

**SMART LENDER -APPLICANT CREDIBILITY
PREDICTION FOR LOAN APPROVAL**

TEAM ID : PNT2022TMID39687

A PROJECT REPORT

Submitted by

PRABHU RAAM R G -1918501

HARIESH R -1918L05

**MURALI PRASANTH - 1918L07
P**

NANDHA KUMAR -1918L08

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

GOVERNMENT COLLEGE OF TECHNOLOGY,

COIMBATORE

TABLE OF CONTENTS

SI.NO	TITLE	PAGE NO.
1	INTRODUCTION	5
	1.1 Project Overview	5
	1.2 Purpose	5
2	LITERATURE SURVEY	6
	2.1 Existing problem	6
	2.2 References	6
	2.3 Problem Statement Definition	10
3	IDEATION AND PROPOSED SOLUTION	11
	3.1 Empathy Map Canvas	11
	3.2 Ideation and Brainstorming	11
	3.3 Proposed Solution	14
	3.4 Problem Solution fit	15
4	REQUIREMENT ANALYSIS	17
	4.1 Functional requirement	17
	4.2 Non-Functional requirements	17

5	PROJECT DESIGN	19
	5.1 Data Flow Diagrams	19
	5.2 Solution and Technical Architecture	20
	5.3 Users stories	23
6	PROJECT PLANNING AND SCHEDULING	24
	6.1 Sprint Planning and Estimation	24
	6.2 Sprint Delivery Schedule	25
	6.3 Reports from JIRA	26
7	CODING AND SOLUTIONING	29
	7.1 Feature 1	29
	7.2 Feature 2	37
	7.3 Feature 3	51
8	TESTING	63
	8.1 Test Cases	63
	8.2 User Acceptance Testing	80
9	RESULTS	82
	9.1 Performance Metrics	82
10	ADVANTAGES & DISADVANTAGES	84

11	CONCLUSION	85
12	FUTURE SCOPE	85
13	APPENDIX	86
	Source Code	86
	GitHub and Project Demo Link	209

1.

INTRODUCTION

1.1 PROJECT OVERVIEW

The Smart lender project is used to provide a way for the bank to trust the customer and provide the loan to a trust worthy person. This project improves the approval time and decreases the risk associated with the loan. The aim of this project was to train the XGBoost model and it could be used by banks in real world to predict if the loan should be approved or not taking the risk factor in mind.

1.2 PURPOSE

- The smart lender project is used to predict whether a particular applicant is safe to provide loan.
- The complete process can be done by the machine learning model.
- This credit forecasting is very useful for both bank employees and customers.
- The customer can also provide their feedback in this system which helps the bank to improve their service.

2.

LITERATURE SURVEY

2.1 EXISTING PROBLE

Bank employees check the details of applicant and give the loan to eligible applicant. Checking the details of all applicant takes a lot of time. Assessing the risk, which is involved in a loan application, is one of the most important concerns of the banks for survival in the highly competitive market and for profitability. These banks receive number of loan applications from their customers and other people on daily basis. Not everyone gets approved .Most of the banks use their own credit scoring and risk assessment techniques in order to analyze the loan application and to make decisions on credit approval. In spite of this, there are many cases happening every year, where people do not repay the loan amount or they default, due to which these financial institution suffer huge amount of losses.

2.2 REFERENCES

[1] **TITLE:** “An Approach for prediction of loan approval using Machine Learning algorithm”.

AUTHORS: Mohammad Ahmad Sheikh, Amit Kumar Goel, Tapas Kumar.

JOURNAL NAME AND YEAR: ICESC,2020.

DESCRIPTION: This model is marginally better because it includes variables(personal attributes of customers like age, purpose, credit history, credit amount,credit duration etc..) other than checking account information.Therefore,by

using a logistic regression approach, the right customers can be targeted.

LIMITATION: Some other characteristics of customers that play a very important role in lending decisions and forecasting defaulters should be evaluated such as gender and marriage history has not considered in this system.

[2] **TITLE:** "A machine learning approach for predicting bank credit worthiness".

AUTHOR: Turkson, Regina Esi, Edward Yeallakuor Baagyere, and Gideon Evans Wenya.

JOURNAL NAME AND YEAR: IEEE, 2016.

DESCRIPTION: They have employed 15 different learning algorithms on the dataset in order to determine which one is best for studying bank credit data sets. Each of these algorithms achieved an accuracy rate between 76% to over 80%.

LIMITATION: The algorithm Nearest centroid and Gaussian Naïve Bayes have not performed well compared to others in terms of Speed and accuracy.

[3] **TITLE:** "Credit Risk Model Based on Central Bank Credit Registry Data".

AUTHOR: Fisnik Doko, Slobodan Kalajdziski, Igor MishKovski.

JOURNAL NAME AND YEAR: MDPI, 2021.

DESCRIPTION: It has compared five machine learning models to classify credit risk data, i.e., logistic regression, decision tree, random forest, support vector machine (SVM) and neural network. It can predict the credit risk based on credit

history of the population in the country.

LIMITATION:Does not provide better accuracy in lower execution time and have variance and uncertainty in it.

[4] TITLE:”Loan Credibility Prediction System Based On Decision Tree Algorithm”.

AUTHOR:Sivasree M S,Rekha Sunny T.

JOURNAL NAME AND YEAR:IJERT,2015.

DESCRIPTION:It has introduced an effective prediction model for the bankers that help them predict the credible customers who have applied for loan .Decision Tree induction Data mining Algorithm is applied to predict the attribute relevant for credibility.

LIMITATION:Should be incorporated with other techniques that outperform the performance of popular data mining models and should be tested for the domain.

[5] TITLE:”Extracting Prediction Rules for Loan Default Using Neural Networks through Attribute Relevance Analysis”.

AUTHOR: M.V.Jagannatha Reddy and Dr.B.Kavitha.

JOURNAL NAME AND YEAR: IJERT,2010.

DESCRIPTION:This system has extracted prediction rules from the predicted class label and has reduced the number of units required using attribute relevance analysis so that it has increased the speed of neural network technique for

predicting the class label of the tuples and it has used attribute relevance analysis to eliminate irrelevant attributes given as input to neural network.

LIMITATION: In attribute relevance analysis the attributes retained for predicting the class label is very less and the accuracy is appreciable but still can improve the accuracy by calculating the error in wrong predicted rules by adjusting the weights of the neural network .

[6] **TITLE:**”Loan Approval Prediction based on Machine Learning Approach”.

AUTHOR:Kumar Arun,Garg Ishan,Kaur Sanmeet.

JOURNAL NAME AND YEAR:IOSR,2016.

DESCRIPTION:This paper has reduced the risk factor behind selecting the safe person so as to save lots of bank efforts and assets. This is done by mining the big data of the previous records of the people to whom the loan was granted before and on the basis of these records/experiences the machine was trained using the machine learning model which give the most accurate results.

