

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

TEAM ID : PNT2022TMID39687

A PROJECT REPORT

Submitted by

AISHVARYA G - 510819104001

SHAMILI DEVI N B - 510819104018

PRIYADHARSHINI V - 510819104016

PREMKUMAR A - 510819104015

PRAVEENKUMAR V - 510819104014

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

GANADIPATHY TULSIS JAIN ENGINEERING COLLEGE

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 decreases the approval time and decreases the risk associated with the loan. The aim of this project was to compare the various loan prediction models and show which is the best one with least amount of error and could be used by banks in real world to predict if the loan should be approved or not taking the risk factor in mind. After analysing all models Random forest model is selected as the best model for predicting the loan eligibility of the customer.

1.2 PURPOSE

- The smart lender project is used to predict whether a particular applicant is safe to provide loan.
- The entire process of verifying the customer characteristics will be automated by machine learning technology.
- 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 institutions 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 trust worthy 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.

AISHVARYA G

submitting the property documents	Agree to the terms and conditions of the bank	Notify the borrower on repayment date
captcha for security	Automatic recovery of interest rate based on the loan amount and purpose of loan	Generate unique id for each loan approval
get the location of the customer's Land property through google map	add digital signature on agreement	

SHAMILIDEVI N B

getting reviews	providing assumption and provision for applying loan	Give cash back on repayment on time
predict loan approval on annual income	verify their mobile number	check whether the customer has already taken the loan and repayed

PREMKUMAR A

Send repayment date and interest amount to Email mobile number	verify the email id of the customer	Increase the interest rate if they had not repayed the amount on time
charge amount	Add graduate certificate if the customer was graduated	upload the income certificate for verifying customer's income

PRIYADHARSHINI V

Can have an agreement with the borrower	calculate the tax amount	get the pan card details
display the loan amount explicitly based on their income and occupation		

PRAVEEN KUMAR V

get the adhar card details of the customer	Verify the birth date by their birth certificate	can get customer's image for verification
can provide user friendly UI		

Group ideas

verifying documents

submitting the property documents	upload the income certificate for verifying customer's income	Verify the birth date by their birth certificate
Add graduate certificate if the customer was graduated		

details for trusting customer

get the pan card details	get the adhar card details of the customer	get the location of the customer's Land property through google map
--------------------------	--	---

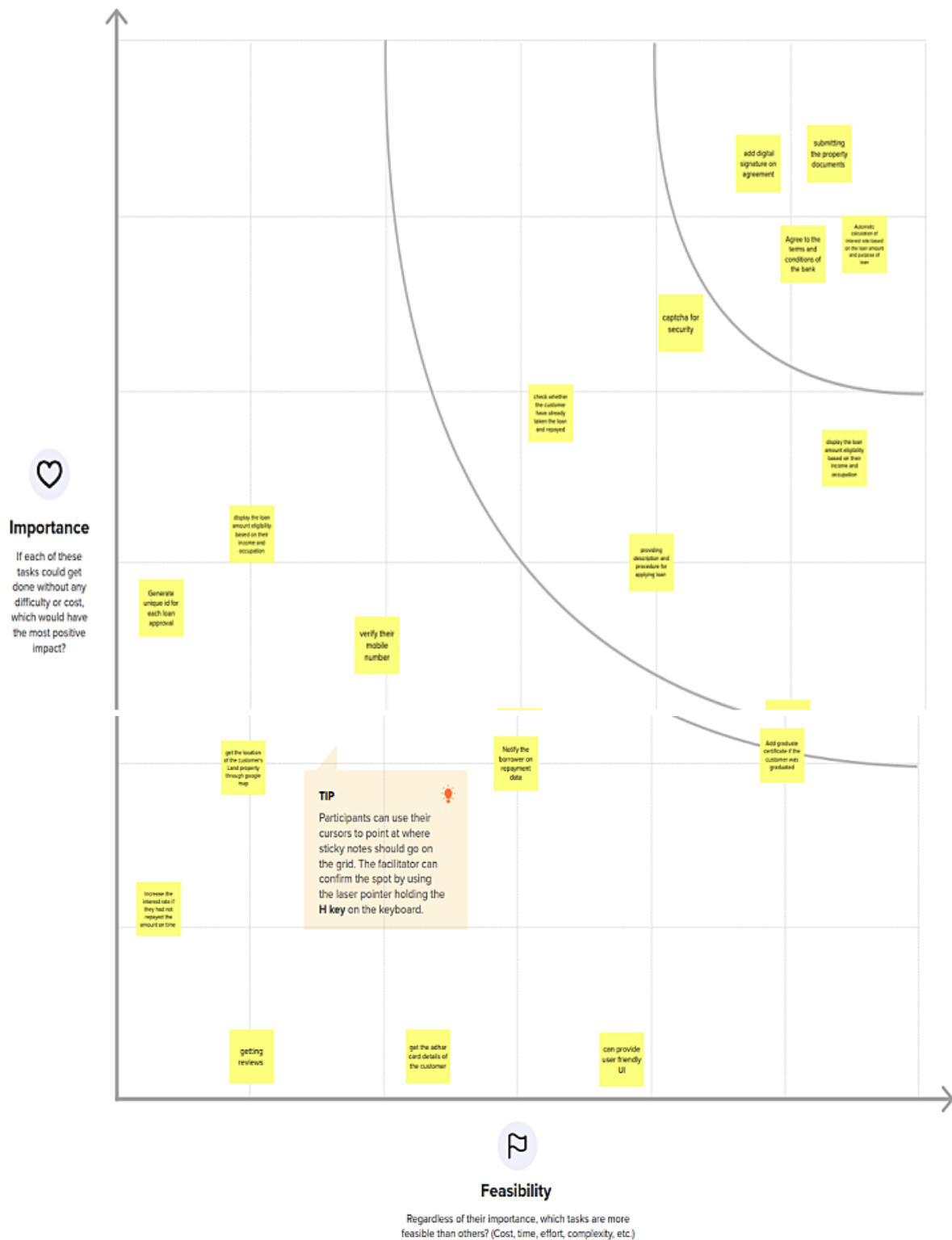
security and safety

Agree to the terms and conditions of the bank	Can have an agreement with the borrower	captcha for security
---	---	----------------------

User friendly features

can provide user friendly UI	Notify the borrower on repayment date	getting reviews
Give cash back on repayment on time		

Prioritizing ideas



3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement	A bank is a financial institution licensed to receive deposits and make loans needs a way to verify and trust the customer details and their documents for getting loan because they need an trustable customer with proper assets ,cash flow, documents and background who can repay the loan amount and interest on time.
2.	Idea/Solution description	<ul style="list-style-type: none">• The customer only need to enter the details,the loan approval status is then predicted automatically and quickly.• The property documents of the customer need to be submitted and the customer should agree to the terms and conditions of the bank• Provide captcha security• Varies efficient machine learning algorithms can be used to predict the loan eligibility of the customer.
3.	Novelty/Uniqueness	<ul style="list-style-type: none">• Provide customer ratings and reviews for understanding the customer.• Agreement of the terms and conditions.• Provides data security.The customer details will not be shared to the third party.• Instant Loan approval status..
4.	Social Impact	<ul style="list-style-type: none">• Easy and fast loan approval process for the customer.• Approves Loan to a trustable person.• Bank can find a genuine person to provide loan.• Secure storage of customer details.

5.	Business Model	<ul style="list-style-type: none"> • Can generate revenue through advertisement. • Can collaborate with many companies. • Can charge the processing fees and service fees from customer. • Can give pre payment option. • Can generate revenue by referencing.
6.	Scalability Of Solution	<ul style="list-style-type: none"> • It can be provided as software as a service. • Both borrower and Lender can use this software. • Any type of customer can predict their loan approval without any discrimination. • Can use this software anytime and anywhere. • This system is easily scalable and efficient. • Easy and user friendly software for all.

3.4 PROBLEM SOLUTION FIT

Define customer segments, fit into customer limitations

1. CUSTOMER SEGMENT(S)	6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES	5. AVAILABLE SOLUTIONS PLUSSES & MINUSES
<div>Add</div> <p>Salary Employee</p> <p>Business people</p> <p>Bank</p>	<div>Add</div> <p>Trusting issues</p> <p>Cibil score, report, sources</p> <p>Bankcard cheque</p> <p>As per customer noms</p> <p>Security issues</p>	<div>Add</div> <p>Assets properties</p> <p>Credit score insurance</p> <p>Manual verification process</p> <p>Delay in loan Approval</p>

Focus on problem, tap into behavior, understand root cause

2. PROBLEMS / PAINS + ITS FREQUENCY	9. ROOT / CAUSE OF PROBLEM	7. BEHAVIOR + ITS INTENSITY
Cheque bounce	Cheque bounce	Positive approach before the loan
EMI not paid by the customer	Providing wrong address	Negative approach after getting loan
Missing customer	Duplicate certification	Confusion to bank in approving loan
Bad debts	Not present in address	
Not repaying the loan amount	Providing wrong address	
Wrong customer details		

Identify strong triggers & emotions

3. TRIGGERS	10. YOUR SOLUTION	8. CHANNELS OF BEHAVIOR
Online advertisement	Verifying the property documents	ONLINE
Continuous calling of the customer	Agreement with customer	Verifying documents in online
Brainwash	Ratings and reviews	OFFLINE
	Secure data storage	Verifying documents in online
4. EMOTIONS		
Harsh behavior		
Threatening		
Confusion and fear		

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	User Registration	Registration through Form
FR-2	User Login	Login using username and password
FR-3	Bank Login	Login using bank credentials
FR-4	User View Procedure	User will find the complete details and steps for applying loan
FR-5	User Application	User upload their details and document proof
FR-6	Contact	User can contact the bank in case of any queries
FR-7	Loan prediction	Predicts the eligibility of loan
FR-8	Credit Verification	Bank verifies the user details and documents and approves loan
FR-9	Display status	User view loan approval status

