staff.

Implement NLP and AI models to handle complex financial queries.

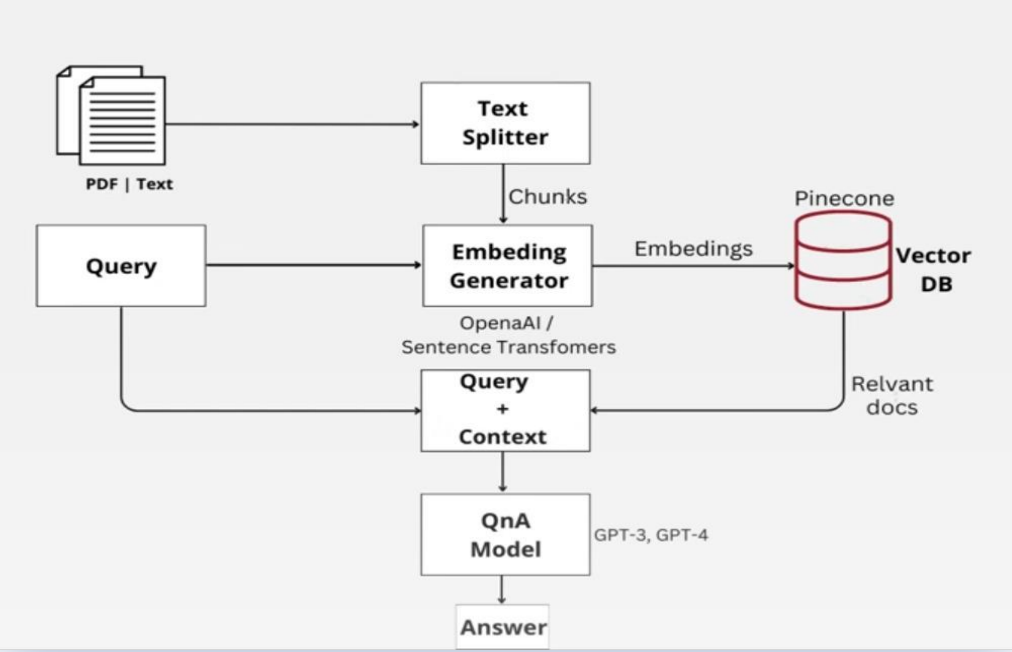
Ensure chatbot accuracy by training it with bank-specific data.