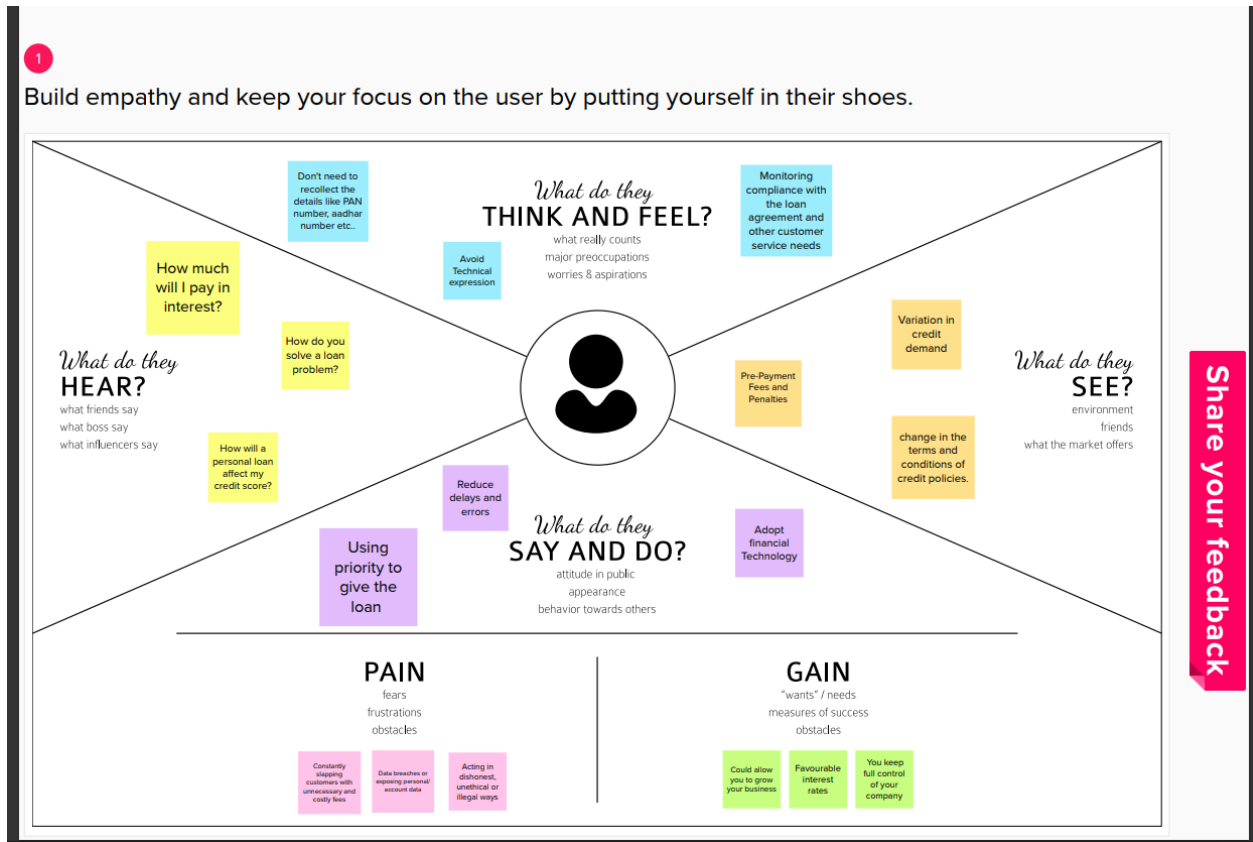
LIMITATION:The disadvantage of this model is that it emphasize different weights to each factor but in real life sometimes loan can be approved on the basis of single strong factor only,which is not possible through this system.

2.3 PROBLEM STATEMENT DEFINITION:

A bank is a financial institution licensed to receive deposits and make loans needs a way to verify the customer details and their documents for getting loan because they need a trustworthy customer with proper credentials who can repay the loan amount and interest on time.

3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING

Define your problem statement: A bank is a financial institution licensed to receive deposits and make loans needs a way to verify the customer details and their documents for getting loan because they need a trust worthy customer with proper credentials who can repay the loan amount and interest on time.



Brainstorm & idea prioritization

In this Template share ideas and further ideas can be written here to modify accordingly , leader will modify these chart based on mentor feedback.

🕒 2 months to prepare
🗓️ 1 month to collaborate
👥 4 Members

🗨️ Share template feedback



Before we collaborate

We have to make sure whether the IBM management provide us good data , we have to make proper planning , analyzing the problem and learn additional skills like storytelling , stakeholder analysis , etc.



Team gathering
Prathy(team leader) will gather group and instruct , ask idea and lead the group further.



Set the goal
Higher Accuracy.
Clean Visuals.
Clean Code.
More Insights



Learn how to use the facilitation tools
1. Youtube and IBM sessions to learn concepts.
2. Use documentation to code new concepts.
3. Use discord , stackoverflow to clear doubts.



Applicant Credibility Prediction for Loan Approval

This data science project will help finance and banking people who give 100's of loan to their applicant and this group project will help stakeholder will come to the number if applicant who are eligible and not eligible by using data visualization , machine learning algorithms and stakeholder will make data driven decisions from this project.

PROBLEM

We are gonna solve this problem by using machine learning algorithms using sci-kit learn and other conventional libraries like spark to handle big data, numpy and pandas for reshaping ,cleaning data,etc.

2

Brainstorm

ideas that come to mind that address your problem statement.

Prabhu	Murali	Hariesh	Nandha
1. Get Big data	1. Use spark to store big data	1. Use spark to store big data	1. Try to use big data and use
2. Use Hadoop, HBase, Hive, Pig, etc.	2. Use Hadoop, HBase, Hive, Pig, etc.	2. Use Hadoop, HBase, Hive, Pig, etc.	2. Use Hadoop, HBase, Hive, Pig, etc.
3. Use Hadoop, HBase, Hive, Pig, etc.	3. Use Hadoop, HBase, Hive, Pig, etc.	3. Use Hadoop, HBase, Hive, Pig, etc.	3. Use Hadoop, HBase, Hive, Pig, etc.
4. Use Hadoop, HBase, Hive, Pig, etc.	4. Use Hadoop, HBase, Hive, Pig, etc.	4. Use Hadoop, HBase, Hive, Pig, etc.	4. Use Hadoop, HBase, Hive, Pig, etc.
5. Use Hadoop, HBase, Hive, Pig, etc.	5. Use Hadoop, HBase, Hive, Pig, etc.	5. Use Hadoop, HBase, Hive, Pig, etc.	5. Use Hadoop, HBase, Hive, Pig, etc.
6. Use Hadoop, HBase, Hive, Pig, etc.	6. Use Hadoop, HBase, Hive, Pig, etc.	6. Use Hadoop, HBase, Hive, Pig, etc.	6. Use Hadoop, HBase, Hive, Pig, etc.
7. Use Hadoop, HBase, Hive, Pig, etc.	7. Use Hadoop, HBase, Hive, Pig, etc.	7. Use Hadoop, HBase, Hive, Pig, etc.	7. Use Hadoop, HBase, Hive, Pig, etc.
8. Use Hadoop, HBase, Hive, Pig, etc.	8. Use Hadoop, HBase, Hive, Pig, etc.	8. Use Hadoop, HBase, Hive, Pig, etc.	8. Use Hadoop, HBase, Hive, Pig, etc.
9. Use Hadoop, HBase, Hive, Pig, etc.	9. Use Hadoop, HBase, Hive, Pig, etc.	9. Use Hadoop, HBase, Hive, Pig, etc.	9. Use Hadoop, HBase, Hive, Pig, etc.
10. Use Hadoop, HBase, Hive, Pig, etc.	10. Use Hadoop, HBase, Hive, Pig, etc.	10. Use Hadoop, HBase, Hive, Pig, etc.	10. Use Hadoop, HBase, Hive, Pig, etc.

3

Group ideas

Share ideas and we can make further planning based on mentor feedback.