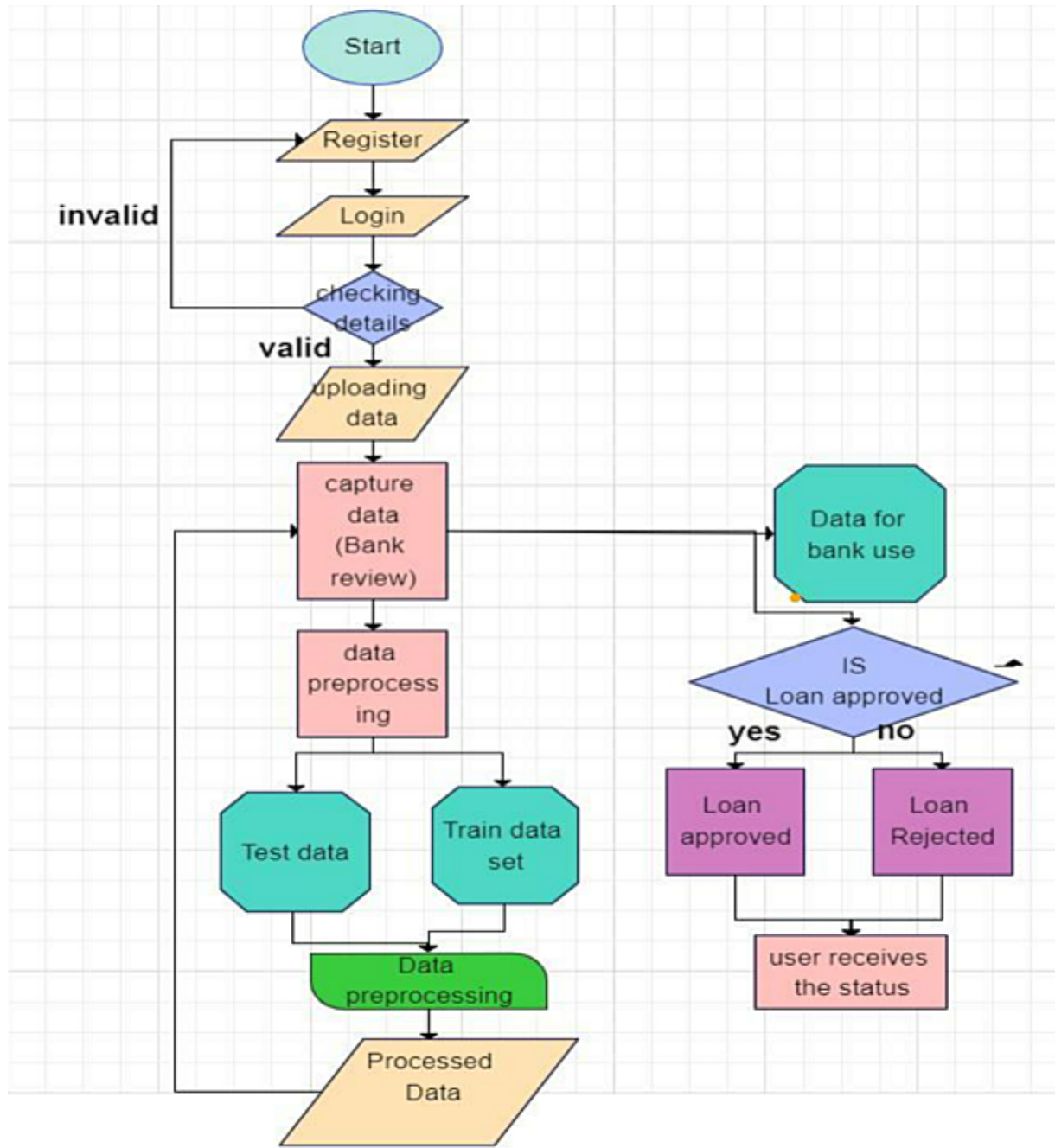
4.2 NON-FUNCTIONAL REQUIREMENT

Following are the non-functional requirements of the proposed solution

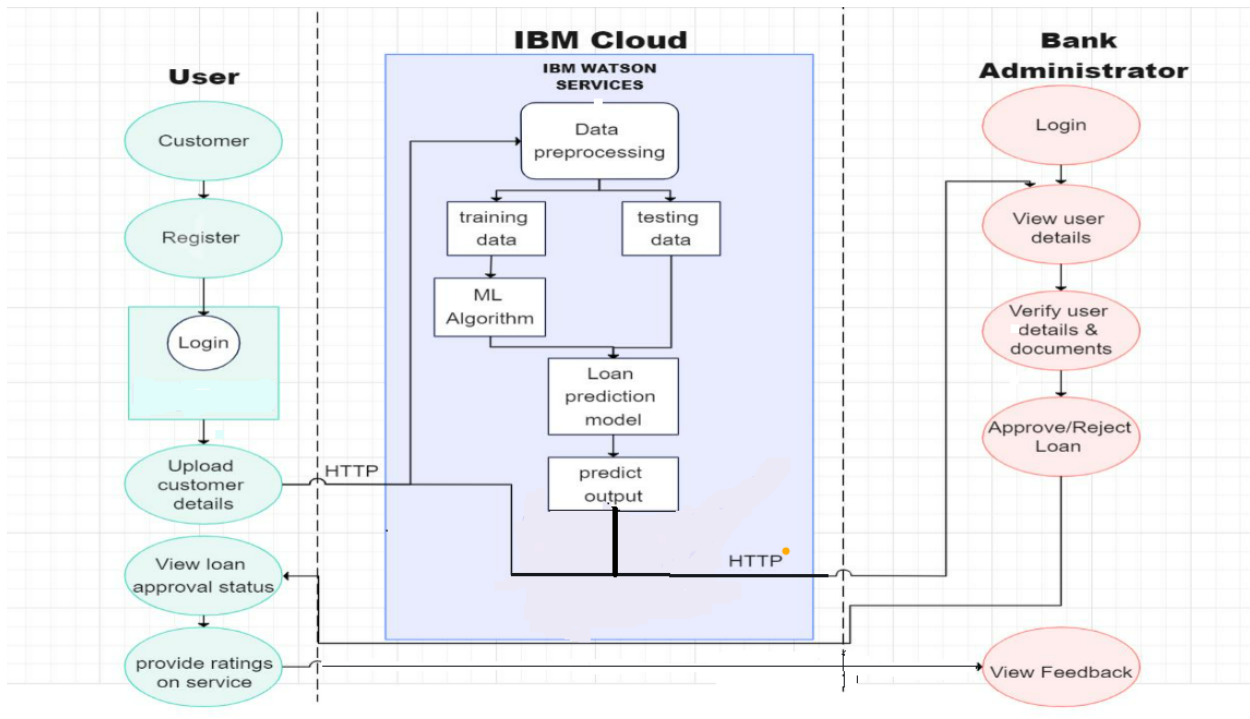
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none"> • User friendly UI • Easy navigation • Easy to use
NFR-2	Security	<ul style="list-style-type: none"> • Login verification • User details not shared to third party • Verification by bank • Captcha Security
NFR-3	Reliability	<ul style="list-style-type: none"> • Handles changes • Fault recovery • Data backup
NFR-4	Performance	<ul style="list-style-type: none"> • Responsive website • Quick loading
NFR-5	Availability	<ul style="list-style-type: none"> • Good service Availability • Satisfies user needs • Meets user requirements
NFR-6	Scalability	<ul style="list-style-type: none"> • The system can grow without negative influence • Adaptable • Handles multiple users

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



Components & Technologies

S.No	Component	Description	Technology
1.	User Interface	Web UI- Customer UI, Bank Admin UI, Loan approval UI	HTML, CSS, JavaScript
2.	Registration	User register with their credentials	HTML, CSS, JavaScript, PHP, Bootstrap
3.	Login	User login with their username and password and bank login with their credentials	HTML, CSS, JavaScript

4.	Upload user details and documents	The customer will upload their personal details, Loan details and documents	HTML,CSS,JavaScript
5.	Contact	The customer can contact the bank through contact page in case of any queries	Javascript
6.	Bank Verification	Bank will verify the details of the customer and documents and can resolve the customer queries	HTML,CSS, javascript
7.	Loan Approval status	The Loan prediction model gives the eligibility of loan and the bank admin will give the loan approval status	Python, JavaScript
8.	Ratings and Reviews	User provide their feedback	JavaScript
9.	Database	Datatype- String, Integer, Float, Boolean RDBMS database Stores user details, review details, document details	MySQL
10.	Database	Platform as a service Used to store, analyze and retrieve the data The contact information of the customer is stored	MYSQL Googlesheet
11.	File Storage	Network File Storage	IBM Cloud object storage, IBM Watson studio

12.	External API	To store the user feedback and contact	Google sheet
13.	Machine Learning Model	To predict the Loan approval status of the user	Python, IBM Watson Service
14.	Infrastructure	Cloud Server Configuration: Speed Feature flag Central configuration storage Virtual processors 2vcpu RAM 8GB Storage100GB Local server:Wamp64 server RAM:4GB Storage:456GB Platform:Windows 10	IBM Cloud Server Wamp64 server

Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Bootstrap Flask Scikit, NumPy	HTML, CSS JavaScript Python
2	Security Implementations	Captcha Security	JavaScript
3	Scalable Architecture	3-tier architecture	HTML, CSS, JavaScript Python
4	Availability	Single server	IBM cloud server

5	Performance	More Accurate prediction and Cookie Free Domain	Python
---	-------------	---	--------

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Priority Release
Customer (web user, mobile user)	Registration	USN-1	As a user, I can register for the application by entering my name, email ID, password, Account number etc.	I can access my account	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering the user's name and password	I can login to the website	High	Sprint-1
	Navigation	USN-3	As a user, I can navigate to different tabs like home, description, contactt	I can navigate and view different details	Low	Sprint-2
	View procedure	USN-4	As a user, I can view the procedure to apply for loan	I can view complete steps to apply for loan	Medium	Sprint-2
	Contact	USN-5	As a user, I can contact bank	I can clarify my doubtts	Low	Sprint-2
	Ratings	USN-6	As a user, I can provide ratings for the service Provided	I can give my review for the service	Medium	Sprint-3
Bank administrator	View user details	USN-1	As a Bank administrator, I can view the user details	I can view all the details uploaded by the customer	Medium	Sprin-3
	Credit verification	USN-2	As a Bank administrator, I can verify the credibility of the customer	I can verify all the details of the customer	High	Sprint-3
	Document Verification	USN-3	As a Bank administrator, I can verify all the document proof and ID proof of the customer	I can verify all the documents submitted by the customer	High	Sprint-4
	Loan approval Status	USN-4	As s Bank administrator, I can Approve/Reject the loan for the customer based on them details	I can give the loan approval status of the customer	High	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	AISHVARYA G SHAMILIDEVI N B
Sprint-1	Login	USN-2	As a user, I can log into the application by entering the user name and password.	2	High	AISHVARYA G SHAMILIDEVI N B
Sprint 1		USN-3	As a user,I can log in using Gmail.	1	Low	AISHVARYA G SHAMILIDEVI N B
Sprint 2	Upload details	USN-4	As a user,I can upload my details and documents.	3	High	AISHVARYA G
Sprint-2	Navigation	USN-5	As a user, I can navigate to different tabs like home, description, contact,login,procedure.	2	Low	PREMKUMAR A PRIYADHARSHINI V
Sprint-2	View procedure	USN-6	As a user, I can view the procedure to apply for loan.	1	Medium	PREMKUMAR A PRIYADHARSHINI V
Sprint-2	Contact	USN-7	As a user, I can contact bank.	1	Low	PREMKUMAR A PRIYADHARSHINI V
Sprint-3	Ratings	USN-8	As a user, I can provide ratings for the service Provided.	2	Medium	AISHVARYA G SHAMILIDEVI N B
Sprint-3	View user details	USN-9	As a Bank administrator , I can view the user details.	2	Medium	PRAVEEN KUMAR V AISHVARYA G
Sprint-3	Credit verification	USN-10	As a Bank administrator, I can verify the credibility of the customer.	3	High	PRAVEEN KUMAR V SHAMILIDEVI N B
Sprint-4	Document Verification	USN-11	As a Bank administrator,I can Verify all the documents proof and ID proof of the customer.	3	High	AISHVARYA G SHAMILIDEVI N B
Sprint-4	Loan approval status	USN-12	As a Bank administrator, I can Approve/Reject the loan for the customer based on	3	High	AISHVARYA G PREMKUMAR A

			their details.			
Sprint-4		USN -13	As a user, I can get confirmation of loan approval through email. .	3	High	SHAMILIDEVI N B AISHVARYA G

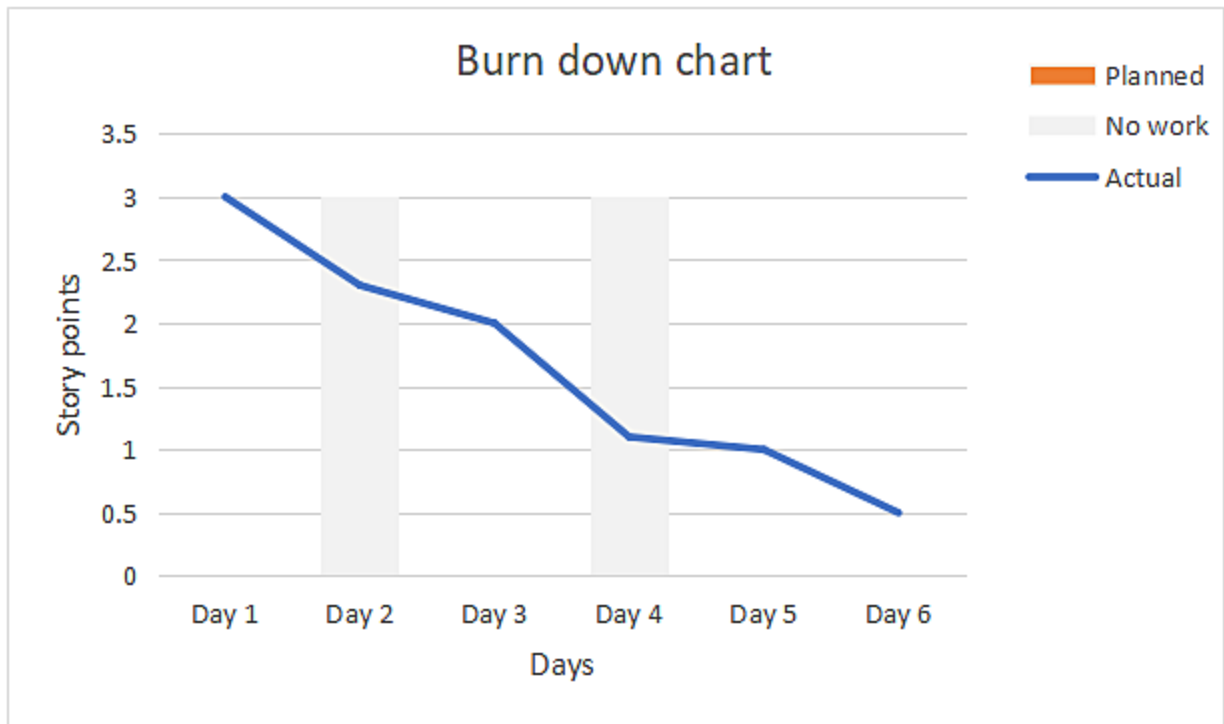
6.2 SPRINT DELIVERY SCHEDULE

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	09 Nov 2022
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	6	17 Nov 2022

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







AVERAGE VELOCITY=6/6=1










Burndown Chart



6.3 REPORTS FROM JIRA

	22	23	24	25	26	27	28	29
	OCT							
Sprints			SACPFLA Sprint 1					
> SACPFLA-18 Registration								
> SACPFLA-18 Login								

	NOV					
	31	1	2	3	4	5
Sprints	SACPFLA Sprint 2					
>  <u>SACPFLA-1 Registration</u> DONE						
>  <u>SACPFLA-18 Login</u> DONE						
v  <u>SACPFLA-20 Upload details</u> <input checked="" type="checkbox"/> <u>SACPFLA-21 As a user,I can...</u> DONE 5108191...						
v  <u>SACPFLA-22 Navigation</u> <input checked="" type="checkbox"/> <u>SACPFLA-23 As a user i can...</u> DONE 5108191...						
v  <u>SACPFLA-24 View procedure</u> <input checked="" type="checkbox"/> <u>SACPFLA-25 As a user, I can...</u> DONE 5108191...						
v  <u>SACPFLA-26 Contact</u> DONE <input checked="" type="checkbox"/> <u>SACPFLA-27 As a user, I can contact th...</u> DONE						

	NOV				NOV				NOV				NOV													
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Sprints	SACPFLA Sprint...																									
>  <u>SACPFLA-1 Registration</u> DONE																										
>  <u>SACPFLA-18 Login</u> DONE																										
>  <u>SACPFLA-20 Upload details</u>																										
>  <u>SACPFLA-22 Navigation</u>																										
>  <u>SACPFLA-24 View procedure</u>																										
>  <u>SACPFLA-26 Contact</u> DONE																										
v  <u>SACPFLA-28 Ratings</u> <input checked="" type="checkbox"/> <u>SACPFLA-29 As a user,I can provide rat...</u> DONE																										
v  <u>SACPFLA-30 View user details</u> <input checked="" type="checkbox"/> <u>SACPFLA-31 As a Bank administrator, I...</u> DONE																										
v  <u>SACPFLA-32 Credit verification</u> <input checked="" type="checkbox"/> <u>SACPFLA-33 As a Bank administrator,I c...</u> DONE																										

	NOV						NOV						DEC						DEC						DEC											
	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Sprints	SACPLA Sprint 4																																			
» SACPLA-1: Registration																																				
» SACPLA-15: Login																																				
» SACPLA-20: Upload details																																				
» SACPLA-22: Navigation																																				
» SACPLA-24: View procedure																																				
» SACPLA-26: Contact																																				
» SACPLA-28: Ratings																																				
» SACPLA-30: View user details																																				
» SACPLA-32: Credit verification																																				
» SACPLA-34: Document verification																																				
» SACPLA-35: As a Bank administrator, I c...																																				
» SACPLA-36: As a Bank administrator, I...																																				
» SACPLA-37: Loan approval status																																				
» SACPLA-38: As a user, I can get confir...																																				