SYSTEM DIAGRAM

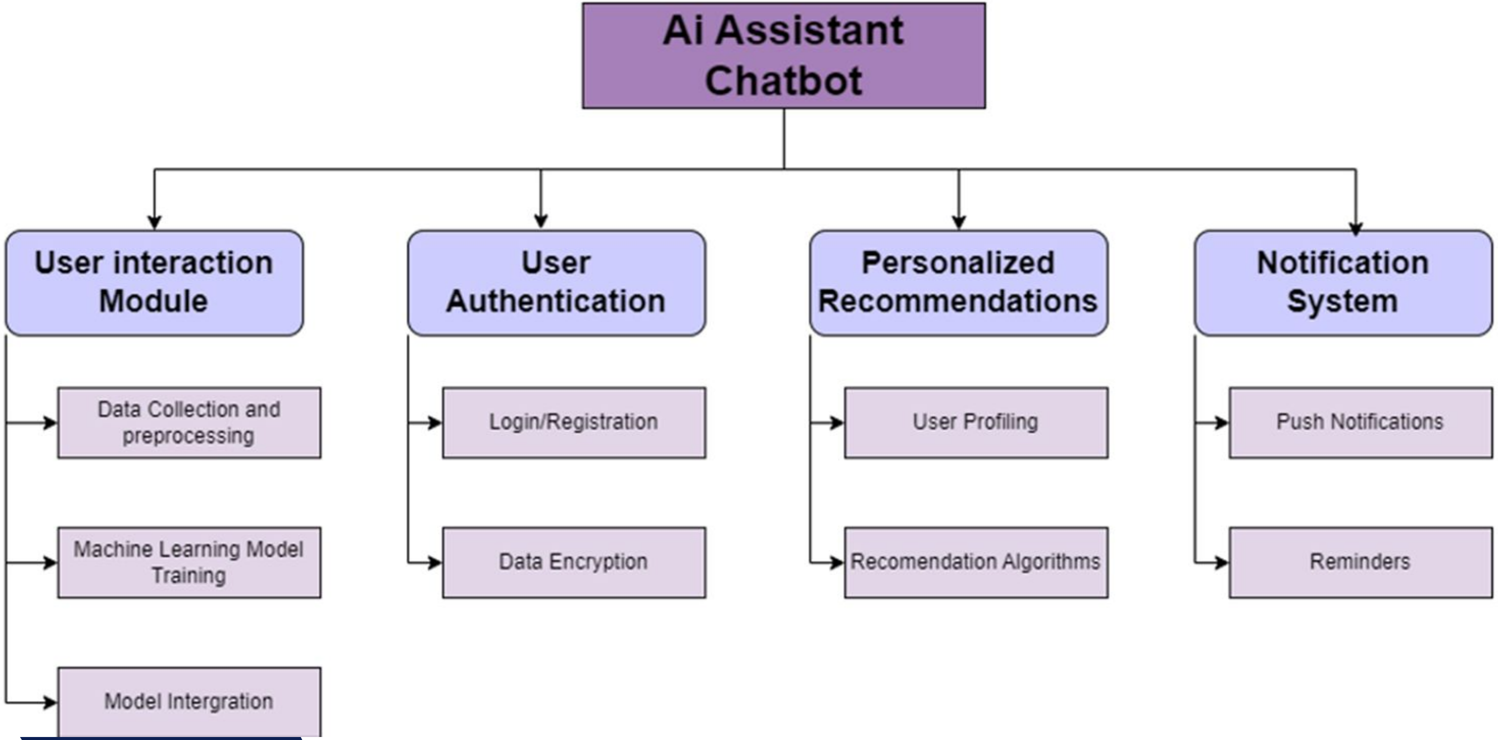


SYSTEM ARCHITECTURE

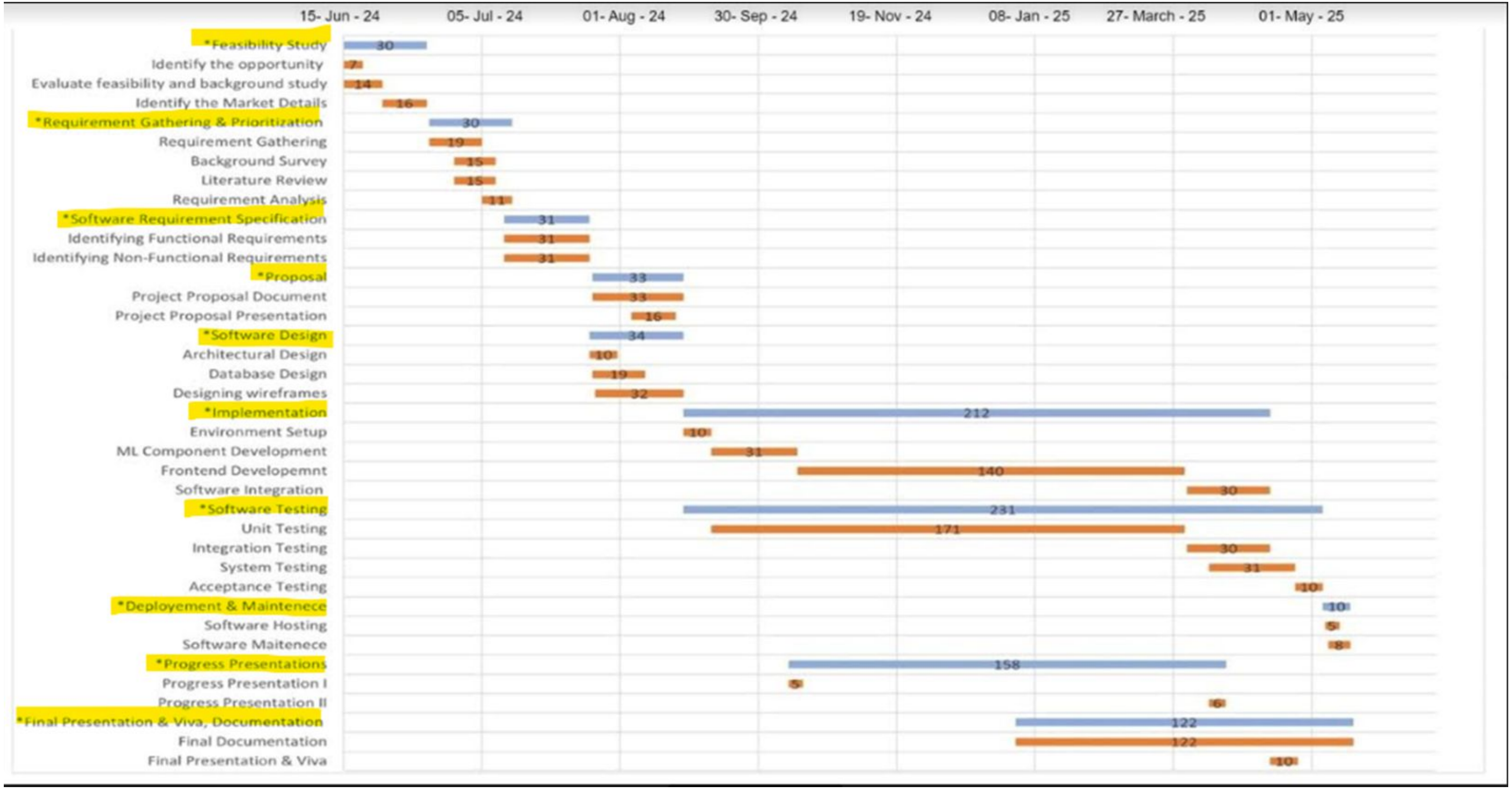
CUSTOM CHATBOT WITH KNOWLEDGE BASE



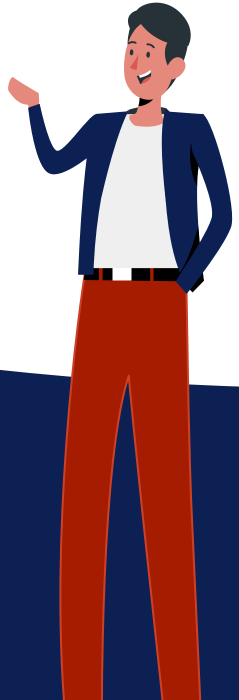
WORK BREAKDOWN STRUCTURE



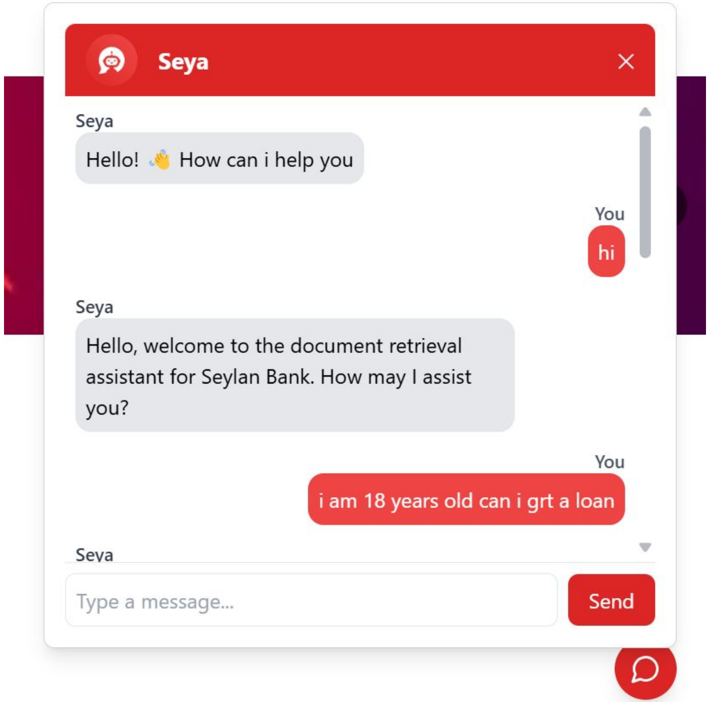
GANTT CHART


15- Jun - 24
05- Jul - 24
01- Aug - 24
30- Sep - 24
19- Nov - 24
08- Jan - 25
27- March - 25
01- May - 25
*Feasibility Study 30
Identify the opportunity 7
Evaluate feasibility and background study 14
Identify the Market Details 16
*Requirement Gathering & Prioritization 30
Requirement Gathering 19
Background Survey 15
Literature Review 15
Requirement Analysis 11
*Software Requirement Specification 31
Identifying Functional Requirements 31
Identifying Non-Functional Requirements 31
*Proposal 33
Project Proposal Document 33
Project Proposal Presentation 16
*Software Design 34
Architectural Design 10
Database Design 19
Designing wireframes 32
*Implementation 212
Environment Setup 10
ML Component Development 31
Frontend Developemnt 140
Software Integration 30
*Software Testing 231
Unit Testing 171
Integration Testing 30
System Testing 31
Acceptance Testing 10
*Deployement & Maintenece 10
Software Hosting 5
Software Maitenece 8
*Progress Presentations 158
Progress Presentation I 5
Progress Presentation II 6
*Final Presentation & Viva, Documentation 122
Final Documentation 122
Final Presentation & Viva 10