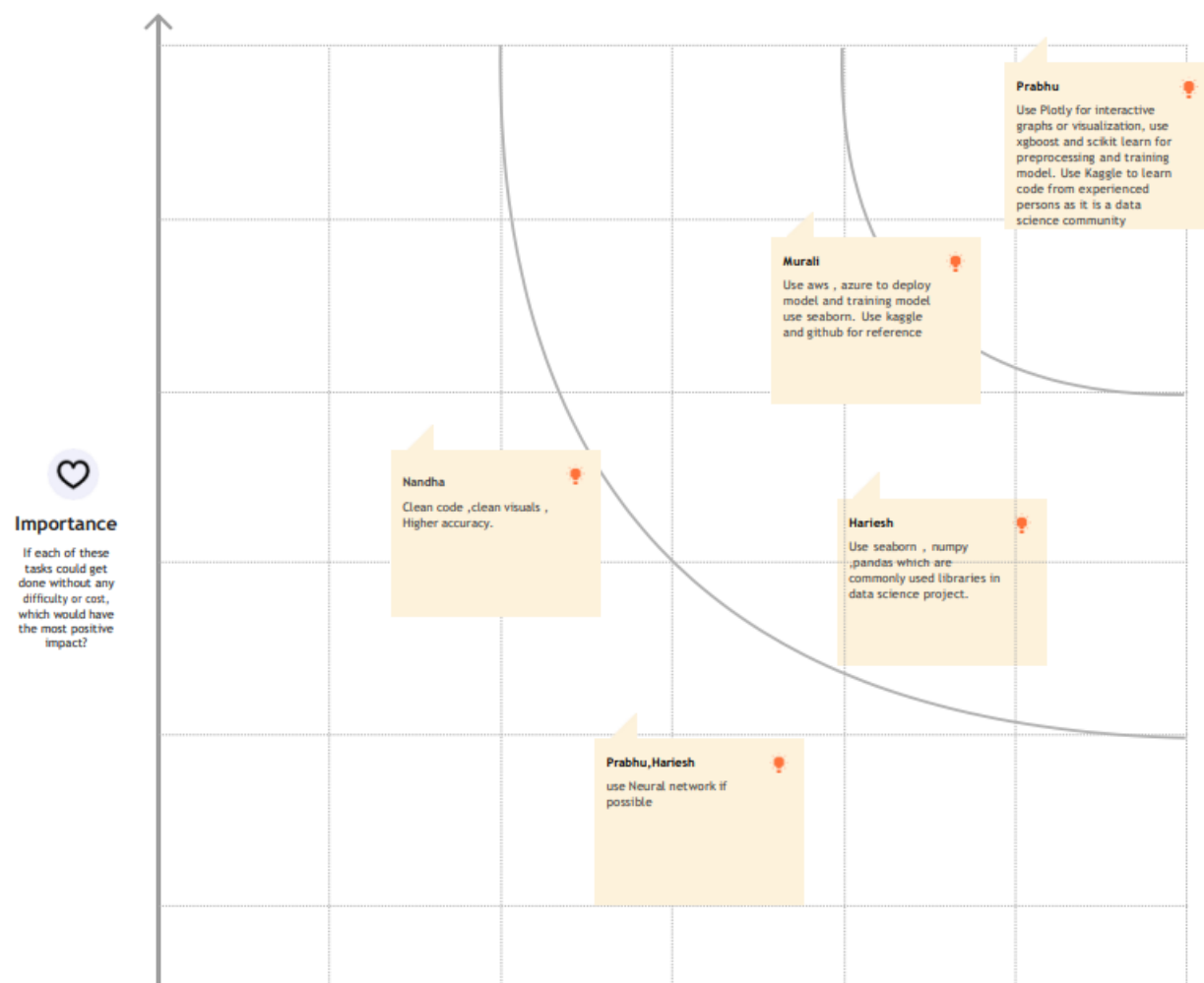
Prabhu Use Numpy , pandas , plotly	Murali Use Matplotlib	Hariesh Use seaborn for clean visualization , use testing techniques if possible.	Nandha Refactor code if possible , use clean visuals and use required libraries to reduce complexity
Prabhu Use Apache spark to store big data	Murali Use Numpy , pandas , Matplotlib		
Prabhu Use Xgboost for regression	Murali Use charts like barchart , piechart , ribbon chart based on data provided		
Prabhu Use aws or azure for model training and deploying model.			

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

🕒 20 minutes



3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement	The banking and finance people who are giving 100's of loans to their applicant will predict, if the applicant is eligible or non eligible for the loan approval.
2.	Idea/Solution description	Using data visualization , machine learning algorithm ,the stakeholder will make data driven decision from this project
3.	Novelty/Uniqueness	<ul style="list-style-type: none">• In our model, there is no need to recollect details like PAN and Aadhar.• We will create a model to monitor compliance with the loan agreement and other customer service needs. Agreement of the terms and conditions.
4	Social Impact	From the analysis of data, it's very clear that it reduces all the frauds done at the time of loan approval.

5.	Business Model	<ul style="list-style-type: none"> It's quite efficient and reliable model which will not only reduce the time of the bank but also reduce the waiting time of the customer
6.	Scalability Of Solution	<ul style="list-style-type: none"> It's possible to make changes to software, which can accept new testing data and should also take part in training data and predict accordingly. In future prediction, module can be more improved and integrated

3.4 PROBLEM SOLUTION FIT

Problem-Solution fit canvas 2.0		To find a solution that can help the banking sector identify whether a given person is eligible for a loan.	
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Our Customers are officials from the banking sector who handle the loaning process for their customers.	6. CUSTOMER CONSTRAINTS CC Choices of solutions are limited by the budget, knowledge required to use the solution, database access, internet connectivity, etc.	5. AVAILABLE SOLUTIONS AS Solutions include: 1. Creating an database functions that gives results based on information of a person. Pros: simple, Cons: low accuracy. 2. Using past records of other people with conditions similar to the current person. Pros: Good Accuracy, Cons: Past records cannot exist for all types of conditions of the current person. 3. Creating an applicaion that uses machine learning techniques to predict the loan eligibility of person.
	Focus on J&P, tap into	2. JOBS-TO-BE-DONE / PROBLEMS J&P Problems include deciding whether or not a customer is eligible for availing a loan scheme from bank, increasing customer interaction, etc.	9. PROBLEM ROOT CAUSE RC Customer are required to increase their ability to recognise a person's loan eligibility in order increase the productivity of the company and cater to needs of the customers.
Identify strong TR & EM		3. TRIGGERS TR Customers are triggered from the need to standardise the loan process and make their work more customer-friendly.	10. YOUR SOLUTION SL Our Solution involve using ensemble ML models which can predict the loan eligibility of a person in the form of score that can be used as standard in the banking sector. It will also include an application framework that uses this ML model and some UI/UX to provide the results directly to the customer.
	4. EMOTIONS: BEFORE / AFTER EM Customers, from being indecisive, become more confident about the whether or not the person-in-question is able to avail their loan schemes.	8.2 OFFLINE Customers spend time slowly analysing the various details of people inorder to derive the result.	

4.

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	Home Page	<ul style="list-style-type: none">• Smart Lender Applicant Credibility description Information about Credibility details required for loan approval• if new user , REGISTER• if already exist , SIGN IN
FR-2	User Registration	Enter Mail Id and other personal details required for Registering
FR-3	User login	User Mail Id and Password for Login
FR-4	Loan Approval form	Credibility details should be entered for prediction
FR-5	Result	if Approved - It display the information about what is done to be next. if Not Approved - It display the information about what rejection criteria you are not eligible for the loan.

4.2 NON-FUNCTIONAL REQUIREMENT

Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none"> It describes the context Who,What,When,Where and Why. The specific activities the requirements describe should reflect the both range of goal that the system must support and business goals for creating new system.
NFR-2	Security	<ul style="list-style-type: none"> Security functionality that ensures one of many different security properties of software is being satisfied.Security requirements are derived from industry standards,applicable laws,and a history pf past vulnerabilities.
NFR-3	Reliability	<ul style="list-style-type: none"> It is the measure of the stability or consistency of the test score
NFR-4	Performance	<ul style="list-style-type: none"> It defines how well the software system accomplishes certain functions under specific condition.
NFR-5	Availability	<ul style="list-style-type: none"> It defines how long the IT system can be unavailable without impacting operations.
NFR-6	Scalability	<ul style="list-style-type: none"> It is the measure of a system ability to increase or decrease in performance and cost in response.