7.CODING AND SOLUTIONING

7.1 Feature 1:Contact

The user can contact the bank in case of any queries through this contact feature .They should fill their name,email ID ,mobile number and the message they want to send to the

Code:

contact.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>contact</title>

<link rel="stylesheet" href="static/contact.css">

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

<script src='https://kit.fontawesome.com/a076d05399.js' crossorigin='anonymous'></script>

</head>
```

```

<body>

<a href="home.html"><button class="but">Back</button></a>

<div class="contact-section">

<div class="contact-info">

<div>Address : Vellore,Tamilnadu,India</div>

<div>Email : Bank@gmail.com</div>

<div>Mobile No : 9000033456</div>

<div>Working Hours : Mon - Fri 10:00 AM to 4:30 PM</div>

</div>

<div class="box">

<div class="title">

<h1>Contact Us</h1><br>

<h2>We are ready</h2>

</div>

<form action="" name="contact-us">

<input type="text" name="name" class="form-control" id="name" placeholder="Your Name" required><br>

<input type="text" name="phone" class="form-control" id="phone" placeholder="Your mobile Number" required><br>

<input type="email" name="email" class="form-control" id="email" placeholder="Your Email id" required><br>

<textarea name="message" class="form-control" id="message" rows="4" placeholder="Message"></textarea><br>

```

```

        <input type="submit" name="" class="form-control submit" value="SEND">

    </form>

</div>

</div>

<script>

    const scriptURL =
'https://script.google.com/macros/s/AKfychwRbYoFq0mx9RGsOWGGg3mUgBc4Yap6Zo9EIRAHaB5_TvvPICT4UXUkAXihqNOER0_gfA/ex
ec'

    const form = document.forms['contact-us']

    form.addEventListener('submit', e => {

        e.preventDefault()

        fetch(scriptURL, { method: 'POST', body: new FormData(form)})

        .then(response => console.log('Success!', response))

        .catch(error => console.error('Error!', error.message))

        form.reset()

        alert('Success!')

    })

</script>

</body>

```

```
</html>
```

contact.css:

```
*{  
  
    margin: 0;  
  
    padding: 0;  
  
    box-sizing: border-box;  
  
}  
  
body{  
  
    background-color:white;  
  
}  
  
.container{  
  
    width: auto;  
  
    margin: auto;  
  
}  
  
.icon{  
  
    width: 200px;  
  
    float: left;  
  
    height: 70px;  
  
}
```



```
ul{

    display: flex;

    list-style-type: none;

}

.contact-section{

    width: 100%;

    display: flex;

    justify-content: center;

    align-items: center;

}

.contact-info{

    justify-content: left;

    align-items: left;

    color: #ff226f;

    max-width: 500px;

    line-height: 65px;

    padding-left: 25px;

    font-size: 20px;

    font-weight: bold;
```

```
}

.contact-info i{

    margin-left: 20px;

    font-size: 25px;

}

.contact-info{

    flex: 1;

}

.title{

    text-align: center;

    margin-top: 100px;

    color: #fff;

}

.title h1{

    font-size: 32px;

    line-height: 10px;

}

.title h2{

    font-size: 16px;
```

```
}

form{

    margin-top: 50px;

    color: aqua;

}

.form-control{

    width: 600px;

    background: transparent;

    background-color: #fff;

    border: none;

    outline: none;

    border-bottom: 1px solid #ff226f;

    color:black;

    font-size: 18px;

    margin-bottom: 16px;

    padding: 10px;

    font-weight: 200px;

}

input{

    height: 45px;
```

```
}

form .submit{

  background-color: #ff226f;

  border-color: transparent;

  color: #fff;

  font-size: 20px;

  letter-spacing: 2px;

  height: 50px;

  margin-top: 20px;

  transition: 0.4s ease;

}

form .submit:hover{

  background-color:#ef3a79;

  cursor: pointer;

}

h1{

  color:#ff226f;

}

.but{

  background-color:#ff226f;
```

```
color: white;

padding: 10px;

margin:30px;

border-radius: 10px;

width:100px;

font-weight: bold;

}
```

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:

prediction.html:

```
<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

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

```
<!-- Bootstrap CSS -->

<link rel="stylesheet" href="static/prediction.css">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-eOJMYsd53ii+scO/bJGFiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

<link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">

<title>prediction</title>

</head>

<body>

<script>

function valid(){

var Ai=document.getElementById("ApplicantIncome").value;

var Co=document.getElementById("CoapplicantIncome").value;

var LA=document.getElementById("LoanAmount").value;

var LT=document.getElementById("Loan_Amount_Term").value;

if(Ai > 1000000000000000000000000000000000000000000){

    alert("Applicant income is too large enter a valid number")

return false;

}

if(Co > 1000000000000000000000000000000000000000000){

    alert("Coapplicant income is too large enter a valid number")
```

```
return false;

}

if(LA > 10000000000000000000000000000000000000000000000000000000){

    alert("Loan Amount is too large enter a valid number")

    return false;

}

if(LT > 10000000000000000000000000000000000000000000000000000000){

    alert("loan amount term is too large enter a valid number")

    return false;

}

var name=document.getElementById("Name").value;

var letters=/^[a-zA-Z]*$/;

if(!name.match(letters)){

    alert("Name must contain only alphabets")

    return false;

}

var num=/^[0-9]+$;/;

if(!Ai.match(num)){

    alert("Enter only valid numbers alphabets are not allowed ")

    return false;
```

```

    }

    if(!Co.match(num)){

        alert("Enter only valid numbers alphabets are not allowed ")

        return false;

    }

    if(!LA.match(num)){

        alert("Enter only valid numbers alphabets are not allowed ")

        return false;

    }

    if(!LT.match(num)){

        alert("Enter only valid numbers alphabets are not allowed ")

        return false;

    }

    var mo=document.getElementById("mon").value;

    var mn=/^[0-9]{10}$/;

    if(!mo.match(mn)){

        alert("Please enter only 10 digit mobile number")

        return false;

    }

}

```



```

</script>

<section class="text-gray-600 body-font">

<div class="container px-5 py-24 mx-auto">

<div class="flex flex-col text-center w-full mb-20">

<h1 class="Heading">LOAN ELIGIBILITY PREDICTION</h1><br>

<p class="fill">Fill the form for prediction</p>

</div>

<div>

</div>

</div>

<div class="mb-3">

<a class="btn btn-primary" href="/" id="back" role="button">Back</a></div>

<form action="/prediction.html" method="post" onsubmit="return valid()">

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Name</label>

<input type="text" class="form-control" id="Name" name="Name" placeholder="Enter your Name" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Email ID</label>

<input type="email" class="form-control" id="email" name="email" placeholder="Enter your Email ID" required >

```

```
</div>
```

```
<div class="mb-3">
```

```
<label for="exampleFormControlInput1" class="form-label">Mobile Number</label>
```

```
<input type="text" class="form-control" id="mon" name="mon" placeholder="Enter your Mobile number" required>
```

```
</div>
```

```
<div class="mb-3">
```

```
<label for="exampleFormControlInput1" class="form-label"> Gender</label>
```

```
<select class="form-select" id="gender" name="gender" aria-label="Default select example" required >
```

```
<option selected>-- select gender --</option>
```

```
<option value="Male">Male</option>
```

```
<option value="Female">Female</option>
```

```
</select>
```

```
</div>
```

```
<div class="mb-3">
```

```
<label for="exampleFormControlInput1" class="form-label"> Married status</label>
```

```
<select class="form-select" id="married" name="married" aria-label="Default select example" required >
```

```
<option selected>-- select married status --</option>
```

```
<option value="Yes">Yes</option>
```

```
<option value="No">No</option>
```

```
</select>
```

```

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Dependents</label>

  <select class="form-select" id="dependents" name="dependents" aria-label="Default select example" required>

    <option selected>-- select dependents --</option>

    <option value="0">0</option>

    <option value="1">1</option>

    <option value="2">2</option>

    <option value="3+">3+</option>

  </select>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Education</label>

  <select class="form-select" id="education" name="education" aria-label="Default select example" required>

    <option selected>-- select education --</option>

    <option value="Graduate">Graduate</option>

    <option value="Not Graduate">Not Graduate</option>

  </select>

</div>

<div class="mb-3">

```

```

<label for="exampleFormControlInput1" class="form-label">Self_Employed</label>

<select class="form-select" id="employed" name="employed" aria-label="Default select example" required>

<option selected>-- select Self_Employed --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Credit_History</label>

<select class="form-select" id="credit" name="credit" aria-label="Default select example" required >

<option selected >-- select Credit_History --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Property_Area</label>

<select class="form-select" id="proparea" name="proparea" aria-label="Default select example" required>

<option selected>-- select Property_Area --</option>

```

```

<option value="Semiurban">Semiurban</option>

<option value="Urban">Urban</option>

<option value="Rural">Rural</option>

</select>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Enter ApplicantIncome</label>

  <input type="text" class="form-control" id="ApplicantIncome" name="ApplicantIncome" placeholder="ApplicantIncome" required>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Enter CoapplicantIncome</label>

  <input type="text" class="form-control" id="CoapplicantIncome" name="CoapplicantIncome" placeholder="CoapplicantIncome"
required>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Purpose of loan</label>

  <select class="form-select" id="pur" name="pur" aria-label="Default select example" required>

    <option selected>-- select the purpose of loan --</option>

    <option value="person">Personal loan</option>

```

```

<option value="Bussiness">Bussiness loan</option>

<option value="Education">Education loan</option>

<option value="Home">Home loan</option>

<option value="Other">other</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter LoanAmount</label>

<input type="text" class="form-control" id="LoanAmount" name="LoanAmount" placeholder="LoanAmount" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Loan_Amount_Term</label>

<input type="text" class="form-control" id="Loan_Amount_Term" name="Loan_Amount_Term" placeholder="Loan_Amount_Term"
required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Adhar Number</label>

<input type="text" class="form-control" id="Adhar" name="Adhar" placeholder="Adhar Number" required >

</div>

<div class="mb-3">

```

```

<label for="exampleFormControlInput1" class="form-label">Enter PAN card ID</label>

<input type="text" class="form-control" id="PAN " name="PAN " placeholder="PAN card ID" required>

</div>

<div class="mb-3">

<label for="property document" class="form-label">Property Document</label><br><input type="file" required >

</div>

<div class="mb-3">

<label for="Govt ID proof" class="form-label">Govet ID proof</label><br><input type="file" required>

</div>

<div class="mb-3">

<input type="checkbox" required>

I accept the <a href="terms.html">Terms and conditions</a>

</div>

<br><br>

<div class="mb-3">

<input type="submit" class="but" value="PREDICT">

</div>

</form>

</div>

</section>

```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLtS3qm9Ekf" crossorigin="anonymous"></script>

</body>

</html>
```

prediction.css:

```
body{

    margin:100px;

}

.form-label{

    color:rgb(235, 27, 97);

    font-weight: bolder;

    font-size: 18px;

}

.Heading{

    color:rgb(235, 27, 97);

    font-size: 30px;

    font-weight:bolder;
```



```
padding: 10px;

margin: 20px;

font-family: 'Times New Roman', Times, serif;

width: 100%;

}

.fill{

color: black;

font-size: 20px;

font-weight: 15px;

font-family: 'Times New Roman', Times, serif;

}

#but{

text-align: center;

align-items: center;

justify-content: center;

}

.mb-3{

width: 85%;

padding-left: 15%;
```

```
}

.form-control{

    margin:10px 2px;

    outline: solid 1px lightblue;

}

.form-select{

    margin:10px 2px;

    outline: solid 1px lightblue;

}

.but{

    background-color:rgb(235, 27, 97);

    color:white;

    border-radius: 10px;

    width:100px;

    border-color:white ;

    height:50px;

    padding: 10px;

}

#back{
```

```
background-color:rgb(235, 27, 97);

color:white;

border-radius: 10px;

width:100px;


}
```

7.3 FEATURE 3 :Rating and Reviews

The user can provide their feedback at last about the service so that the bank can improve their service.

Code:

```
<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Loan approval status</title>

<link rel="stylesheet" href="static/approve.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

<style>
```

```
@import url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap');
```

```
*{
```

```
margin: 0;
```

```
padding: 0;
```

```
box-sizing: border-box;
```

```
font-family: 'Poppins', sans-serif;
```

```
}
```

```
html,body{
```

```
display: grid;
```

```
height: 100%;
```

```
place-items: center;
```

```
text-align: center;
```

```
background:white;
```

```
}
```

```
.container{
```

```
position: relative;
```

```
width: 400px;
```

```
background:black;
```

```
padding: 20px 30px;
```

```
border: 1px solid #444;
```

```
border-radius: 5px;

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

}

.container .post{

display: none;

}

.container .text{

font-size: 25px;

color: #666;

font-weight: 500;

}

.container .edit{

position: absolute;

right: 10px;

top: 5px;

font-size: 16px;

color: #666;
```

```
font-weight: 500;

cursor: pointer;

}

.container .edit:hover{

text-decoration: underline;

}

.container .star-widget input{

display: none;

}

.star-widget label{

font-size: 40px;

color: #444;

padding: 10px;

float: right;

transition: all 0.2s ease;

}

input:not(:checked) ~ label:hover,

input:not(:checked) ~ label:hover ~ label{

color: #fd4;

}

}
```

```
input:checked ~ label{

    color: #fd4;

}

input#rate-5:checked ~ label{

    color: #fe7;

    text-shadow: 0 0 20px #952;

}

#rate-1:checked ~ form header:before{

    content: "I just hate the service";

}

#rate-2:checked ~ form header:before{

    content: "I don't like the service ";