COMPLETION OF PP2

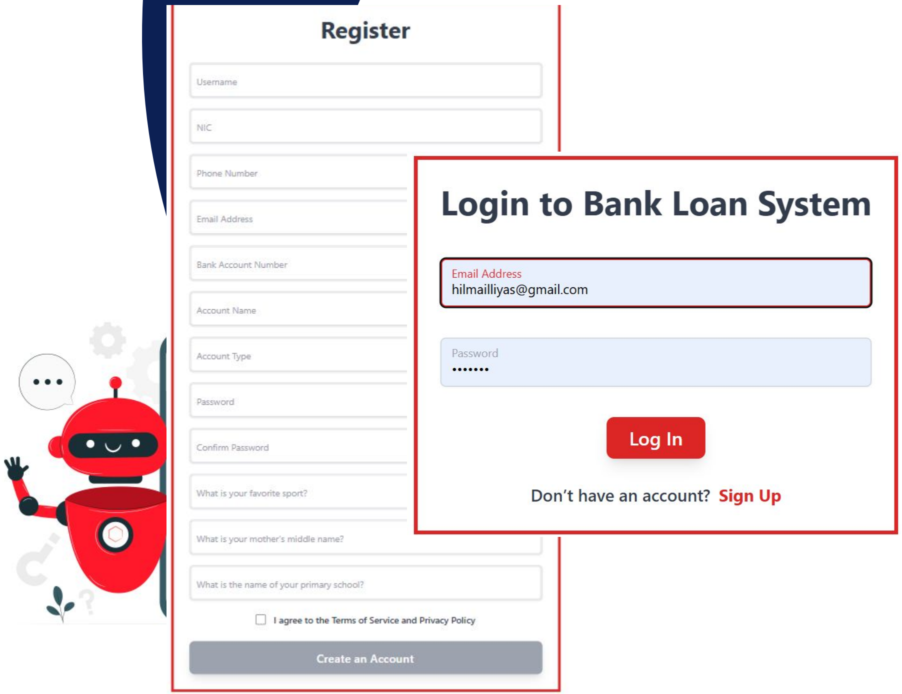


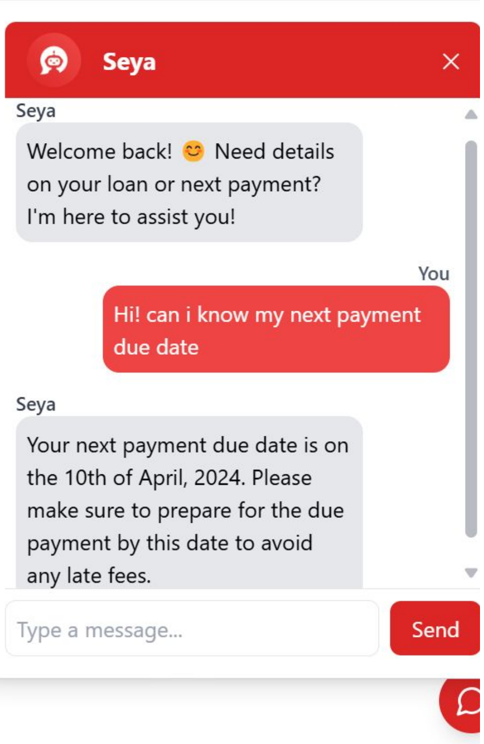
MEET SEYA!

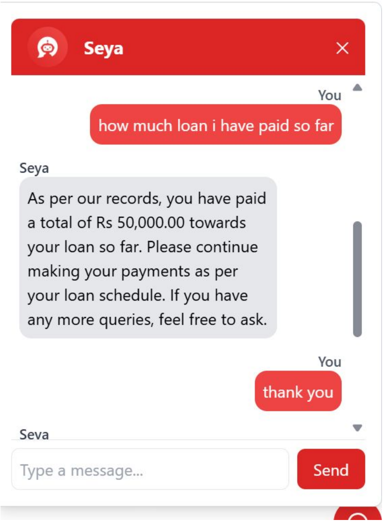


It extracts data, processes it into embeddings, stores it in a vector database, and retrieves answers when queried