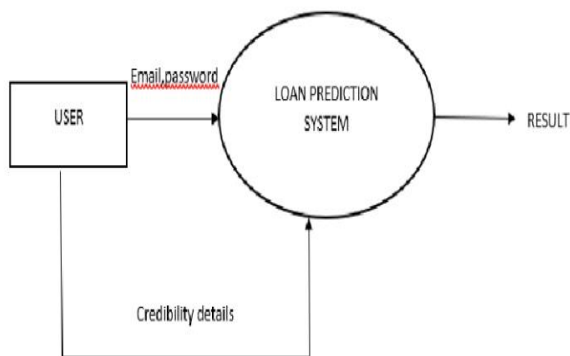
5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

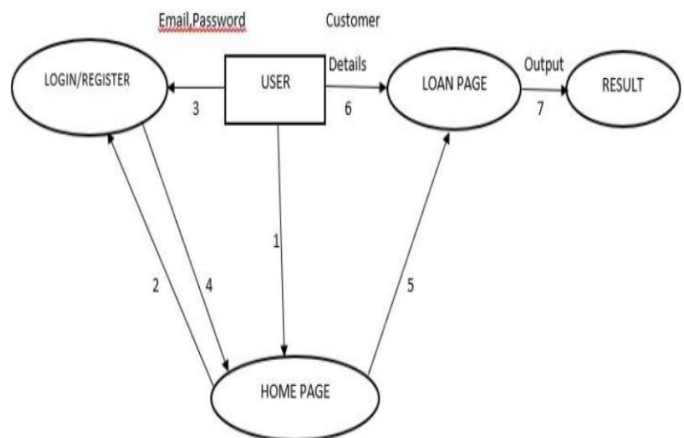
Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

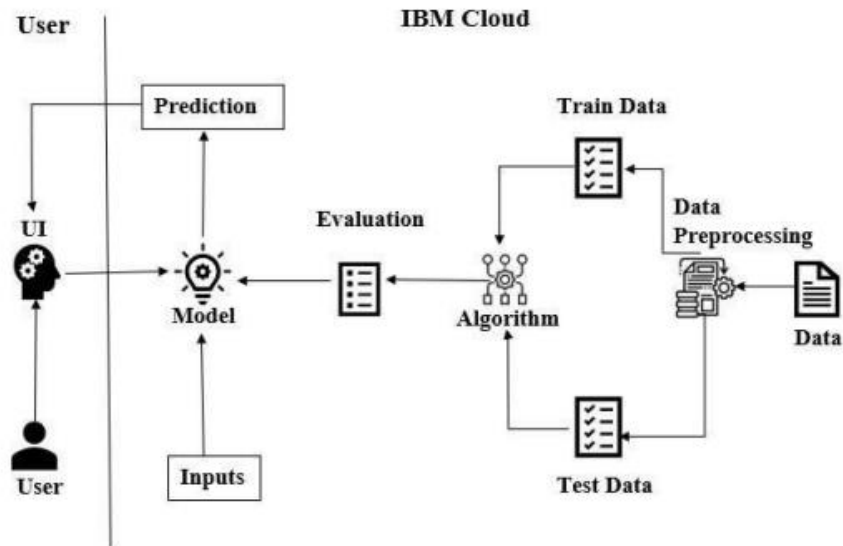
0th LEVEL DFD:



1st LEVEL DFD:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	User interact with our application through web User Interface.	HTML, CSS and Python flask.
2.	Application Logic-1-Login.	When the user click on the login button , he/she is directed to login page, if they are registered already.	HTML, CSS, Python flask
3.	Application Logic-Registration	When the user click on the Register button , he/she is directed to Register page for further process.	HTML, CSS, Python flask

4.	Application Logic- Credibility details	After Logged in , when the user click on the credibility details form button, he/she directed to the form page to enter the details of applicant for prediction	Front end- HTML ,CSS , MySQL, Pythonflask Back end-Python
5.	Database	Data type - String ,Numeric.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM
7.	File Storage	File storage requirements	NIL
8.	External API-1	Purpose of External API used in the application	NIL
9.	External API-2	Purpose of External API used in the application	Aadhar API
10.	Machine Learning Model	Get the data from the user and predict the data with tested and trained dataset models	Data Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	NIL

Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	International Business Machines	Cloud
2	Security Implementations	Access permission for login page using CAPTCHA	Encryption
3	Scalable Architecture	The key of Three tier architecture is improving scalability.	Three Tier architecture.
4	Availability	Load balancer or ADC is the key component that ensures high availability by sending request.	Load balancer

5	Performance	The system should be able to handle large number of users at the time	Load balancer
---	-------------	---	---------------

5.3 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Home Page	USN - 1	Loan approval prediction description	I can view /access my homepage.	Low	Sprint - 3
		USN - 2	Information about the credibility details required for the prediction		Low	Sprint - 3
	User Register	USN - 3	Enter Email ID and other personal details required for Register.	I can successfully register by receiving mail.	Medium	Sprint - 2
	User Login	USN - 4	Uses Email ID and Password for login	I have successfully logged in.	Medium	Sprint - 2
	Loan approval Form	USN - 5	Credibility details required for loan should be entered for prediction.	I can access the customer details form	High	Sprint - 1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Result	USN - 6	Results will be displayed.	I got my result successfully.	High	Sprint - 1
		USN - 7	<ul style="list-style-type: none"> If Approved, The information about what is done to be next is displayed. If Not approved, The information about which rejection criteria you are not eligible for the loan is displayed. 	I got useful information	Low	Sprint - 4

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my details.	2	High	Prabhu raam, Murali
Sprint-1	Login	USN-2	As a user, I can log into the application by entering the user name and password.	2	High	Prabhu raam, Murali
Sprint-1		USN-3	As a user,I can log in using Gmail.		Low	Prabhu raam, Murali
Sprint-2	Uploaded details	USN-4	As a user,I can upload my details and documents.	3	High	Hariesh, Nandhakumar
Sprint-2	Navigation	USN-5	As a user, I can navigate to different tabs like home, description, contact,login,procedure.	2	Low	Hariesh, Nandhakumar
Sprint-2	View Procedure	USN-6	As a user, I can view the procedure to apply for loan.	1	Medium	Hariesh, Nandhakumar
Sprint-2	Contact	USN-7	As a user, I can contact bank.	1	Low	Prabhu raam, Murali, Hariesh, Nandhakumar

Sprint-3	Ratings	USN-8	As a user, I can provide ratings for the service Provided.	2	Medium	Prabhu raam, Murali, Hariesh, Nandhakumar
Sprint-3	View User Details	USN-9	As a Bank administrator , I can view the user details.	2	Medium	Prabhu raam, Murali, Hariesh, Nandhakumar
Sprint-3	Credit Verification	USN-10	As a Bank administrator, I can verify the credibility of the customer.	3	High	Prabhu raam, Murali, Hariesh, Nandhakumar
Sprint-4	Document Verification	USN-11	As a Bank administrator,I can Verify all the documents proof and ID	3	High	Prabhu raam, Murali, Hariesh, Nandhakumar

6.2 SPRINT DELIVERY SCHEDULE

			proof of the customer.			
Sprint-4	Loan Approval Status	USN-12	As a Bank administrator, I can Approve/Reject the loan for the customer based on their details.	3	High	Prabhu raam, Murali, Hariesh, Nandhakumar
Sprint-4		USN-13	As a user, I can get confirmation of loan approval through email.	3	High	Prabhu raam, Murali, Hariesh, Nandhakumar

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	5	6 Days	24 Oct 2022	29 Oct 2022	5	29 Oct 2022
Sprint-2	7	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	7	6 Days	07 Nov 2022	12 Nov 2022	7	12 Nov 2022
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	6	19 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{20}{10} = 2$$

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time it is often used in agile software development methodologies such as scrum. However, burn downcharts can be applied to any project containing measurable progress over time



CODING AND SOLUTIONING

7.1 Feature 1: Prediction

To predict the credibility, XGBoost has been used.