}

#rate-3:checked ~ form header:before{

    content: "Awesome service";

}

#rate-4:checked ~ form header:before{

    content: "Satisfied service";

}

#rate-5:checked ~ form header:before{
```

```
content: "Excellent service ";
```

```
}
```

```
.container form{
```

```
display: none;
```

```
}
```

```
input:checked ~ form{
```

```
display: block;
```

```
}
```

```
form header{
```

```
width: 100%;
```

```
font-size: 25px;
```

```
color: #fe7;
```

```
font-weight: 500;
```

```
margin: 5px 0 20px 0;
```

```
text-align: center;
```

```
transition: all 0.2s ease;
```

```
}
```

```
form .textarea{
```

```
height: 100px;
```

```
width: 100%;
```



```
overflow: hidden;

}

form .textarea textarea{

height: 100%;

width: 100%;

outline: none;

color: #eee;

border: 1px solid #333;

background: #222;

padding: 10px;

font-size: 17px;

resize: none;

}

.textarea textarea:focus{

border-color: #444;

}

form .btn{

height: 45px;

width: 100%;

margin: 15px 0;
```

```
}

form .btn button{

height: 100%;

width: 100%;

border: 1px solid #444;

outline: none;

background: #222;

color: #999;

font-size: 17px;

font-weight: 500;

text-transform: uppercase;

cursor: pointer;

transition: all 0.3s ease;

}

form .btn button:hover{

background: #1b1b1b;

}

h1{

background:lightpink;

width:1500px;
```

```
margin:none;

color: rgb(99, 22, 23);

padding-top: 20px;

height:100px;

margin-top:none;

border: solid 5px rgb(99, 22, 23);

}

h2{

color:rgb(42, 117, 7);

font-size: 40px;

font-family: 'Times New Roman', Times, serif;

}

h3{

font-size:30px;

color:rgb(246, 187, 11);

}

</style>

</head>
```

```

<body>

<h1>LOAN APPROVAL STATUS</h1>

<h2>{{ prediction_text }}</h2>



<h3>Please provide your feedback</h3>

<div class="container">

    <div class="post">

        <div class="text">Thanks for rating us!</div>

        <div class="edit">EDIT</div>

    </div>

    <div class="star-widget">

        <input type="radio" name="rate" id="rate-5">

        <label for="rate-5" class="fas fa-star"></label>

        <input type="radio" name="rate" id="rate-4">

        <label for="rate-4" class="fas fa-star"></label>

        <input type="radio" name="rate" id="rate-3">

        <label for="rate-3" class="fas fa-star"></label>

        <input type="radio" name="rate" id="rate-2">

        <label for="rate-2" class="fas fa-star"></label>

```

```

<input type="radio" name="rate" id="rate-1">

<label for="rate-1" class="fas fa-star"></label>

<form action="#">

  <header></header>

  <div class="textarea">

    <textarea cols="30" placeholder="Describe your experience.."></textarea>

  </div>

  <div class="btn">

    <button type="submit">Post</button>

  </div>

</form>

</div>

</div>

<script>

const btn = document.querySelector("button");

const post = document.querySelector(".post");

const widget = document.querySelector(".star-widget");

const editBtn = document.querySelector(".edit");

btn.onclick = ()=>{

  widget.style.display = "none";

```

```

post.style.display = "block";

editBtn.onclick = ()=>{

    widget.style.display = "block";

    post.style.display = "none";

}

return false;

}

</script>

</body>

</html>

```

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: aaa@gmail.com password: aaishu	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: aaaa password:aishu	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: aaaa@gmail.com password: aishu	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: aaa@gmail.com password: aishu valid captcha:1xc9830 entered captcha:1xc9830	Application should login to prediction page	pass
LoginPage_TC_O10	Functional		Username: aaa@gmail.com password: aishu 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.user name text box d.password text box e.Mobile number text box f.Register button	pass
Registration_TC_O13	Functional	Verify user is able to register into application with Valid credentials	Name:aaaa emailID:aaa@gmail.com username:aaa password:aishu mobile number: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

Procedure_TC_015	Functional	Verify user is able to see the Procedure page when clicked on Procedure button	procedure.html	Procedure page should be displayed	pass
Procedure_TC_016	Functional	Verify the UI elements in Register page	procedure.html	Application should show below UI elements: a. Back button b. An Image c. content on procedure	pass
About_TC_017	Functional	Verify user is able to see the About us page when user clicked on About page	about.html	About page should be displayed	pass
About_TC_018	UI	Verify the UI elements in about page	about.html	Application should show below UI elements: a. Back button b. An Image c. content about the loan prediction project	pass
Contact_TC_019	Functional	Verify user is able to see the Contact us page when user clicked on contact button	contact.html	Contact page should be displayed	pass
Contact_TC_020	UI	Verify the UI elements in contact page	contact.html	Application should show below UI elements: a. Back button b. Name text box c. Email ID text box d. Mobile number e. Message text area f. send button	pass
Contact_TC_021	Functional	Verify whether user is able to send their message with valid credentials	Name:aaaa emailID:aaa@gmail.com mobile number:9089097878 Message:Hi	Application should give an alert message of success and the provided data should be stored in the google sheet	pass

Contact_TC_022	Functional	Verify whether user is able to send their message with invalid credentials	Name:aaaa emailID:aaagmail.com mobile number:9089097878 Message:Hi	Application should display 'please include @ symbol in emailId'	pass
Prediction_TC_023	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_024	UI	Verify the UI elements in prediction page	prediction.html	Application should show below UI elements: 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. Adhar number p.Pan card ID q.Property document input as file type r.Govt ID proof input as file type s.I accept the terms and conditions check box t.Predict button u.Back button	pass

Prediction_ TC_025	Functional	Verify user is able to predict the results with Valid credentials	Name:aishu 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 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

Prediction_ TC_028	Functional	Verify user is able to predict the results with Invalid mobile number	Name:aishu Email ID:aishugmail.com Mobile number:90878909 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 enter only 10 digit mobile number'	pass
Prediction_ TC_029	Functional	Verify user is able to predict the results with Invalid Applicant income(entering alphabets)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:asfg Coapplicant Income:5000 Loan amount:7000	Application should display 'Enter only valid Applicant income alphabets are not allowed'	

			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_030	Functional	Verify user is able to predict the results with Invalid Coapplicant income(entering alphabets)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:10000 Coapplicant Income:sdfg 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 'Enter only valid Coapplicant income alphabets are not allowed'	

Prediction_ TC_031	Functional	Verify user is able to predict the results with Invalid Loan amount(entering alphabets)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:10000 Coapplicant Income:9000 Loan amount:abcg 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 'Enter only valid Loan amount alphabets are not allowed'	pass
Prediction_ TC_032	Functional	Verify user is able to predict the results with Invalid Loan amount term(entering alphabets)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:10000 Coapplicant Income:9000 Loan	Application should display 'Enter only valid Loan amount term alphabets are not allowed'	pass

			amount:8000 Loan amount term:asfj 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_033	Functional	Verify user is able to predict the results with Invalid Applicant income(entering large values)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:100000000000 000000000000000000 00000000 Coapplicant Income:9000 Loan amount:8000 Loan amount term:9000 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'Applicant income is too large enter a valid number'	pass

Prediction_ TC_034	Functional	Verify user is able to predict the results with Invalid Coapplicant income(entering large values)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:9000 Coapplicant Income:100000000000 000000000000000000 00000000 Loan amount:8000 Loan amount term:9000 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'Coapplicant income is too large enter a valid number'	pass
Prediction_ TC_035	Functional	Verify user is able to predict the results with Invalid Loan amount(entering large values)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:9000 Coapplicant	Application should display'Loan amount is too large enter a valid number'	pass

			Income:10000 Loan amount:100000000000 00000000000000000000 00000000 Loan amount term:9000 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_036	Functional	Verify user is able to predict the results with Invalid Loan amount term(entering large values)	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:Female Education:Graduate Self-Employed:No Married:No Dependents:3+ Property area:Urban Credit History:No Applicant Income:9000 Coapplicant Income:10000 Loan amount:1000 Loan amount term:1000000000000000 00000000000000000000 000000 Adhar number:2857 6789 6784 PAN card ID:9078508844 Property document:property.pdf Govt ID proof:Adhar.pdf	Application should display'Loan amount term is too large enter a valid number'	pass

			checked the check box'I accept the terms and conditions		
Loan approval status_TC_037	Functional	Verify user is able to see the Approve page when user clicked on predict button and when prediction value is 1	approve.html	approve page should display	pass
Loan approval status_TC_038	UI	Verify the UI elements in approve page	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:male Education:Not Graduate Self-Employed:No Married:yes Dependents:0 Property area:Urban Credit History:yes Applicant Income:2583 Coapplicant Income:2358 Loan amount:120 Loan amount term:360 Adhar number:2857 6789 6784 PAN card ID:9078508844	Application should show below UI elements: a.predicted result b.an approve image c.star rating d.description e.POST button	

			Property document:property.pdf Govt ID proof:Adhar.pdf checked the check box'I accept the terms and conditions		
Loan approval status_TC_039	Functional	Verify user is able to see the predicted result in approval page when user clicked on predict button and the predicted value is 1.	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:male Education:Not Graduate Self-Employed:No Married:yes Dependents:0 Property area:Urban Credit History:yes Applicant Income:2583 Coapplicant Income:2358 Loan amount:120 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'Congratulations! aishu You are eligible for loan'	pass

Loan approval status_TC_040	Functional	Verify user is able to see the reject page when user clicked on predict button and when prediction value is 0	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:male Education:Graduate Self-Employed:No Married:yes Dependents:1 Property area:Urban Credit History:yes Applicant Income:4583 Coapplicant Income:1508 Loan amount:128 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	reject.html	pass
Loan approval status_TC_041	UI	Verify the UI elements in reject page	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:male Education:Graduate Self-Employed:No Married:yes Dependents:1 Property area:Urban Credit History:yes Applicant Income:4583 Coapplicant Income:1508 Loan amount:128 Loan amount term:360	Application should show below UI elements: a.predicted result b.an approve image c.star rating d.description e.POST button	pass

			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		
Loan approval status_TC_042	Functional	Verify user is able to see the predicted result in the reject page when user clicked on predict button and when prediction value is 0	Name:aishu Email ID:aishugmail.com Mobile number:90878909 Gender:male Education:Graduate Self-Employed:No Married:yes Dependents:1 Property area:Urban Credit History:yes Applicant Income:4583 Coapplicant Income:1508 Loan amount:128 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'Sorry aishu You are not eligible for loan'	pass

Rating_TC_043	Functional	Verify whether user is able to see the message while providing rating after clicking on POST button in approve page	Rating: clicking on 5th star Description:I have got my prediction	Application should display 'Thanks for rating' as a pop up and the data should be stored in the google sheet	pass
Rating_TC_044	Functional	Verify whether user is able to see the message while providing rating after clicking on POST button in the reject page	Rating: clicking on 5th star Description:I have got my prediction	Application should display 'Thanks for rating' as a pop up and the data should be stored in the google sheet	pass
Rating_TC_045	Functional	Verify user is able to click on the stars to provide rating in the approve page	clicking on the star icons	The star should be turned to golden yellow colour on clicking	pass
Rating_TC_046	Functional	Verify user is able to click on the stars to provide rating in the reject page	clicking on the star icons	The star should be turned to golden yellow colour on clicking	pass

8.2 USER ACCEPTANCE TESTING

Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	7	2	1	1	11
Duplicate	1	0	0	0	1
External	2	3	0	1	6

Fixed	1	2	4	12	19
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	2	1	3
Totals	11	7	9	16	4 3

Test Case Analysis

Section	Total Cases	Not Tested	Fail	Pass
Login	8	0	0	8
Registration	4	0	0	4
Procedure	2	0	0	2
About	2	0	0	2
Contact	4	0	0	4

Prediction	14	0	0	14
Loan approval status	6	0	0	6
Rating	2	0	0	2

9.RESULTS

9.1 PERFORMANCE METRICS

S.N o.	Paramet er	Values	Screenshot
-----------	---------------	--------	------------

1.

Metrics

Classification Model:

Confusion Matrix -

Col_0	Loan status	0	1
0		52	0
1		16	117

Accuracy Score-

Random Forest Model Testing Accuracy

0.9135135135135135

Random Forest Model Training Accuracy

0.9137529137529138

Classification Report -

	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

pd.crosstab(ytest,ypredR)

col_0	0	1
Loan_Status		
0	52	0
1	16	117

print("Random Forest Model Testing Accuracy")
print(accuracy_score(ytest,ypredR))
print("Random Forest Model Training Accuracy")
print(accuracy_score(ytrain,ypred2R))

Random Forest Model Testing Accuracy
0.9135135135135135
Random Forest Model Training Accuracy
0.9137529137529138

print(classification_report(ypredR,ytest))

	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

Hyperparameter Tuning -

No tuning is performed as we have got 91% accuracy

Parameters used-

n_estimators=5000,max_depth=80,max_features='log2'

Validation Method -

In-sample validation

Random Forest Model

Model=RandomForestClassifier(n_estimators=5000,max_depth=80,max_features='log2')
Model.fit(x_train,y_train)

RandomForestClassifier(max_depth=80, max_features='log2', n_estimators=5000)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook. On GitHub, the HTML representation is unable to render, please try loading this page with [viewer.org](#).

ypredR=Model.predict(x_test)
ypred2R=Model.predict(x_train)

print(accuracy_score(ytest,ypredR),accuracy_score(ytrain,ypred2R))

print(classification_report(ytest,ypredR))

	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

10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- The customer can predict their eligibility from any part of the world and at any time so it provides user convenience
- Eligible applicant will be sanctioned loan without any delay
- Minimal documentation is required and there is no physical submission of documents
- Whole process will be automated,so human error will be avoided

- 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 had also provided the detailed procedure for applying loan and customer can also provide 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 Random Forest model is selected as the best model for predicting the loan approval status of the customer after evaluating its performance ,as it got 91% 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,payment option can be included in this application for exchanging money between the lender and borrower and bank can verify the customer document online using AI which makes the process of verification simpler and could be made more secure,trustworthy and dynamic weight conformation and in near future this module can be integrated with the module of automated processing system.

13.APPENDIX

SOURCE CODE

home.html