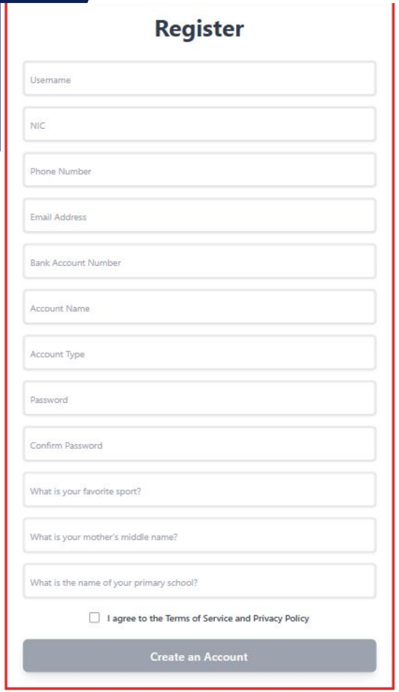
MEET SEYA! -Existing Customer

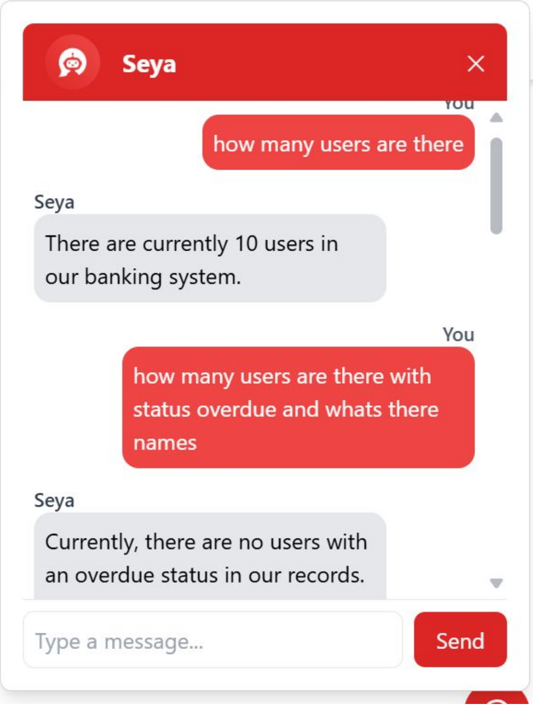


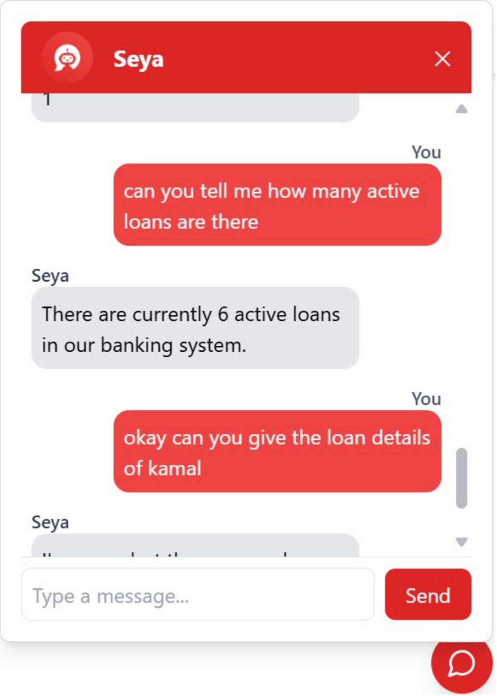




MEET SEYA! - Banker

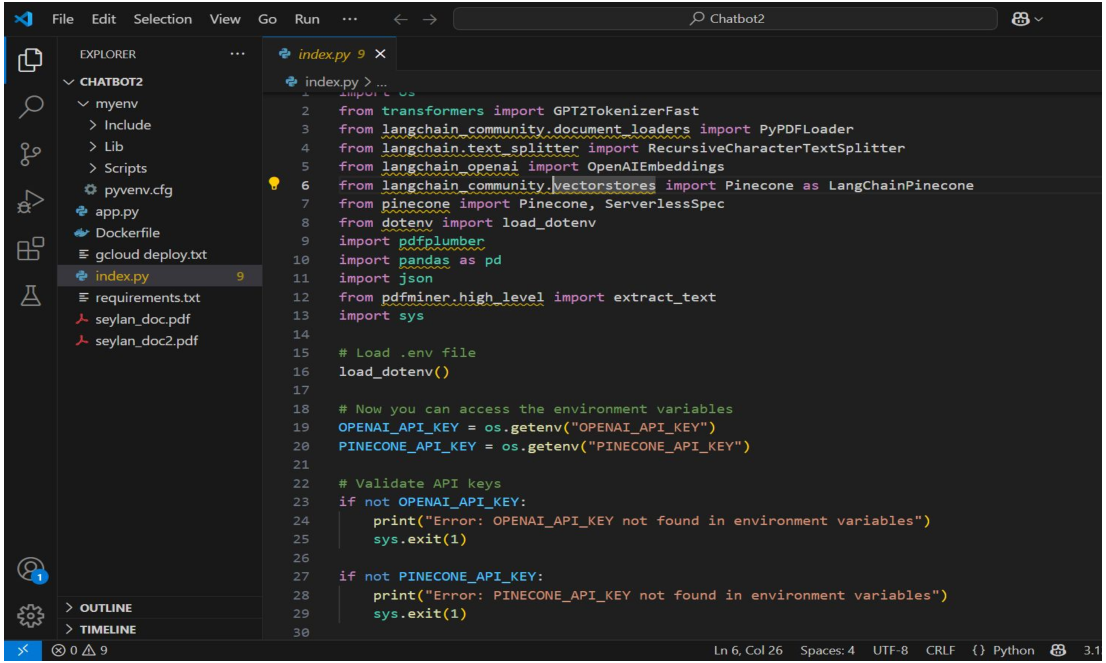




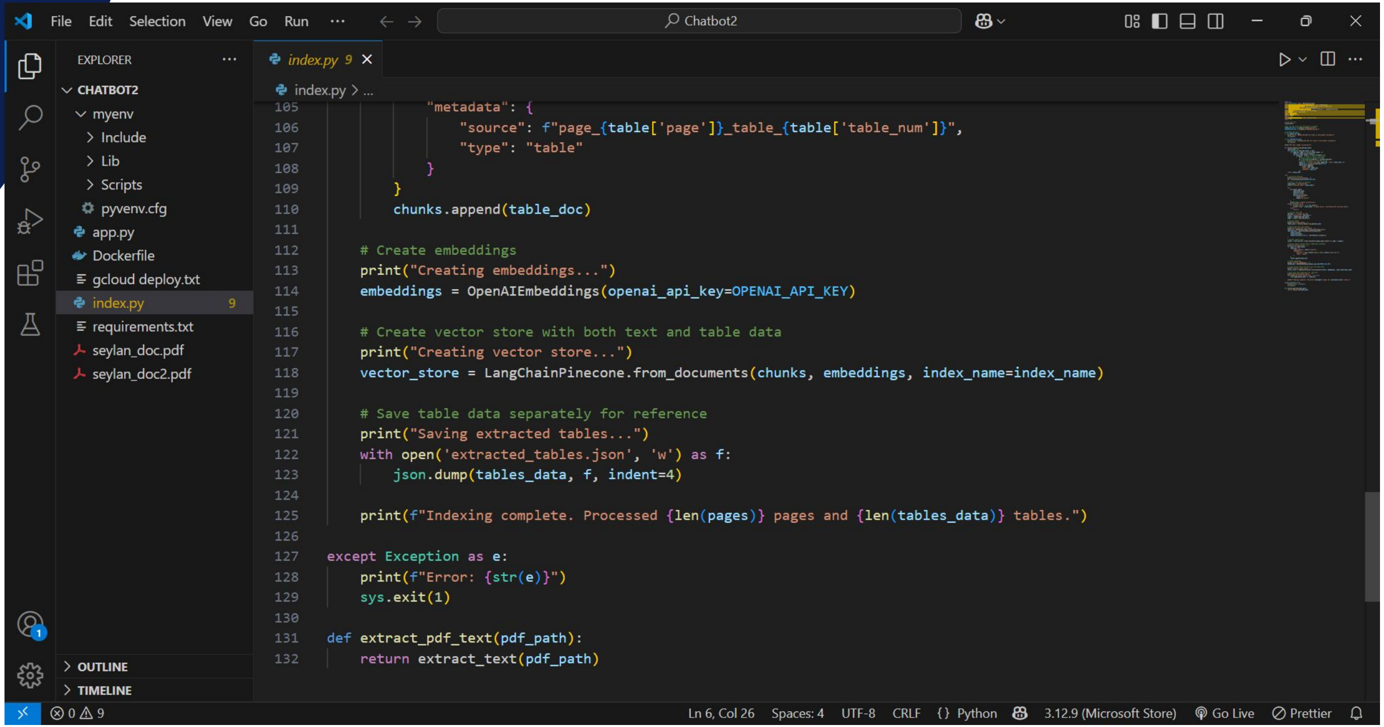


SLIIT FACULTY OF COMPUTING

Custom chatbot



```python
    # Load and split PDF
    pdf_path = "./seylan_doc.pdf"
    print(f"Loading PDF: {pdf_path}")
    loader = PyPDFLoader(pdf_path)
    pages = loader.load_and_split()

    # Extract tables
    print("Extracting tables...")
    tables_data = extract_tables_from_pdf(pdf_path)

    # Convert PDF text into chunks
    print("Converting text into chunks...")
    tokenizer = GPT2TokenizerFast.from_pretrained("gpt2")
    text_splitter = RecursiveCharacterTextSplitter(
        chunk_size=512,
        chunk_overlap=24,
        length_function=lambda x: len(tokenizer.encode(x))
    )

    # Process regular text
    chunks = text_splitter.create_documents([page.page_content for page in pages])

    # Process tables and add them as additional documents
    print("Processing tables...")
    for table in tables_data:
        table_doc = {
            "page_content": table["content"],
            "metadata": {
                "source": f"page_{table['page']}_table_{table['table_num']}",
```


File Edit Selection View Go Run ...