Code:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import cross_val_score

#the model used to fit&predict
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
#pipeline with its' preprocessor's transformers
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelEncoder
from sklearn.compose import ColumnTransformer

#used for estimating model accuracy and getting reports
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

df = pd.read_csv(r"C:\Users\mural\OneDrive\Desktop\Surya_project-main\Surya_project-main\Loan_Approval_Prediction\LoanPrediction.csv")
print(df.info())
df.head()
```

RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	Loan_ID	614 non-null	object
1	Gender	601 non-null	object
2	Married	611 non-null	object
3	Dependents	599 non-null	object
4	Education	614 non-null	object
5	Self_Employed	582 non-null	object
6	ApplicantIncome	614 non-null	int64
7	CoapplicantIncome	614 non-null	float64
8	LoanAmount	592 non-null	float64
9	Loan_Amount_Term	600 non-null	float64
10	Credit_History	564 non-null	float64
11	Property_Area	614 non-null	object
12	Loan_Status	614 non-null	object

dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB

```

None
Out[2]:
  Loan_ID  Gender  Married  Dependents  Education  Self_Employed  ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  Credit_History  Property_Area  Loan
0 LP001002   Male    No         0   Graduate      No             5849              0.0          NaN          360.0            1.0        Urban
1 LP001003   Male    Yes         1   Graduate      No             4583             1508.0        128.0          360.0            1.0        Rural
2 LP001005   Male    Yes         0   Graduate      Yes             3000              0.0          66.0          360.0            1.0        Urban
3 LP001006   Male    Yes         0   Not Graduate  No             2583             2358.0        120.0          360.0            1.0        Urban
4 LP001008   Male    No         0   Graduate      No             6000              0.0          141.0         360.0            1.0        Urban

In [3]:
[print(col, df[col].nunique()) for col in df.columns if df[col].dtype=='object']
df.drop('Loan_ID', axis=1, inplace=True)

Loan_ID 614
Gender 2
Married 2
Dependents 4
Education 2
Self_Employed 2
Property_Area 3
Loan_Status 2

In [4]:
y = LabelEncoder().fit_transform(df['Loan_Status'])
X = df.drop(['Loan_Status'], axis=1)

train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.2, random_state=0)

In [5]:
print(y)

[1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0
 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0
 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]

In [5]:
print(y)

[1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0
 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 1 0
 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1
 0 1 0 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0 1 1 0 0 0 1 0 1
 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 1 0 0 0 1 1 1 1 0 1 0 1 0 1
 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 1 1 0 1 0 0 0
 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1
 1 1 0 0 0 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1
 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 0
 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 1 0 1 1
 1 0 0 0 1 0 1 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1
 1 1 1 0 0 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 1 0 1
 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1 1 0
 0 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 0 1 0 1 1 0 0
 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1 0
 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0]

In [6]:
num_cols = [col for col in X.columns if X[col].dtype in ['int64', 'float64']]
cat_cols = [col for col in X.columns if X[col].dtype == 'object']
print(num_cols)
print(cat_cols)

['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History']
['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Property_Area']

In [7]:
num_transformer = Pipeline(steps=[
    ('impute', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

cat_transformer = Pipeline(steps=[
    ('impute', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore', sparse=False))
])

preprocessor = ColumnTransformer(transformers=[
    ('num', num_transformer, num_cols),

```



```
preprocessor = ColumnTransformer(transformers=[
    ('num', num_transformer, num_cols),
    ('cat', cat_transformer, cat_cols)
])
```

```
In [8]: def train_predict(full_pipeline, train_X, train_y, test_X, test_y):
        full_pipeline.fit(train_X, train_y)
        pred_y = full_pipeline.predict(test_X)
        print('accuracy_score:', accuracy_score(test_y, pred_y))
        print('classification_report:', classification_report(test_y, pred_y))

        print('cross_val_score:')
        scores = cross_val_score(full_pipeline, X, y, cv=5, scoring='accuracy')
        print(scores)
        print('mean of scores:', scores.mean())
```

```
In [14]: from xgboost import XGBClassifier

        models = [

            ('XGB', XGBClassifier(learning_rate=0.1,
                                   n_estimators=100,
                                   max_depth=6,
                                   min_child_weight = 1,
                                   gamma=0.,
                                   subsample=0.8,
                                   scale_pos_weight=1,
                                   random_state=27)),

        ]

        for model_name, model in models:
            print('\nModel %s.' % model_name)
            full_pipeline = Pipeline(steps=[
                ('preprocessor', preprocessor),
                ('model', model)
            ])
            train_predict(full_pipeline, train_X, train_y, test_X, test_y)
            print('*' * 50)

        ]
        train_predict(full_pipeline, train_X, train_y, test_X, test_y)
        print('*' * 50)
```

```
Model XGB.
accuracy_score: 0.8211382113821138
classification_report:

```

			precision	recall	f1-score	support
	0	0.70	0.58	0.63	33	
	1	0.85	0.91	0.88	90	
accuracy			0.82	123		
macro avg	0.78	0.74	0.76	123		
weighted avg	0.81	0.82	0.82	123		

```

cross_val_score:
[0.74796748 0.73170732 0.75609756 0.78861789 0.78688525]
mean of scores: 0.7622550979608156
*****
```

```
In [15]: import pickle

        pickle.dump(full_pipeline, open('Decision_Tree_Model.pkl', 'wb'))
```

```
In [16]: predicted = full_pipeline.predict(pd.DataFrame(columns=['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome']))
```