```
<html>

<head>

<meta charset="utf-8">

<title>Loan Prediction</title>
```

```
<link rel="stylesheet" href="static/home.css">
```

```
</head>
```

```
<body>
```

```
<div class="container">
```

```
<div class="navbar">
```

```
<nav>
```

```
<ul>
```

```
<li><a href="home.html">Home</a></li>
```

```
<li><a href="About.html">About</a></li>
```

```
<li><a href="procedure.html">Procedure</a></li>
```

```
<li><a href="contact.html">Contact Us</a></li>
```

```
<li><a href="login.html">User login</a></li>
```

```
<li><a href="bank login.html">Bank login</a></li>
```

```
</ul>
```

```
</nav>
```

```
</div>
```

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

```
<h1>Smart Lender - Applicant Credibility Prediction For Loan Approval </h1>
```

```
<p> Predit your loan eligibility here</p><br><br>
```

```
<a href="prediction.html" class="btn">PREDICT</a>
```

```
<br><br>
```

```
<h2>Team ID -PNT2022TMID39687</h2><br>
```

```
<h3>Team members</h3>
```

```
<p>AISHVARYA G</p>
```

```
<p>SHAMILI DEVI N B</p>
```

```
<p>PREMKUMAR A</p>
```

```
<p>PRIYADHARSHINI V</p>
```

```
<p>PRAVEEN KUMAR V</p>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

home.css

```
*{
```

```
margin: 0;
```

```
padding: 0;
```

```
}

.container{

    height: 100vh;

    width: 100%;

    background-image: url(images/background.png);

    background-position: center;

    background-size: cover;

    padding-left: 5%;

    padding-right: 5%;

    box-sizing: border-box;

    position: relative;

}

.navbar{

    width: 100%;

    height: 15vh;

    margin: auto;

    display: flex;

    align-items: center;

}

.logo{
```



```
width: 160px;

cursor: pointer;

}

a:hover{

color:#ff08a5;

font-size:20px;

}

nav{

flex: 1;

padding-left: 60px;

}

nav ul li{

display: inline-block;

list-style: none;

margin: 0px 20px;

}

nav ul li a{

text-decoration: none;

color: #578;

}
```

```
.content h1{

    font-size: 60px;

    font-weight: 100;

    margin-top: 24px;

    margin-bottom: 15px;

    color: #232d60;

}

.content p{

    font-size: 20px;

    color: #6a7199;

}

.content{

    margin-left: 10%;

    margin-top: 10%;

}

.content .btn{

    display: inline-block;

    background: linear-gradient(45deg, #87adfe, #ff77cd);
```

```
border-radius: 6px;

padding: 10px 20px;

box-sizing: border-box;

text-decoration: none;

box-shadow: 3px 8px 22px rgba(94,28,68,0.15);

color:rgb(6, 2, 255);

font-weight: bold;

}

.arrow-icons{

margin-top: 40px;

display: flex;

}

.arrow-icons img{

width: 40px;

margin-right: 25px;

}

.feature-img{

height: 90%;

position: absolute;
```

```
    bottom: 0;

    right: 160px;

}

.social-links{

    transform: rotate(-90deg);

    position: absolute;

    left: -80px;

    bottom: 180px;

}

.social-links a{

    text-decoration: none;

    color: #6a7199;

    padding-right: 20px;

    font-size: 14px;

}
```

about.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>About</title>

<link rel="stylesheet" href="static/About.css">

<link rel="stylesheet" href="Font-Awesome/all.min.css"/>

</head>

<body>

<a href="home.html"><button class="but">Back</button></a>

<section id="about-section">

<div class="about-left">



</div>

<div class="about-right">

<h1>About Us</h1>

<p>This project is used to automate the loan eligibility prediction for the customer as the cost of assets is increasing day by day and the capital required to purchase an entire asset is very high. So purchasing it out of your savings is not possible. The easiest way to get the required funds is to apply for a loan. But taking a loan is a very time consuming process. The Application has to go through a lot of stages and it's still not necessary that it will be approved. To decrease the approval time and to decrease the risk associated with the loan we have created this project</p>

```

<div class="address">

Phone No - 9876544321

Email ID - Bank@gmail.com

Place - Kanchipuram

Working days - Mon-Fri


```

        </div>

        <h3>Our speciality</h3>

        <ul>

        <br><li>1. Quick and convenient loans that can be availed online without having to go through a rigorous screening
process.</li>

        <br> <li>2. 24/7 Availiability</li>

        <br><li>3. Accurate prediction for loan eligibility</li>

        <br><li>4. Providing detailed procedure for loan approval</li>

        <br><li>5. Providing easy way to check the loan eligibility</li>

        </ul>

        </div>

    </section>

</li>

</ul>

</div>

</body>

</html>

```

about.css

```

/* -- About Section Start From here -- */

```

```

#about-section{

width: 90%;

height: auto;

display: flex;

justify-content: space-between;

align-items: center;

padding: 40px 5%;

}<!doctype html>

<html>

<head>

<meta charset="utf-8">

<title>Procedure</title>

<link rel="stylesheet" href="static/procedure.css">

</head>

<body>

<div class="title">

<a href="home.html"><button class="but">Back</button></a>



```


<header><h1>VIEW PROCEDURE</h1>
</header>

<h2>1.check the eligibility criteria</h2>

Maximum age to apply is 21 Years, not exceeding 65 years at the time of loan resume

You must be Indian resident and citizens

You must have necessary documents required for the personal loan along with duly filled and signed application form

<h2>2.check interest rates and other charges</h2>

The interest rates offered by Bank vary based on the borrow's credit worthiness,amount and tenure of the loan and other factors.

<h2>3.calculate your EMI</h2>

EMI calculators help to calculate the monthly EMI to be paid on the loan. This calculation depends on the total loan amount, interest rate and tenure

<h2>4.check required documents</h2>

Recent passport-sized photographs

Age proof

Address proof

Salary slips/proofs

processing fee cheque

<h2>5.wait for loan approval</h2>

Bank will process the personal loan application then a representative will be assigned to you who further take care of the procedure.


```
<br><h2 class="cons">Considerations before applying for loan</h2>
```

```
<br><li>Maintain your credit score</li>
```

```
<li>Plan the repayment beforehand</li>
```

```
<li>Be familiar with loan details</li>
```

```
<li>Organise your documents</li>
```

```
<li>Go through the terms and conditions of the loan</li>
```

```
<br><h2 class="cons">Steps to be followed</h2>
```

```
<br><li>Start Applying for loan</li>
```

```
<li>Estimate ypur EMIs</li>
```

```
<li>Fill the application form</li>
```

```
<li>Provide your correct details</li>
```

```
<li>Prepare your documents</li>
```

```
<li>Get funds in your bank account</li>
```

```
</div>
```

```
</body>
```

```
</html>
```

```
.about-right{
```

```
width: 57%;

}

.about-right ul li{

display: flex;

align-items: center;

}

.about-right h1{

color: #e74d06;

font-size: 37px;

margin-bottom: 5px;

}

.about-right p{

color: #444;

line-height: 26px;

font-size: 15px;

}

.about-right .address{

margin: 25px 0;

}

.about-right .address ul li{
```

```
margin-bottom: 5px;

}

.address .address-logo{

margin-right: 15px;

color: #e74d06;

}

.address .saprater{

margin: 0 20px;

}

.about-right .expertise ul{

width: 80%;

display: flex;

align-items: center;

justify-content: space-between;

}

.expertise h3{

margin-bottom: 10px;

}

.expertise .expertise-logo{

font-size: 19px;
```

```
margin-right: 10px;

color: #e74d06;

}

h3{

color: #e74d06;


}

.but{

background-color:#e74d06;

color: white;

padding: 5x;

margin:5px;

border-radius: 5px;

width:70px;

font-weight: bold;

}
```

procedure.html

```
<!doctype html>

<html>
```

```

<head>

<meta charset="utf-8">

<title>Procedure</title>

<link rel="stylesheet" href="static/procedure.css">

</head>

<body>

<div class="title">

<a href="home.html"><button class="but">Back</button></a>



<header><h1>VIEW PROCEDURE</h1><br></header>

<h2>1.check the eligibility criteria</h2>

<br><li>Maximum age to apply is 21 Years, not exceeding 65 years at the time of loan resume</li>

<li>You must be Indian resident and citizens</li>

<li>You must have necessary documents required for the personal loan along with duly filled and signed application form </li>

<br><h2>2.check interest rates and other charges</h2>

<br><li>The interest rates offered by Bank vary based on the borrow's credit worthiness,amount and tenure of the loan and other
factors.</li>

```


<h2>3.calculate your EMI</h2>

EMI calculators help to calculate the monthly EMI to be paid on the loan. This calculation depends on the total loan amount, interest rate and tenure

<h2>4.check required documents</h2>

Recent passport-sized photographs

Age proof

Address proof

Salary slips/proofs

processing fee cheque

<h2>5.wait for loan approval</h2>

Bank will process the personal loan application then a representative will be assigned to you who further take care of the procedure.

<h2 class="cons">Considerations before applying for loan</h2>

Maintain your credit score

Plan the repayment beforehand

Be familiar with loan details

Organise your documents

Go through the terms and conditions of the loan

<h2 class="cons">Steps to be followed</h2>

Start Applying for loan

Estimate ypur EMIs

```
<li>Fill the application form</li>

<li>Provide your correct details</li>

<li>Prepare your documents</li>

<li>Get funds in your bank account</li>


</div>

</body>

</html>
```

procedure.css

```
*{

margin: 0;

padding: 0;

box-sizing: border-box;

}

body{

background-color:white;

margin-left: 40px;
```



```
}

.title{

    text-align: left;

    margin-top: 100px;

    color: rgb(234, 30, 122);

}

.title h1{

    font-size: 50px;

    color: rgb(140, 3, 245);

    line-height: 5px;

    text-align: center;

    padding-bottom:50px;

    padding-left: 50px;

    margin-left:300px;

    margin-bottom: 10px;

}

.title h2{

    font-size: 21px;

    text-transform: uppercase;

    background-color:rgb(216, 243, 252);
```

```
padding: 10px;

}

.title li{

    font-size: 18px;

    color:rgb(23, 22, 22);

    padding-bottom: 20px;

}

.img{

    float: right;

    padding-top:500x;

    padding-bottom: 200px;

    width:350px;

}

.title .cons{

    background-color:rgb(248, 245, 192);

}
```

```
.but{

    background-color:#ff226f;

    color: white;

    padding: 5px;

    border-radius: 10px;

    font-weight:bold;

}
```

login.html

```
<!DOCTYPE html>

<html>

<head>

    <title>LogIn Page</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<style>

    .group{

        padding-top: 100px;

    }
```

```

</style>

<body>

<div class="container">

<div class="row">

<div style="width: 40%; margin: 25px auto;">

<div class="group">

<h3 style="text-align: center;">Login Page</h3>

<form action="login1.php" method="POST" >

<div class="form-group">

<label>UserName:</label><input type="text" name="username" class="form-control" autofocus placeholder="Enter your username or
gmail ID" required="">

</div>

<div class="form-group">

<label>Password:</label><input type="Password" name="password" class="form-control" autofocus placeholder="Password"
required="">

</div>

<label>Enter Captcha:</label>

<div class="form-row">

<div class="form-group col-md-6">

<input type="text" class="form-control" readonly id="capt" required="">

```

```

</div>

<div class="form-group col-md-6">

  <input type="text" class="form-control" id="textinput" required="">

</div>

</div>

<div class="form-group">

<button onclick="validcap()" name="save" class="btn btn-lg btn-success btn-block" >Submit</button>

</div>

</form>

<h6>Captcha not visible </h6>

<p>New Here?<a href="register.html">Register</a> </p>

</div>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function cap(){

var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V',

, 'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',

, 'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!', '@', '#', '$', '%', '^', '&', '*', '+'];

```

```
var a = alpha[Math.floor(Math.random()*71)];

var b = alpha[Math.floor(Math.random()*71)];

var c = alpha[Math.floor(Math.random()*71)];

var d = alpha[Math.floor(Math.random()*71)];

var e = alpha[Math.floor(Math.random()*71)];

var f = alpha[Math.floor(Math.random()*71)];

var final = a+b+c+d+e+f;

document.getElementById("capt").value=final;

}

function validcap(){

var stg1 = document.getElementById('capt').value;

var stg2 = document.getElementById('textinput').value;

if(stg1==stg2){

// alert("Form is validated Succesfully");

return true;

}else{

alert("Please enter a valid captcha");

return false;

}

}
```

```

</script>

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

```

register.html

```

<!DOCTYPE html>

<html lang="en">

<head>

<title>Register</title>

<link rel="stylesheet" type="text/css" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">

<link rel="stylesheet" type="text/css" href="static/register.css">

<script >

function check(x)

{

var number=/^([0-9]{10})+$/;

if(x.value.match(number)){

```

```

        alert("Valid email address!");

        document.myform.mon.focus();

        return true;

    }

    else{

        alert("Please enter only your 10 digit mobile number");

        document.myform.mon.focus();

        return false;

    }

}

function ValidateEmail(input) {

    var validRegex = /^[a-zA-Z0-9.!#$%&'*/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;

    if (input.value.match(validRegex)) {

        alert("Valid email address!");

        document.myform.email.focus();

        return true;

    }

    else {

        alert("Invalid email address!");
    }
}

```



```

document.myform.email.focus();

return false;

}

}

</script>

</head>

<body>

<div class="container">

    <form name="myform" method="post" class="form-signup" onsubmit="return check(document.myform.mon)" onsubmit="return
ValidateEmail(document.myform.email)">

        <h1 class="reg">Register</h1>

        <p>Create your account</p>

        <div class="form-group">

            <input type="text" class="form-control" name="name" placeholder="Enter your name" required >