Chatbot2
EXPLORER
CHATBOT2
myenv
Include
Lib
Scripts
pyvenv.cfg
app.py
Dockerfile
gcloud deploy.txt
index.py 9
requirements.txt
seylan_doc.pdf
seylan_doc2.pdf
OUTLINE
TIMELINE
index.py 9
index.py > ...
"metadata": {
"source": f"page_{table['page']}_table_{table['table_num']}",
"type": "table"
}
}
chunks.append(table_doc)
# Create embeddings
print("Creating embeddings...")
embeddings = OpenAIEmbeddings(openai_api_key=OPENAI_API_KEY)
# Create vector store with both text and table data
print("Creating vector store...")
vector_store = LangChainPinecone.from_documents(chunks, embeddings, index_name=index_name)
# Save table data separately for reference
print("Saving extracted tables...")
with open('extracted_tables.json', 'w') as f:
json.dump(tables_data, f, indent=4)
print(f"Indexing complete. Processed {len(pages)} pages and {len(tables_data)} tables.")
except Exception as e:
print(f"Error: {str(e)}")
sys.exit(1)
def extract_pdf_text(pdf_path):
return extract_text(pdf_path)
0 9
Ln 6, Col 26 Spaces: 4 UTF-8 CRLF {} Python 3.12.9 (Microsoft Store) Go Live Prettier

Tasks done

Chatbot Architecture & Design

- Designed chatbot workflow for general customers, existing customers, and bank staff.
- Planned data retrieval methods for loan-related queries.
- Integrated Google Cloud & Firebase for backend management.

Knowledge Base Creation