```
In [17]: predicted
```

```
Out[17]: array([0])
```

```
In [18]: if(predicted == 1):
        print('loan_approved')
        else:
        print("Not_approved")
```

Not_approved

7.2 FEATURE 2:Prediction

The user will upload all their details in this prediction page to predict the eligibility of them to to get the loan

Code:

8.TESTING

8.1 TEST CASES

Test case ID	Functional	Test Scenario	Test data	Result	status
Home page_TC_001	Functional	Verify user is able to see the Home page when the user enters the	url of the web application	Home page should be displayed	pass

		url			
Home page_TC_002	UI	Verify the UI elements in the home page	home.html	Application should display UI elements: a.home tab b.about tab c.procedure tab d.user login tab e.Bank login tab f.predict button	pass
LoginPage_TC_003	Functional	Verify user is able to see the Login page when user clicked on User login button	home.html	Login page should display	pass
LoginPage_TC_004	UI	Verify the UI elements in Login page	login.html	Application should show below UI elements: a.user name or email id text box b.password text box c.captcha text box d.Login button with green colour e.New user? Register link	pass
LoginPage_TC_005	Functional	Verify user is able to log into application with valid credentials	Username: aaa@gmail.com password: aishu	User should navigate to prediction page	pass
LoginPage_TC_006	Functional	Verify user is able to log into application with invalid credentials	Username:aaa/EmailId: john2@gmail.com password: john123	Application should show 'Login failed Invalid username or gmail ID or password' validation message.	pass
LoginPage_TC_007	Functional	Verify user is able to log into application with Invalid credentials	Username: john password:john123	Application should show 'Login failed Invalid username or gmail ID or password ' validation message.	pass
LoginPage_TC_008	Functional	Verify user is able to log into application with InValid	Username: john2@gmail.co password: john1234	Application should show 'Login failed Invalid username or gmail ID or password ' validation message.	pass

		credentials			
LoginPage_TC_OO9	Functional	Verify user is able to log into application with valid captcha	Username: john@gmail.co password: john123 valid captcha: 1xc9830 entered captcha: 1xc9830	Application should login to prediction page	pass
LoginPage_TC_O10	Functional		Username: john@gmail.co password: john123 Valid captcha: 1xc9830 entered captcha: 2xcv987	Application should display 'Invalid captcha' message	pass
Registration_TC_O11	Functional	Verify user is able to see the Registration page when user clicked on Registration button	login.html	Registration page should be displayed	pass
Registration_TC_O12	UI	Verify the UI elements in Register page	register.html	Application should show below UI elements: a.Name text box b.email ID text box c.password text box d.Register button	pass
Registration_TC_O13	Functional	Verify user is able to register into application with Valid credentials	Name: vijay username: vijay@02 password: vv@002 mobilenumber: 9089097878	Application displays 'Successful registration'	pass
Registration_TC_O14	Functional	Verify user is able to register into application with Invalid credentials	Name: aaaaa emailID: aaa@gmail.com username: aaaa password: aishu mobile number: 9089097878	Application displays 'someone had already registered with this details'	pass

Prediction_ TC_15	Functional	Verify user is able to see the prediction page when user clicked on predict button	prediction.html	Prediction page should be displayed	pass
Prediction_ TC_16	UI	Verify the UI elements in prediction page	prediction.html	<p>Application should show below UI elements:</p> <ul style="list-style-type: none"> a. Name text box b. Email ID text box c. Mobile number text box d. Gender drop down menu e. Education drop down menu f. Self_Employed drop down menu g. married drop down menu h. Dependents i. Property area drop down menu j. credit History k. Applicant income text box l. Coapplicant income text box m. Loan amount text box n. Loan amount term text box o. I accept the terms and conditions check box p. Predict button q. Back button 	pass

Prediction_ TC_025	Functional	Verify user is able to predict the results with Valid credentials	Name:john Email ID:john@gmail.com Mobile number:9087890980 Gender:male Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:10000 Coapplicant Income:5000 Loan amount:7000 Loan amount term:360 Adhar number:2857 6789 6784 PAN card ID:9078508844 Property document:property.pdf Govt ID proof:Adhar.pdf checked the check box'I accept the terms and conditions	Application should be directed to the approve page or reject page baased on their given data	
Prediction_ TC_026	Functional	Verify user is able to predict the results with Invalid name	Name:12345 Email ID:aishu@gmail.com Mobile number:9087890980 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:10000 Coapplicant Income:5000 Loan	Application should display 'Name must contain only alphabets'	pass

			amount:7000 Loan amount term:360 Adhar number:2857 6789 6784 PAN card ID:9078508844 Property document:property.pdf Govt ID proof:Adhar.pdf checked the check box'I accept the terms and conditions		
Prediction_ TC_027	Functional	Verify user is able to predict the results with Invalid emailID	Name:aishu Email ID:aishugmail.com Mobile number:9087890980 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:10000 Coapplicant Income:5000 Loan amount:7000 Loan amount term:360 Adhar number:2857 6789 6784 PAN card ID:9078508844 Property document:property.pdf Govt ID proof:Adhar.pdf checked the check box'I accept the terms and conditions	Application should display 'please include an @ symbol in emailId'	pass

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	3	2	2	20
Duplicate	1	0	3	0	3
External	5	2	0	1	5
Fixed	10	3	4	20	34
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	25	11	13	25	71

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested


Section	Total Cases	Not Tested	Fail	Pass
Print Engine	11	0	0	8
Client Application	48	0	0	48
Security	2	0	0	2

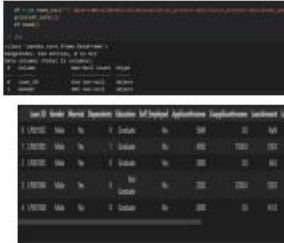
Outsource Shipping	3	0	0	3
Exception Reporting	8	0	0	8
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.

RESULTS

9.1 PERFORMANCE METRICS

S. No	Parameter	Values	Screenshot																														
1.	Metrics	<p>Classification Model:</p> <p>Accuracy Score- Xgboost Model Testing Accuracy 0.905135135135135 Xgboost Model Training Accuracy 0.905135135135135</p> <p>Classification Report -</p> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>1.00</td><td>0.76</td><td>0.87</td><td>68</td></tr><tr><td>1</td><td>0.88</td><td>1.00</td><td>0.94</td><td>117</td></tr></tbody></table> <table><tbody><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>185</td></tr><tr><td>macro avg</td><td>0.94</td><td>0.88</td><td>0.90</td><td>185</td></tr><tr><td>weighted avg</td><td>0.92</td><td>0.91</td><td>0.91</td><td>185</td></tr></tbody></table>		precision	recall	f1-score	support	0	1.00	0.76	0.87	68	1	0.88	1.00	0.94	117	accuracy			0.91	185	macro avg	0.94	0.88	0.90	185	weighted avg	0.92	0.91	0.91	185	
	precision	recall	f1-score	support																													
0	1.00	0.76	0.87	68																													
1	0.88	1.00	0.94	117																													
accuracy			0.91	185																													
macro avg	0.94	0.88	0.90	185																													
weighted avg	0.92	0.91	0.91	185																													

2.	Tune the Model	<p>Hyperparameter Tuning - No tuning is performed as we have got 91% accuracy</p> <p>Parameters used- n_estimators=5000,max_depth=80,max_features='log2'</p> <p>Validation Method - In-sample validation</p>	
----	----------------	--	--

10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- Customer can be able to know whether they are eligible for the loan from any part of the world.
- Because of using machine learning model, there will be no chance of cheating happen.
- Minimal documentation is required and there is no physical submission of documents.
- Every process will be easily accessible by both the parties.
- Time period for loan sanctioning will be reduced and more Accurate prediction for loan eligibility will be given.
- The customer can contact bank at any time in case of any queries and we hadalso provided the detailed procedure for applying loan and customer can alsoprovide the ratings

DISADVANTAGES

- The customer can contact the lender only through online using email or call them in case of any queries
- The bank should externally connect to database and use this software in real time we had provided only the feature
- There may be some risk associated with security of the customers as they are providing all their details in online
- The Accuracy of prediction can also be improved

11.

CONCLUSION

Conclusion

The analysis has started from data preprocessing, handling missing value, exploratory analysis and different models were build like Decision tree model,KNN model,Xgboost model and Random Forest model and there performance were evaluated , as a result the XGBoost model is selected as the best model for predicting the loan approval status of the customer after evaluating its performance ,as it got 76 % accuracy in prediction. This application is

then tested and it functions properly, and it also meets all the requirements of the bank in selecting the trust worthy person to provide loan.

12. FUTURE SCOPE

Future scope

In future development, transactions for money exchange while repaying loan has been simplified.

13. APPENDIX

SOURCE CODE

XGBoost.ipynb

```
import pandas as pd
import numpy as np
from sklearn.model_selection import cross_val_score

#the model used to fit&predict
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
#pipeline with its' preprocessor's transformers
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelEncoder
from sklearn.compose import ColumnTransformer

#used for estimating model accuracy and getting reports
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

```
df = pd.read_csv(r"C:\Users\mural\OneDrive\Desktop\Surya_project-main\Surya_project-main\Loan_Approval_Prediction\LoanPrediction.csv")
print(df.info())
df.head()
```

```
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Loan_ID               614 non-null    object
1   Gender                601 non-null    object
2   Married               611 non-null    object
3   Dependents            599 non-null    object
4   Education              614 non-null    object
5   Self_Employed         582 non-null    object
6   ApplicantIncome       614 non-null    int64
7   CoapplicantIncome     614 non-null    float64
8   LoanAmount            592 non-null    float64
9   Loan_Amount_Term      600 non-null    float64
10  Credit_History         564 non-null    float64
11  Property_Area         614 non-null    object
12  Loan_Status           614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
None
```

```
Out[2]:
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	0
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	1
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	0
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	0
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	0

```
In [3]: [print(col, df[col].nunique()) for col in df.columns if df[col].dtype=='object']
df.drop('Loan_ID', axis=1, inplace=True)
```

```
Loan_ID 614
Gender 2
Married 2
Dependents 4
Education 2
Self_Employed 2
Property_Area 3
Loan_Status 2
```

```
In [4]: y = LabelEncoder().fit_transform(df['Loan_Status'])
X = df.drop(['Loan_Status'], axis=1)

train_X, test_X, train_y, test_y = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
In [5]: print(y)