        </div>

        <div class="form-group">

            <input type="email" class="form-control" name="email" placeholder="Enter your emailID" required>

        </div>

```

```

<div class="form-group">

    <input type="user name" class="form-control" name="username" placeholder="Enter your username" required>

</div>

<div class="form-group">

    <input type="password" class="form-control" name="password" placeholder="Enter your password" required>

</div>

<div class="form-group">

    <input type="text" class="form-control" name="mon" placeholder="Enter your mobile number" required >

</div>

<div class="form-group">

    <label>

        <input type="checkbox">

        I accept the <a href="terms.html">Terms and conditions</a>

    </label>

</div>

<input type="submit" class="btn btn-success btn-block" name="" value="submit">

</form>

</div>

</body>

```

```
</html>
```

bank login.html:

```
<!DOCTYPE html>

<html>

<head>

<title>LogIn Page</title>

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css" integrity="sha384-
TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkglIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

</head>

<style>

.group{

padding-top: 100px;

}

</style>

<body>

<div class="container">

<div class="row">
```

```

<div style="width: 40%; margin: 25px auto;">

    <div class="group">

        <h3 style="text-align: center;">Bank Login Page</h3>

        <form method="POST" action="bank1.php">

            <div class="form-group">

                <label>Bank user ID:</label><input type="text" name="BankUserName" class="form-control" autofocus placeholder="Enter the
Bank User ID" required>

            </div>

            <div class="form-group">

                <label>Bank Email ID:</label><input type="email" name="bankemail" class="form-control" autofocus placeholder="Enter the Bank
Email ID" required>

            </div>

            <div class="form-group">

                <label>Password:</label><input type="Password" name="Password" class="form-control" autofocus placeholder="Password"
required>

            </div>

            <label>Enter Captcha:</label>

            <div class="form-row">

                <div class="form-group col-md-6">

                    <input type="text" class="form-control" readonly id="capt" required>

                </div>

```

```

<div class="form-group col-md-6">

  <input type="text" class="form-control" id="textinput" required>

</div>

</div>

<div class="form-group">

  <button onclick="validcap()" name="Submit" class="btn btn-lg btn-success btn-block">Submit</button>

</div>

</form>

<h6>Captcha not visible </h6>

</div>

</div>

</div>

</div>

<script type="text/javascript">

function cap(){

var alpha = ['A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V'

,'W','X','Y','Z','1','2','3','4','5','6','7','8','9','0','a','b','c','d','e','f','g','h','i',

'j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z', '!','@','#','$','%','^','&','*','+'];

var a = alpha[Math.floor(Math.random()*71)];

```

```
var b = alpha[Math.floor(Math.random()*71)];

var c = alpha[Math.floor(Math.random()*71)];

var d = alpha[Math.floor(Math.random()*71)];

var e = alpha[Math.floor(Math.random()*71)];

var f = alpha[Math.floor(Math.random()*71)];

var final = a+b+c+d+e+f;

document.getElementById("capt").value=final;

}

function validcap(){

var stg1 = document.getElementById('capt').value;

var stg2 = document.getElementById('textinput').value;

if(stg1==stg2){

    alert("Form is validated Succesfully");

    return true;

}else{

    alert("Please enter a valid captcha");

    return false;

}

}

</script>
```

```

</body>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>

</html>

```

contact.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>contact</title>

<link rel="stylesheet" href="static/contact.css">

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

<script src='https://kit.fontawesome.com/a076d05399.js' crossorigin='anonymous'></script>

</head>

<body>

```

```

<a href="home.html"><button class="but">Back</button></a>

<div class="contact-section">

  <div class="contact-info">

    <div>Address : Vellore,Tamilnadu,India</div>

    <div>Email : Bank@gmail.com</div>

    <div>Mobile No : 9000033456</div>

    <div>Working Hours : Mon - Fri 10:00 AM to 4:30 PM</div>

  </div>

  <div class="box">

    <div class="title">

      <h1>Contact Us</h1><br>

      <h2>We are ready</h2>

    </div>

    <form action="" name="contact-us">

      <input type="text" name="name" class="form-control" id="name" placeholder="Your Name" required><br>

      <input type="text" name="phone" class="form-control" id="phone" placeholder="Your mobile Number" required><br>

      <input type="email" name="email" class="form-control" id="email" placeholder="Your Email id" required><br>

      <textarea name="message" class="form-control" id="message" rows="4" placeholder="Message"></textarea><br>

      <input type="submit" name="" class="form-control submit" value="SEND">

```



```
</form>

</div>

</div>

<script>

const scriptURL =
'https://script.google.com/macros/s/AKfycbwRbYoFq0mx9RGsOWGGg3mUgBc4Yap6Zo9EIRAHaB5_TvvPICT4UXUkAXihqNOER0_gfA/ex
ec'

const form = document.forms['contact-us']

form.addEventListener('submit', e => {

e.preventDefault()

fetch(scriptURL, { method: 'POST', body: new FormData(form)})

.then(response => console.log('Success!', response))

.catch(error => console.error('Error!', error.message))

form.reset()

alert('Success!')

})

</script>

</body>

</html>
```

contact.css:

```
*{  
  
    margin: 0;  
  
    padding: 0;  
  
    box-sizing: border-box;  
  
}  
  
body{  
  
    background-color:white;  
  
}  
  
.container{  
  
    width: auto;  
  
    margin: auto;  
  
}  
  
.icon{  
  
    width: 200px;  
  
    float: left;  
  
    height: 70px;  
  
}
```

```
ul{

    display: flex;

    list-style-type: none;

}

.contact-section{

    width: 100%;

    display: flex;

    justify-content: center;

    align-items: center;

}

.contact-info{

    justify-content: left;

    align-items: left;

    color: #ff226f;

    max-width: 500px;

    line-height: 65px;

    padding-left: 25px;

    font-size: 20px;

    font-weight: bold;
```

```
}

.contact-info i{

    margin-left: 20px;

    font-size: 25px;

}

.contact-info{

    flex: 1;

}

.title{

    text-align: center;

    margin-top: 100px;

    color: #fff;

}

.title h1{

    font-size: 32px;

    line-height: 10px;

}

.title h2{

    font-size: 16px;
```

```
}

form{

    margin-top: 50px;

    color: aqua;

}

.form-control{

    width: 600px;

    background: transparent;

    background-color: #fff;

    border: none;

    outline: none;

    border-bottom: 1px solid #ff226f;

    color:black;

    font-size: 18px;

    margin-bottom: 16px;

    padding: 10px;

    font-weight: 200px;

}

input{

    height: 45px;
```

```
}

form .submit{

    background-color: #ff226f;

    border-color: transparent;

    color: #fff;

    font-size: 20px;

    letter-spacing: 2px;

    height: 50px;

    margin-top: 20px;

    transition: 0.4s ease;

}

form .submit:hover{

    background-color:#ef3a79;

    cursor: pointer;

}

h1{

    color:#ff226f;

}

.but{

    background-color:#ff226f;
```

```
color: white;

padding: 10px;

margin:30px;

border-radius: 10px;

width:100px;

font-weight: bold;

}
```

prediction.html

```
<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

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

<!-- Bootstrap CSS -->

<link rel="stylesheet" href="static/prediction.css">

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
eOJMYsd53ii+scO/bJGFsICZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6" crossorigin="anonymous">

<link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet">
```

```
<title>prediction</title>

</head>

<body>

<script>

function valid(){

var Ai=document.getElementById("ApplicantIncome").value;

var Co=document.getElementById("CoapplicantIncome").value;

var LA=document.getElementById("LoanAmount").value;

var LT=document.getElementById("Loan_Amount_Term").value;


if(Ai > 10000000000000000000000000000000000000000000000000000000){

alert("Applicant income is too large enter a valid number")

return false;

}

if(Co > 10000000000000000000000000000000000000000000000000000000){

alert("Coapplicant income is too large enter a valid number")

return false;

}

if(LA > 10000000000000000000000000000000000000000000000000000000){

alert("Loan Amount is too large enter a valid number")
```



```
return false;

}

if(LT > 10000000000000000000000000000000000000000000000000000000){

    alert("loan amount term is too large enter a valid number")

    return false;

}

var name=document.getElementById("Name").value;

var letters=/^[a-zA-Z]*$/;

if(!name.match(letters)){

    alert("Name must contain only alphabets")

    return false;

}

var num=/^[0-9]+$;/

if(!Ai.match(num)){

    alert("Enter only valid numbers alphabets are not allowed ")

    return false;

}

if(!Co.match(num)){

    alert("Enter only valid numbers alphabets are not allowed ")

    return false;
```

```

    }

    if(!LA.match(num)){

        alert("Enter only valid numbers alphabets are not allowed ")

        return false;

    }

```

```

    if(!LT.match(num)){

        alert("Enter only valid numbers alphabets are not allowed ")

        return false;

    }

```

```

    var mo=document.getElementById("mon").value;

```

```

    var mn=/^[0-9]{10}$/;

```

```

    if(!mo.match(mn)){

        alert("Please enter only 10 digit mobile number")

        return false;

    }

```

```

}

```

```

</script>

```

```

<section class="text-gray-600 body-font">

```

```

<div class="container px-5 py-24 mx-auto">

```

```

<div class="flex flex-col text-center w-full mb-20">

```

```

<h1 class="Heading">LOAN ELIGIBILITY PREDICTION</h1><br>

<p class="fill">Fill the form for prediction</p>

</div>

<div>

</div>

<div class="mb-3">

<a class="btn btn-primary" href="." id="back" role="button">Back</a></div>

<form action="/prediction.html" method="post" onsubmit="return valid()">

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Name</label>

<input type="text" class="form-control" id="Name" name="Name" placeholder="Enter your Name" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Email ID</label>

<input type="email" class="form-control" id="email" name="email" placeholder="Enter your Email ID" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Mobile Number</label>

```

```

<input type="text" class="form-control" id="mon" name="mon" placeholder="Enter your Mobile number" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Gender</label>

<select class="form-select" id="gender" name="gender" aria-label="Default select example" required >

<option selected>-- select gender --</option>

<option value="Male">Male</option>

<option value="Female">Female</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label"> Married status</label>

<select class="form-select" id="married" name="married" aria-label="Default select example" required >

<option selected>-- select married status --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Dependents</label>

```

```

<select class="form-select" id="dependents" name="dependents" aria-label="Default select example" required>

<option selected>-- select dependents --</option>

<option value="0">0</option>

<option value="1">1</option>

<option value="2">2</option>

<option value="3+">3+</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Education</label>

<select class="form-select" id="education" name="education" aria-label="Default select example" required>

<option selected>-- select education --</option>

<option value="Graduate">Graduate</option>

<option value="Not Graduate">Not Graduate</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Self_Employed</label>

<select class="form-select" id="employed" name="employed" aria-label="Default select example" required>

<option selected>-- select Self_Employed --</option>

```

```

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Credit_History</label>

<select class="form-select" id="credit" name="credit" aria-label="Default select example" required >

<option selected >-- select Credit_History --</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Property_Area</label>

<select class="form-select" id="proparea" name="proparea" aria-label="Default select example" required>

<option selected>-- select Property_Area --</option>

<option value="Semiurban">Semiurban</option>

<option value="Urban">Urban</option>

<option value="Rural">Rural</option>

```

```

</select>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Enter ApplicantIncome</label>

  <input type="text" class="form-control" id="ApplicantIncome" name="ApplicantIncome" placeholder="ApplicantIncome" required>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Enter CoapplicantIncome</label>

  <input type="text" class="form-control" id="CoapplicantIncome" name="CoapplicantIncome" placeholder="CoapplicantIncome"
required>

</div>

<div class="mb-3">

  <label for="exampleFormControlInput1" class="form-label">Purpose of loan</label>

  <select class="form-select" id="pur" name="pur" aria-label="Default select example" required>

    <option selected>-- select the purpose of loan --</option>

    <option value="person">Personal loan</option>

    <option value="Bussiness">Bussiness loan</option>

    <option value="Education">Education loan</option>

    <option value="Home">Home loan</option>

```

```

<option value="Other">other</option>

</select>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter LoanAmount</label>

<input type="text" class="form-control" id="LoanAmount" name="LoanAmount" placeholder="LoanAmount" required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Loan_Amount_Term</label>

<input type="text" class="form-control" id="Loan_Amount_Term" name="Loan_Amount_Term" placeholder="Loan_Amount_Term"
required>

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter Adhar Number</label>

<input type="text" class="form-control" id="Adhar" name="Adhar" placeholder="Adhar Number" required >

</div>

<div class="mb-3">

<label for="exampleFormControlInput1" class="form-label">Enter PAN card ID</label>

<input type="text" class="form-control" id="PAN " name="PAN " placeholder="PAN card ID" required>

</div>

```



```

<div class="mb-3">

<label for="property document" class="form-label">Property Document</label><br><input type="file" required >

</div>

<div class="mb-3">

<label for="Govt ID proof" class="form-label">Govet ID proof</label><br><input type="file" required>

</div>

<div class="mb-3">

<input type="checkbox" required>

I accept the <a href="terms.html">Terms and conditions</a>

</div>

<br><br>

<div class="mb-3">

<input type="submit" class="but" value="PREDICT">

</div>

</form>

</div>

</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkLts3qm9Ekf" crossorigin="anonymous"></script>