- Compiled Seylan Bank-specific banking & loan information.
- Structured FAQs & complex banking queries for AI training.
- Implemented embedding techniques for better responses.

AI Model Implementation & Integration

- Used LLMs & vector databases (Pinecone) for chatbot intelligence.
- Developed query processing pipeline using OpenAI API & Flask.
- Optimized encoding & embeddings for accurate responses.

References

•[1] B. Kumar, A. V. Singh, and P. Agarwal, "Trust in Banking Management System using Firebase in Python using AI," in 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021

•[2] R. N. Deborah, S. A. Rajiv, A. Vinora, C. M. Devi, S. M. Arif, and G. S. M. Arif, "An Efficient Loan Approval Status Prediction Using Machine Learning," in 2023 International Conference on Advanced Computing Technologies and Applications (ICACTA), 2023

•[3] W. Pfoertsch and K. Sulaj, "Integrating Artificial Intelligence with Customer Experience in Banking: An Empirical Study on how Chatbots and Virtual Assistants Enhance Empathy," in 2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), 2023

•[4] P. G. Thirumagal, S. Vaddepalli, T. Das, S. Das, S. Madem, and P. S. Immaculate, "AI-Enhanced IoT Data Analytics for Risk Management in Banking Operations," in 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), 2024

THANK YOU