[1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0
 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 1 1 0 1 1 1 0
 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

```
In [5]: print(y)

[1 0 1 1 1 1 0 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0
 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 1 1 0 1 1 1 1 0 0 0 0 0 1 1 0 1 1 1 0
 1 0 0 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1
 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 1
 0 1 0 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0 1 0 0 0 1 0 1
 1 0 1 1 1 1 0 0 1 1 0 1 1 1 0 1 1 0 1 1 1 1 1 0 0 0 1 1 1 1 0 1 0 1 0 1
 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0
 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1
 1 1 0 0 0 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1
 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 0
 1 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 0 0 1 1 0 1 1
 1 0 0 0 1 0 1 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1
 1 1 1 0 0 0 0 1 0 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 0 1 0 1 1 1 1 1 0 1 0 1
 1 1 1 1 1 0 0 1 0 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 0
 0 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 0 1 0 1 1 1 0 0
 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 1 0 0 1 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1 0
 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0]
```

```
In [6]: num_cols = [col for col in X.columns if X[col].dtype in ['int64', 'float64']]
cat_cols = [col for col in X.columns if X[col].dtype == 'object']
print(num_cols)
print(cat_cols)

['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount', 'Loan_Amount_Term', 'Credit_History']
['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Property_Area']
```

```
In [7]: num_transformer = Pipeline(steps=[
    ('impute', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

cat_transformer = Pipeline(steps=[
    ('impute', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore', sparse=False))
])

preprocessor = ColumnTransformer(transformers=[
    ('num', num_transformer, num_cols),
    ('cat', cat_transformer, cat_cols)
])
```

```
In [8]: def train_predict(full_pipeline, train_X, train_y, test_X, test_y):
    full_pipeline.fit(train_X, train_y)
    pred_y = full_pipeline.predict(test_X)
    print('accuracy_score:', accuracy_score(test_y, pred_y))
    print('classification_report:', classification_report(test_y, pred_y))

    print('cross_val_score:')
    scores = cross_val_score(full_pipeline, X, y, cv=5, scoring='accuracy')
    print(scores)
    print('mean of scores:', scores.mean())
```

```
In [14]: from xgboost import XGBClassifier

models = [
    ('XGB', XGBClassifier(learning_rate=0.1,
                          n_estimators=100,
                          max_depth=6,
                          min_child_weight = 1,
                          gamma=0.,
                          subsample=0.8,
                          scale_pos_weight=1,
                          random_state=27)),
]

for model_name, model in models:
    print('\nModel %s.' % model_name)
    full_pipeline = Pipeline(steps=[
        ('preprocessor', preprocessor),
        ('model', model)
    ])
    train_predict(full_pipeline, train_X, train_y, test_X, test_y)
    print('*' * 50)
```

```

])
train_predict(full_pipeline, train_X, train_y, test_X, test_y)
print('*' * 50)

```

```

Model XGB.
accuracy_score: 0.8211382113821138
classification_report:

```

			precision	recall	f1-score	support
	0	0.70	0.58	0.63		33
	1	0.85	0.91	0.88		90
accuracy			0.82			123
macro avg		0.78	0.74	0.76		123
weighted avg		0.81	0.82	0.82		123

```

cross_val_score:
[0.74796748 0.73170732 0.75609756 0.78861789 0.78688525]
mean of scores: 0.7622550979608156
*****

```

```

In [15]: import pickle

pickle.dump(full_pipeline, open('Decision_Tree_Model.pkl', 'wb'))

```

```

In [16]: predicted = full_pipeline.predict(pd.DataFrame(columns=['Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome']))

```

```

In [17]: predicted

```

```

Out[17]: array([0])

```

```

In [18]: if(predicted == 1):
          print('loan_approved')
        else:
          print("Not_approved")

```

```

Not_approved

```

Home.html


```

1  <!DOCTYPE html>
2  <html lang="en">
3  <title>Welcome To This prediction</title>
4  <head>
5
6      <meta charset="UTF-8">
7      <meta http-equiv="X-UA-Compatible" content="IE=edge">
8      <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10     <link rel="preconnect" href="https://fonts.googleapis.com">
11     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12     <link href="https://fonts.googleapis.com/css2?family=Poppins&display=swap" rel="stylesheet">
13     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/T
14     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-u10knCvkvV5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7Nn1kvbZlHgTP00mMi466
15     <script src="https://cdn.jsdelivr.net/npm/popper.js/core@2.11.6/dist/umd/popper.min.js" integrity="sha384-o8qDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAe8AsjFuCZSmKbSSUnQlmmh/jp3"
16     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.min.js" integrity="sha384-7VPbUDkoPSGFnVtYi0QogXtr74QeVeeIs99Qf5YCF+TidwNldjvaKZX19NZ/e6oz" cro
17     <title>Index Page</title>
18     <link rel="stylesheet" href="../static/css/home.css">
19
20 </head>
21 <style>
22     *{
23         font-family:'Poppins', sans-serif;;
24     }
25
26     .card{
27         padding-bottom: 40px;
28         border-radius: 1rem;
29
30     }

```

```

30     }
31 </style>
32
33 <body>
34
35 <nav class="navbar fixed-top">
36
37     <div class="container">
38         <a class="navbar-brand" href="#">HOME</a>
39     </div>
40
41     <div >
42         <nav class="navmenu">
43             <li class="nav-item">
44                 <a class="nav-link active" aria-current="page" href="AccountRegister.html">Sign Up</a>
45             </li>
46             <li class="nav-item">
47                 <a class="nav-link active" aria-current="page" href="about.html">About</a>
48             </li>
49         </nav>
50     </div>
51 </nav>
52     <div class="ripple-background"></div>
53     <div class="circle xxlarge shade1"></div>
54     <div class="circle xlarge shade2"></div>
55     <div class="circle large shade3"></div>
56     <div class="circle mediun shade4"></div>
57     <div class="circle small shade5"></div>
58
59 <br>
60 <!-- FIRST ROW -->

```

```

60 <!-- FIRST ROW -->
61 <div class="row" style="height: 640px;">
62
63     <!-- LEFT COLUMN FOR PICTURE -->
64
65     <div class="col-6">
66         <div class="second">
67             <div class="card text-start" style="height:380px; width:780px;">
68                 <div class="card-body text-start">
69                     <h1 style=" font-size:50px;height:40px"><b>PREDICT THE LOAN</b></h1><br>
70                     <p style="font-size:28px;">Predict and automate the loan eligibility process based on customer detail provided while filling online application form </
71                 </div>
72                 <div class="card-body-end text-center">
73                     <div class="form-group text-center">
74                         <div>
75                             <form action="/login">
76                                 <input class="btn btn-primary p-2" type="submit" value="Lets Start">
77                             </form>
78                             <!-- <button > </button>
79                             <a href="../../templates/" class="btn btn-primary p-2" style="height:50px; width: 190px;"><h5 align="center">Lets Start</h5></a> -->
80                         </div>
81                     </div>
82                 </div>
83             </div>
84         </div>
85
86     </div>
87
88     <div class="col-2">
89
90
91 </div>
92 <div class="col-4">
93     <div class="text-end">
94         
95     </div>
96
97 </div>
98
99 </div>
100
101 <!-- RIGHT COLUMN FOR FORM -->
102
103
104 </div>
105
106
107
108
109 </body>
110
111 </html>