```

```
</body>
```

```
</html>
```

prediction.css

```
body{
```

```
  margin:100px;
```

```
}
```

```
.form-label{
```

```
  color:rgb(235, 27, 97);
```

```
  font-weight: bolder;
```

```
  font-size: 18px;
```

```
}
```

```
.Heading{
```

```
  color:rgb(235, 27, 97);
```

```
  font-size: 30px;
```

```
  font-weight:bolder;
```

```
padding: 10px;

margin: 20px;

font-family: 'Times New Roman', Times, serif;

width: 100%;

}

.fill{

color: black;

font-size: 20px;

font-weight: 15px;

font-family: 'Times New Roman', Times, serif;

}

#but{

text-align: center;

align-items: center;

justify-content: center;

}

.mb-3{

width: 85%;

padding-left: 15%;
```

```
}

.form-control{

    margin:10px 2px;

    outline: solid 1px lightblue;

}

.form-select{

    margin:10px 2px;

    outline: solid 1px lightblue;

}

.but{

    background-color:rgb(235, 27, 97);

    color:white;

    border-radius: 10px;

    width:100px;

    border-color:white ;

    height:50px;

    padding: 10px;

}

#back{
```

```
background-color:rgb(235, 27, 97);

color:white;

border-radius: 10px;

width:100px;


}
```

approve.html

```
<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Loan approval status</title>

<link rel="stylesheet" href="static/approve.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

</head>

<body>

<h1>LOAN APPROVAL STATUS</h1>

<h2>{{ prediction_text }}</h2>


```

```

<h3>Please provide your feedback</h3>

<div class="container">

  <div class="post">

    <div class="text">Thanks for rating us!</div>

    <div class="edit">EDIT</div>

  </div>

  <div class="star-widget">

    <input type="radio" name="rate" id="rate-5">

    <label for="rate-5" class="fas fa-star"></label>

    <input type="radio" name="rate" id="rate-4">

    <label for="rate-4" class="fas fa-star"></label>

    <input type="radio" name="rate" id="rate-3">

    <label for="rate-3" class="fas fa-star"></label>

    <input type="radio" name="rate" id="rate-2">

    <label for="rate-2" class="fas fa-star"></label>

    <input type="radio" name="rate" id="rate-1">

    <label for="rate-1" class="fas fa-star"></label>

    <form action="" name="rating">

  </div>

</header></header>

```

```

<div class="textarea">

    <textarea cols="30" placeholder="Enter your name"></textarea>

</div>

<div class="textarea">

    <textarea cols="30" placeholder="Describe your experience.."></textarea>

</div>

<div class="btn">

    <button type="submit">Post</button>

</div>

</form>

</div>

</div>

<script>

    const scriptURL ='https://script.google.com/macros/s/AKfycbwZ_9addvPIcvu2Zpo2nxMo0sqzkbEb4mQroMYe-hx3zii-
tSa0c3J2x_Db_I6hTLqQxw/exec'
    const form = document.forms['rating']
    form.addEventListener('submit', e => {
        e.preventDefault()
        fetch(scriptURL, { method: 'POST', body: new FormData(form)})
        .then(response => console.log('Success!', response))
        .catch(error => console.error('Error!', error.message))
        form.reset()
        alert("Thanks for Rating!")

    })

</script>

```

```
</body>
```

```
</html>
```

approve.css

```
@import url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap');
```

```
{
```

```
margin: 0;
```

```
padding: 0;
```

```
box-sizing: border-box;
```

```
font-family: 'Poppins', sans-serif;
```

```
}
```

```
html,body{
```

```
display: grid;
```

```
height: 100%;
```

```
place-items: center;
```

```
text-align: center;
```

```
background:white;
```

```
}
```

```
.container{
```

```
position: relative;
```



```
width: 400px;

background:black;

padding: 20px 30px;

border: 1px solid #444;

border-radius: 5px;

display: flex;

align-items: center;

justify-content: center;

flex-direction: column;

}

.container .post{

display: none;

}

.container .text{

font-size: 25px;

color: #666;

font-weight: 500;

}

.container .edit{

position: absolute;
```

```
right: 10px;

top: 5px;

font-size: 16px;

color: #666;

font-weight: 500;

cursor: pointer;

}

.container .edit:hover{

text-decoration: underline;

}

.container .star-widget input{

display: none;

}

.star-widget label{

font-size: 40px;

color: #444;

padding: 10px;

float: right;

transition: all 0.2s ease;

}
```

```

input:not(:checked) ~ label:hover,

input:not(:checked) ~ label:hover ~ label{

    color: #fd4;

}

input:checked ~ label{

    color: #fd4;

}

input#rate-5:checked ~ label{

    color: #fe7;

    text-shadow: 0 0 20px #952;

}

#rate-1:checked ~ form header:before{

    content: "I just hate the service";

}

#rate-2:checked ~ form header:before{

    content: "I don't like the service ";

}

#rate-3:checked ~ form header:before{

    content: "Awesome service";

}

```

```
#rate-4:checked ~ form header:before{
```

```
content: "Satisfied service";
```

```
}
```

```
#rate-5:checked ~ form header:before{
```

```
content: "Excellent service ";
```

```
}
```

```
.container form{
```

```
display: none;
```

```
}
```

```
input:checked ~ form{
```

```
display: block;
```

```
}
```

```
form header{
```

```
width: 100%;
```

```
font-size: 25px;
```

```
color: #fe7;
```

```
font-weight: 500;
```

```
margin: 5px 0 20px 0;
```

```
text-align: center;
```

```
transition: all 0.2s ease;
```

```
}

form .textarea{

    height: 100px;

    width: 100%;

    overflow: hidden;

}

form .textarea textarea{

    height: 100%;

    width: 100%;

    outline: none;

    color: #eee;

    border: 1px solid #333;

    background: #222;

    padding: 10px;

    font-size: 17px;

    resize: none;

}

.textarea textarea:focus{

    border-color: #444;

}
```

```
form .btn{

    height: 45px;

    width: 100%;

    margin: 15px 0;

}

form .btn button{

    height: 100%;

    width: 100%;

    border: 1px solid #444;

    outline: none;

    background: #222;

    color: #999;

    font-size: 17px;

    font-weight: 500;

    text-transform: uppercase;

    cursor: pointer;

    transition: all 0.3s ease;

}

form .btn button:hover{

    background: #1b1b1b;
```

```
}

h1{

    background:lightpink;

    width:1500px;

    margin:none;

    color: rgb(99, 22, 23);

    padding-top: 20px;

    height:100px;

    margin-top:none;

    border: solid 5px rgb(99, 22, 23);

}

h2{

    color:rgb(42, 117, 7);

    font-size: 40px;

    font-family: 'Times New Roman', Times, serif;

}

h3{

    font-size:30px;

    color:rgb(246, 187, 11);
```

```
}
```

reject.html

```
<!DOCTYPE html>

<html lang="en" dir="ltr">

  <head>

    <meta charset="utf-8">

    <title>Loan approval status</title>

    <link rel="stylesheet" href="static/reject.css">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>

  </head>

  <body>

    <h1>LOAN APPROVAL STATUS</h1>

    <h2>{{ prediction_text }}</h2>

    <h3>Please provide your feedback</h3>

    <div class="container">

      <div class="post">

        <div class="text">Thanks for rating us!</div>
```



```

<div class="edit">EDIT</div>

</div>

<div class="star-widget">

  <input type="radio" name="rate" id="rate-5">

  <label for="rate-5" class="fas fa-star"></label>

  <input type="radio" name="rate" id="rate-4">

  <label for="rate-4" class="fas fa-star"></label>

  <input type="radio" name="rate" id="rate-3">

  <label for="rate-3" class="fas fa-star"></label>

  <input type="radio" name="rate" id="rate-2">

  <label for="rate-2" class="fas fa-star"></label>

  <input type="radio" name="rate" id="rate-1">

  <label for="rate-1" class="fas fa-star"></label>

  <form action="" name="rating">

    <header></header>

    <div class="textarea">

      <textarea cols="30" placeholder="Enter your name"></textarea>

    </div>

    <div class="textarea">

      <textarea cols="30" placeholder="Describe your experience.."></textarea>

```

```

</div>

<div class="btn">

  <button type="submit">Post</button>

</div>

</form>

</div>

</div>

<script>

  const scriptURL ='https://script.google.com/macros/s/AKfycbwZ_9addvPIcvu2Zpo2nxMo0sqzkbEb4mQroMYe-hx3zii-
tSa0c3J2x_Db_I6hTLqQxw/exec'
  const form = document.forms['rating']
  form.addEventListener('submit', e => {
    e.preventDefault()
    fetch(scriptURL, { method: 'POST', body: new FormData(form)})
    .then(response => console.log('Success!', response))
    .catch(error => console.error('Error!', error.message))
    form.reset()
    alert("Thanks for Rating!")
  })

</script>

</body>

</html>

```

reject.css

```

@import url('https://fonts.googleapis.com/css?family=Poppins:400,500,600,700&display=swap');

*{

  margin: 0;

```

```
padding: 0;

box-sizing: border-box;

font-family: 'Poppins', sans-serif;

}

html,body{

display: grid;

height: 100%;

place-items: center;

text-align: center;

background:white;

}

.container{

position: relative;

width: 400px;

background:black;

padding: 20px 30px;

border: 1px solid #444;

border-radius: 5px;

display: flex;

align-items: center;
```

```
justify-content: center;
```

```
flex-direction: column;
```

```
}
```

```
.container .post{
```

```
display: none;
```

```
}
```

```
.container .text{
```

```
font-size: 25px;
```

```
color: #666;
```

```
font-weight: 500;
```

```
}
```

```
.container .edit{
```

```
position: absolute;
```

```
right: 10px;
```

```
top: 5px;
```

```
font-size: 16px;
```

```
color: #666;
```

```
font-weight: 500;
```

```
cursor: pointer;
```

```
}
```

```
.container .edit:hover{

    text-decoration: underline;

}

.container .star-widget input{

    display: none;

}

.star-widget label{

    font-size: 40px;

    color: #444;

    padding: 10px;

    float: right;

    transition: all 0.2s ease;

}

input:not(:checked) ~ label:hover,

input:not(:checked) ~ label:hover ~ label{

    color: #fd4;

}

input:checked ~ label{

    color: #fd4;

}
```

```
input#rate-5:checked ~ label{

    color: #fe7;

    text-shadow: 0 0 20px #952;

}

#rate-1:checked ~ form header:before{

    content: "I just hate the service";

}

#rate-2:checked ~ form header:before{

    content: "I don't like the service ";

}

#rate-3:checked ~ form header:before{

    content: "Awesome service";

}

#rate-4:checked ~ form header:before{

    content: "Satisfied service";

}

#rate-5:checked ~ form header:before{

    content: "Excellent service ";

}

.container form{
```

```
display: none;

}

input:checked ~ form{

display: block;

}

form header{

width: 100%;

font-size: 25px;

color: #fe7;

font-weight: 500;

margin: 5px 0 20px 0;

text-align: center;

transition: all 0.2s ease;

}

form .textarea{

height: 100px;

width: 100%;

overflow: hidden;

}

form .textarea textarea{
```

```
height: 100%;

width: 100%;

outline: none;

color: #eee;

border: 1px solid #333;

background: #222;

padding: 10px;

font-size: 17px;

resize: none;

}

.textarea textarea:focus{

border-color: #444;

}

form .btn{

height: 45px;

width: 100%;

margin: 15px 0;

}

form .btn button{

height: 100%;
```



```
width: 100%;

border: 1px solid #444;

outline: none;

background: #222;

color: #999;

font-size: 17px;

font-weight: 500;

text-transform: uppercase;

cursor: pointer;

transition: all 0.3s ease;

}

form .btn button:hover{

background: #1b1b1b;

}

h1{

background:lightpink;

width:1500px;

margin:none;

color: rgb(99, 22, 23);

padding-top: 20px;
```

```
height:100px;

margin-top:none;

border: solid 5px rgb(99, 22, 23);

}

h2{

color:rgb(255, 11, 11);

font-size: 40px;

font-family: 'Times New Roman', Times, serif;

}

h3{

font-size:30px;

color:rgb(246, 187, 11);

}
```

terms.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>

</head>

<style>

.term{

    padding-top:150px;

    padding-left:450px;

    font-size:20px;

}

</style>

<body>

<div class="term">

<h2>Terms and conditions</h2>

<ol>

<li>We are only giving the prediction whether you are eligible for loan or not</li>

<li>You should trust the bank and should give all the required details</li>

<li>Bank will verify your details and inform you about the loan</li>

<li>Upload your correct Property document copy</li>

<li>Don't give any false details if it gets known we will not give you the loan</li>
```

```

<li>Enter your correct income details</li>

<li>Enter your name as per the name in adhar card</li>

<li>Enter your correct mobile number for contacting you</li>

<li>Enter your correct email address</li>

<li>Repay the loan amount and interest on time</li>

<li>Your interest rate will be increased if you don't repay the loan amount on time</li>

</ol>

</div>

</body>

</html>

```

app.py

```

from flask import render_template, Flask, request

import numpy as np

import pickle

from sklearn.preprocessing import scale

app = Flask(__name__, template_folder='templates')

model = pickle.load(open("Rfmodel.pkl", 'rb'))

```

```
@app.route('/')

def home():

    return render_template('home.html')

@app.route('/login.html')

def login():

    return render_template('login.html')

@app.route('/procedure.html')

def procedure():

    return render_template('procedure.html')

@app.route('/bank login.html')

def bank():

    return render_template('bank login.html')

@app.route('/About.html')

def about():

    return render_template('About.html')

@app.route('/terms.html')

def terms():

    return render_template('terms.html')

@app.route('/register.html')

def register():
```

```

        return render_template('register.html')

@app.route('/contact.html')

def contact():

    return render_template('contact.html')

@app.route('/home.html')

def home1():

    return render_template('home.html')

@app.route('/prediction.html')

def formpg():

    return render_template('prediction.html')

@app.route('/rating.html')

def rat():

    return render_template('rating.html')

@app.route('/prediction.html', methods = ['POST'])

def predict():

    if request.method=='POST':

        name=request.form['Name']

        gender=request.form['gender']

        married=request.form['married']

        dependents=request.form['dependents']

```

```
education=request.form['education']

employed=request.form['employed']

credit=request.form['credit']

proparea=request.form['proparea']

ApplicantIncome=float(request.form['ApplicantIncome'])

CoapplicantIncome=float(request.form['CoapplicantIncome'])

LoanAmount=float(request.form['LoanAmount'])