```

Home.css:

```

1  @import url('https://fonts.googleapis.com/css2?family=Poppins&display=swap');
2
3  *{
4      margin: 0;
5      padding: 0;
6      font-family: 'Poppins', sans-serif;
7  }
8
9  body
10 {
11     font-size: 20px;
12 }
13
14 .navbar
15 {
16     display: flex;
17     flex-direction: column;
18     justify-content: space-between;
19     padding: 7px;
20     background-color: #000000;
21     display: flex;
22     height: 70px;
23     color: white;
24     font-size: large;
25     border-radius: 0%;
26     margin-bottom: 0px;
27 }
28
29 .navbar-brand
30 {
31     font-size: large;
32     color: white;
33     cursor: pointer;
34 }
35
36 .circle{
37     position: fixed;
38     border-radius: 50%;
39     background: #FF5858;
40     animation: ripple 15s infinite;
41     box-shadow: 0px 0px 1px 0px #508fb9;
42 }
43
44 .small{
45     width: 100px;
46     height: 100px;
47     left: -10px;
48     bottom: -50px;
49 }
50
51 .medium{
52     background: #2c62f5;
53     width: 200px;
54     height: 200px;
55     left: -100px;
56     bottom: -100px;
57 }
58
59 .large{
60     width: 300px;
61     height: 300px;

```

```
61     height: 300px;
62     right: -150px;
63     bottom: -150px;
64 }
65
66 .xlarge{
67     background: #2c62f5;
68     width: 400px;
69     height: 400px;
70     left: 500px;
71     bottom: -330px;
72 }
73
74 .xxlarge{
75     width: 250px;
76     height: 270px;
77     left: -170px;
78     bottom: -10px;
79 }
80
81 .shade1{
82     opacity: 0.2;
83 }
84 .shade2{
85     opacity: 0.5;
86 }
87
88 .shade3{
89     opacity: 0.7;
90 }
```

```
92     .shade4{
93         opacity: 0.8;
94     }
95
96     .shade5{
97         opacity: 0.9;
98     }
99
100     .btnn{
101         font-size: 80vh;
102     }
103
104     li
105     {
106         overflow-x: visible;
107     }
108
109
110     .navmenu
111     {
112         color: white;
113         display: flex;
114         flex-direction: row;
115         gap: 50px;
116         list-style: none;
117         padding: 15px 50px 0px 0px;
118         overflow: visible;
119     }
120
121     .navmenu li a
122     {
```

```
122 {
123     transition: all 0.3s ease 0s;
124 }
125
126 .navmenu li a:hover
127 {
128     color: #8AB4F8;
129 }
130
131 .navbar-brand
132 {
133     color: white;
134     padding: 15px 60px;
135 }
136
137 .row
138 {
139
140     padding-top: 4px;
141     margin: 0;
142     display: flex;
143     align-items: center;
144 }
145
146 .second
147 {
148     border-radius: 0px;
149     overflow-x: visible;
150     box-shadow: 2px 7px 8px 6px rgba(0,0,0,0);
151     width: 400px;
152     color: #000000;
```

```

149     overflow-x: visible;
150     box-shadow: 2px 7px 8px 6px rgba(0,0,0,0);
151     width: 400px;
152     color: #000000;
153 }
154
155 .jus
156 {
157     display: flex;
158     justify-content: center;
159     align-items: center;
160 }
161
162 .label{
163     color:black;
164     font-weight: 400;
165     font-size: 20px;
166 }
167
168 .form-control
169 {
170     justify-content: center;
171 }
172
173 .card
174 {
175     color: black;
176     padding: 10px;
177     box-shadow: -8px 13px 12px -5px rgba(0,0,0,0.63);
178 }
179

```

Login.html:


```
login.html U X
templates > login.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title> Login </title>
6   <!-- <link rel="stylesheet" href="{{ url_for('static', filename='font.css') }}" -->
7
8   <link rel="stylesheet" href="../static/css/font.css">
9 </head>
10 <body><br><br><br><br>
11 <div align="center" class="first">
12   <div align="center" class="border" +>
13     <div class="header">
14       <h1 class="word">Login</h1>
15     </div><br><br><br>
16     <h2 class="word">
17       <form action="{{ url_for('login') }}" method="post">
18         <div class="msg">{{ msg }}</div>
19         <input id="username" name="username" type="text" placeholder="Enter Your Username" class="t
20         <input id="password" name="password" type="password" placeholder="Enter Your Password" clas
21         <input type="submit" class="btn" value="Sign In"><br><br>
22       </form>
23     </h2>
24     <p class="bottom">Don't have an account? <a class="bottom" href="{{ url_for('reg') }}"> Sign Up here
25   </div>
26 </div>
27 </body>
28 </html>
```

Discovering Python Interpreters Ln 1, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier 23:34 19/11/2022

Register.html:

```

reg.html U x
templates > reg.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title> Register </title>
6   <!-- <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}" -->
7   <link rel="stylesheet" href="../static/css/font.css">
8 </head>
9 <body>
10 </br></br></br></br></br>
11 <div align="center" class="first">
12   <div align="center" class="border">
13     <div class="header">
14       <h1 class="word">Register</h1>
15     </div></br></br></br></br>
16     <h2 class="word">
17       <form action="{{ url_for('reg') }}" method="post">
18         <div class="msg">{{ msg }}</div>
19         <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>
20         <input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></br>
21         <input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox"/></br></br>
22         <input type="submit" class="btn" value="Sign Up"></br>
23       </form>
24     </h2>
25     <p class="bottom">Already have an account? <a class="bottom" href="{{ url_for('login') }}"> Sign In here</a></p>
26   </div>
27 </div>
28 </body>
29 </html>

```

NotApproved.html :

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@500&display=swap"
rel="stylesheet">

```

```
</head>
```

```
<style>
```

```
body{
    font-family: 'Inter', sans-serif;
}
.popup{
background:#ffff;
width: 100%;
height: 100%;
position: absolute;
top: 0;
display: flex;
justify-content: center;
align-items: center;
text-align: center;

}
.pop-content{
    height: 250px;
    width: 500px;
    background:#FF5858;
    padding: 20px;
    border-radius: 5px;
    position: relative;

}

.content{
    font-weight: bold;
    margin-top: 70px;
}
.close{
    border-radius: 50px;
    width: 50px;
```

```
position: absolute;
top: -15px;
right: -15px;
background: #ffff;
cursor: pointer;
box-shadow: 6px 6px 29px --4px rgba(0,0, 0, 0.75);
```

```
}
.wrong{
  border-radius: 50%;
  width: 80px;
  position: absolute;
  left: 120px;
  margin-top: 20px;
```

```
}
```

```
</style>
```

```
<body>
```

```
  <div class="popup">
```

```
    <div class="pop-content">
```

```
      <h1>Loan Prediction</h1>
```

```
      
```

```
      <h3 class="content">Not Approved</h3>
```

```
    </div>
```

```
  </div>
```

```
</body>
```

```
</html>
```

Approved.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>

  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@500&display=swap"
rel="stylesheet">
</head>
<style>

  body{
    font-family: 'Inter', sans-serif;
  }
  .popup{
background:#ffff ;
width: 100%;
height: 100%;
position: absolute;
top: 0;
display: flex;
justify-content: center;
align-items: center;
text-align: center;

}
.pop-content{
```

```

height: 250px;
width: 500px;
background:#FF5858;
padding: 20px;
border-radius: 5px;
position: relative;

}

.content{
    font-weight: bolder;
    margin-top: 70px;
    font-size: x-large;
}
.correct{
    border-radius: 50%;
    width: 80px;
    position:absolute;
    left: 120px;
    margin-top: 20px;

}

</style>
<body>
    <div class="popup">
        <div class="pop-content">
            <h1>Loan Prediction</h1>
            
            <h3 class="content">Approved</h3>

        </div>
    </div>

```

```
</body>  
</html>
```

GitHub and Project Demo Link

GitHub Link: [IBM-EPBL/IBM-Project-11593-1659335647: Smart Lender - Applicant Credibility Prediction for Loan Approval \(github.com\)](https://github.com/IBM-EPBL/IBM-Project-11593-1659335647: Smart Lender - Applicant Credibility Prediction for Loan Approval)

Demo Link: <https://youtu.be/0KcIXmVdvSM>