Loan_Amount_Term=float(request.form['Loan_Amount_Term'])

if gender == 'Male':

    gender = 1

else:

    gender = 0

if married == 'Yes':

    married = 1

else:

    married = 0

if education == 'Graduate':

    education = 0

else:

    education = 1
```

```
if employed == 'Yes':
```

```
    employed = 1
```

```
else:
```

```
    employed = 0
```

```
if dependents == '3+':
```

```
    dependents = 3
```

```
if credit == 'Yes':
```

```
    credit = 1
```

```
else:
```

```
    credit = 0
```

```
if proparea == 'Urban':
```

```
    proparea = 2
```

```
elif proparea == 'Rural':
```

```
    proparea = 0
```

```
else:
```

```
    proparea = 1
```

```
features=[gender,married,dependents,education,employed,ApplicantIncome,CoapplicantIncome,LoanAmount,Loan_Amount_Term,credit,proparea]
```



```

con_features = [np.array(features)]

prediction = model.predict(con_features)

print(prediction)

if prediction==1:

    return render_template('approve.html',prediction_text ='Congratulations! '+name+' You are eligible for loan')

else:

    return render_template('reject.html',prediction_text ='Sorry '+name+' You are not eligible for loan')

if __name__ == "__main__":

    app.run(debug=True)

```

Smartmodel.ipynb:

IMPORTING THE LIBRARIES:

```

import numpy as np

import pandas as pd

import pickle

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import sklearn

```

```
from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from xgboost import XGBClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import scale

from sklearn.preprocessing import MaxAbsScaler

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```

Reading The Dataset:

```
df=pd.read_csv('Loan_dataset.csv')
```

```
df
```

```
df.head()
```

```
df.info()
```

```
df.shape
```

```
df=df.drop(columns=["Loan_ID"],axis=1)
```

UNI-VARIATE ANALYSIS:

```
sns.countplot(df.Gender)
```

```
sns.countplot(df.Education)
```

```
sns.countplot(df.Self_Employed)
```

```
sns.distplot(df.ApplicantIncome)
```

```
sns.countplot(df.Property_Area)
```

```
sns.countplot(df.Married)
```

```
sns.histplot(df.LoanAmount)
```

```
sns.displot(df.CoapplicantIncome)
```

```
plt.pie(df.Property_Area.value_counts(),[0,0,0],labels=['Semi urban','Urban','Rural'])
```

BIVARIATE ANALYSIS:

```
sns.countplot(df['Married'],hue=df['Gender'])
```

```
sns.countplot(df['ApplicantIncome'],hue=df['CoapplicantIncome'])
```

```
sns.countplot(df['LoanAmount'],hue=df['Property_Area'])
```

```
sns.countplot(df['Education'],hue=df['Self_Employed'])
```

```
sns.barplot(df.ApplicantIncome,df.CoapplicantIncome)
```

```
sns.countplot(df['LoanAmount'],hue=df['Loan_Amount_Term'])
```

```
plt.scatter(df.ApplicantIncome,df.LoanAmount)
```

```
sns.countplot(df['Dependents'],hue=df['Gender'])
```

MULTI VARIATE ANALYSIS:

```
sns.heatmap(df.corr(),annot=True)
```

```
plt.plot(df.LoanAmount,df.Loan_Amount_Term,df.ApplicantIncome)
```

```
df.plot.line()
```

```
df.hist()
```

```
plt.plot(df.LoanAmount,df.ApplicantIncome,df.CoapplicantIncome)
```

```
plt.plot(df.Loan_Amount_Term,df.ApplicantIncome,df.CoapplicantIncome)
```

DESCRIPTIVE ANALYSIS:

```
df.describe()
```

```
df.std()
```

```
df.mean()
```

```
df.mode()
```

DATA PRE-PROCESING

CHECK FOR NULL VALUES:

```
df.isnull().any()
```

```
df.isnull().sum()
```

```
df['LoanAmount']=df['LoanAmount'].fillna(df['LoanAmount'].mean())
```

```
df['Loan_Amount_Term']=df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean())
```

```
df['Credit_History']=df['Credit_History'].fillna(df['Credit_History'].mean())
```

```
df['Gender']=df['Gender'].fillna(df['Gender'].mode()[0])
```

```
df['Married']=df['Married'].fillna(df['Married'].mode()[0])
```

```
df['Dependents']=df['Dependents'].fillna(df['Dependents'].mode()[0])
```

```
df['Self_Employed']=df['Self_Employed'].fillna(df['Self_Employed'].mode()[0])
```

```
df.isnull().any()
```

```
df.isnull().sum()
```

HANDLING CATEGORICAL VALUES:

```
df.head()
```

```
le=LabelEncoder()
```

```
df.Gender=le.fit_transform(df.Gender)
```

```
df.Married=le.fit_transform(df.Married)
```

```
df.Education=le.fit_transform(df.Education)
```

```
df.Self_Employed=le.fit_transform(df.Self_Employed)
```

```
df.Property_Area=le.fit_transform(df.Property_Area)
```

```
df.Loan_Status=le.fit_transform(df.Loan_Status)
```

```
df.Dependents=le.fit_transform(df.Dependents)
```

```
df.head()
```

SPLITTING INTO DEPENDENT AND DEPENDENT DATA:

```
df.head()
```

```
x=df.iloc[:, :-1]

y=df.Loan_Status
```

```
x.head()
```

```
y.head()
```

SCALING THE DATA:

```
scaler = MaxAbsScaler()

x_sc=scaler.fit_transform(x)

x_sc
```

BALANCING THE DATASET:

```
sns.countplot(df.Loan_Status)
```

```
rus=RandomUnderSampler(sampling_strategy=1)

x_res,y_res=rus.fit_resample(x,y)

ax=y_res.value_counts().plot.pie(autopct='%.2f')

_=ax.set_title("under-sampling")
```


SPLITTING DATA INTO TRAIN AND TEST;;

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=10)
```

```
xtrain.head()
```

```
xtest.head()
```

```
ytrain.head()
```

```
ytest.head()
```

```
xtrain.shape
```

```
xtest.shape
```

```
ytrain.shape
```

```
ytest.shape
```

MODEL BUILDING

DECISION TREE MODEL:

```
dmodel=DecisionTreeClassifier(random_state=100)
```

```
dmodel.fit(x_res,y_res)
```

```
ypredd=dmodel.predict(xtest)
```

```
ypred2d=dmodel.predict(xtrain)
```

RANDOM FOREST MODEL:

```
Rmodel=RandomForestClassifier(n_estimators=5000,max_depth=80,max_features='log2')
```

```
Rmodel.fit(x_res,y_res)
```

```
ypredR=Rmodel.predict(xtest)
```

```
ypred2R=Rmodel.predict(xtrain)
```

KNN MODEL:

```
kmodel=KNeighborsClassifier()
```

```
kmodel.fit(x_res,y_res)
```

```
ypredk=kmodel.predict(xtest)
```

```
ypred2k=kmodel.predict(xtrain)
```

XGBOOST MODEL:

```
xmodel=XGBClassifier(eval_metric='mlogloss',n_estimators=100,random_state=100)
```

```
xmodel.fit(x_res,y_res)
```

```
ypredx=xmodel.predict(xtest)
```

```
ypred2x=xmodel.predict(xtrain)
```

COMPARE THE MODEL:

```
print("Decision Tree Model Testing Accuracy")

print(accuracy_score(ytest,ypredd))

print("Decision Tree Model Training Accuracy")

print(accuracy_score(ytrain,ypred2d))
```

```
print("Random Forest Model Testing Accuracy")

print(accuracy_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy_score(ytrain,ypred2R))
```

```
print("KNN Model Testing Accuracy")

print(accuracy_score(ytest,ypredk))

print("KNN Model Training Accuracy")

print(accuracy_score(ytrain,ypred2k))
```

```
print("Xgboost Model Testing Accuracy")

print(accuracy_score(ytest,ypredx))

print("Xgboost Model Training Accuracy")

print(accuracy_score(ytrain,ypred2x))
```

EVALUATING PERFORMANCE OF THE MODEL AND SAVING THE MODEL:

```
## RANDOM FOREST MODEL IS SELECTED

print("Random Forest Model Testing Accuracy")

print(accuracy_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy_score(ytrain,ypred2R))
```

```
y=Rmodel.predict([[1,1,1,1,0,4583.0,1508.0,128.0,360.0,1.0,0]])

print(y)

y1=Rmodel.predict([[1,0,0,0,0,5849,0.0,146.412162,360.0,1.0,2]])

print(y1)

y2=Rmodel.predict([[1,0,0,1,0,678,987,90,24,1.0,2]])

print(y2)
```

```
f1_score(ypredR,ytest,average='weighted')
```

```
pd.crosstab(ytest,ypredR)
```

```
print(classification_report(ypredR,ytest))
```

```
##Saving the model by using pickle function

pickle.dump(Rmodel,open('Rfmodel.pkl','wb'))
```

IBM CLOUD DEPLOYMENT:

App_IBM.py:

```
from flask import render_template,Flask,request

import numpy as np

import pickle

import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.

API_KEY = "pYsrmaRyzz3LufvFxHnD2hbld7dSoiqu4iIsV0Rbk8Ry"

token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":API_KEY, "grant_type":
'urn:ibm:params:oauth:grant-type:apikey'})

mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app= Flask(__name__, template_folder='templates')

@app.route("/")

def home():

    return render_template('home.html')
```

```
@app.route('/login.html')

def login():

    return render_template('login.html')

@app.route('/procedure.html')

def procedure():

    return render_template('procedure.html')

@app.route('/bank login.html')

def bank():

    return render_template('bank login.html')

@app.route('/About.html')

def about():

    return render_template('About.html')

@app.route('/terms.html')

def terms():

    return render_template('terms.html')

@app.route('/register.html')

def register():

    return render_template('register.html')

@app.route('/contact.html')

def contact():
```

```
    return render_template('contact.html')

@app.route('/home.html')

def home1():

    return render_template('home.html')

@app.route('/prediction.html')

def formpg():

    return render_template('prediction.html')

@app.route('/rating.html')

def rat():

    return render_template('rating.html')

@app.route('/prediction.html',methods = ['POST'])

def predict():

    if request.method=='POST':

        name=request.form['Name']

        gender=request.form['gender']

        married=request.form['married']

        dependents=request.form['dependents']

        education=request.form['education']

        employed=request.form['employed']

        credit=request.form['credit']
```



```
proparea=request.form['proparea']

ApplicantIncome=float(request.form['ApplicantIncome'])

CoapplicantIncome=float(request.form['CoapplicantIncome'])

LoanAmount=float(request.form['LoanAmount'])

Loan_Amount_Term=float(request.form['Loan_Amount_Term'])


if gender == 'Male':

    gender = 1

else:

    gender = 0

if married == 'Yes':

    married = 1

else:

    married = 0

if education == 'Graduate':

    education = 0

else:

    education = 1

if employed == 'Yes':

    employed = 1
```

```

else:

    employed = 0

    if dependents == '3+':

        dependents = 3

    if credit == 'Yes':

        credit = 1

    else:

        credit = 0

    if proparea == 'Urban':

        proparea = 2

    elif proparea == 'Rural':

        proparea = 0

    else:

        proparea = 1

    features =
[[gender,married,dependents,education,employed,ApplicantIncome,CoapplicantIncome,LoanAmount,Loan_Amount_Term,credit,proparea]]

    #con_features = [np.array(features)]

    con_features = [np.array(features)]

```

```

payload_scoring = {"input_data": [{"fields":
['gender','married','depend','education','self_emp','applicant_income','co_income','loan_amount','loan_term','credit_history','property_area'],
"values":features}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/a4da3004-fab4-4770-b978-
b1fb81b0b561/predictions?version=2022-11-12', json=payload_scoring,headers={'Authorization': 'Bearer ' + mltoken})

print("response_scoring")

prediction = response_scoring.json()

predict = prediction['predictions'][0]['values'][0][0]

#prediction = model.predict(scale_features)

if predict == 1:

    return render_template('approve.html',prediction_text ='Congratulations! You are eligible for loan')

else:

    return render_template('reject.html',prediction_text ='Sorry You are not eligible for loan')

if __name__ == "__main__":

    app.run(debug=True)

```

Cloud_model.ipynb:

```

import os, types

import pandas as pd

```

```

from botocore.client import Config

import ibm_boto3

def __iter__(self): return 0

# @hidden_cell

# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.

# You might want to remove those credentials before you share the notebook.

cos_client = ibm_boto3.client(service_name='s3',

    ibm_api_key_id='SJDKNLjWvqlnVbkTgu9KbEg8wmpOmL-c3wvesKC-_vvp',

    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",

    config=Config(signature_version='oauth'),

    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'loanapproval-donotdelete-pr-n9qj6rgqlfsfne'

object_key = 'Loan_dataset.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body']

# add missing __iter__ method, so pandas accepts body as file-like object

if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )

df = pd.read_csv(body)

df.head()

```

IMPORTING THE LIBRARIES:

```
import numpy as np

import pandas as pd

import pickle

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

import sklearn

from sklearn.preprocessing import LabelEncoder

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

from xgboost import XGBClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import scale

from sklearn.preprocessing import MaxAbsScaler

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```

Reading The Dataset:

```
df=pd.read_csv('Loan_dataset.csv')
```

```
df
```

```
df.head()
```

```
df.info()
```

```
df.shape
```

```
df=df.drop(columns=["Loan_ID"],axis=1)
```

UNI-VARIATE ANALYSIS:

```
sns.countplot(df.Gender)
```

```
sns.countplot(df.Education)
```

```
sns.countplot(df.Self_Employed)
```

```
sns.distplot(df.ApplicantIncome)
```

```
sns.countplot(df.Property_Area)
```

```
sns.countplot(df.Married)
```

```
sns.histplot(df.LoanAmount)
```

```
sns.displot(df.CoapplicantIncome)
```

```
plt.pie(df.Property_Area.value_counts(),[0,0,0],labels=['Semi urban','Urban','Rural'])
```

BIVARIATE ANALYSIS:

```
sns.countplot(df['Married'],hue=df['Gender'])
```

```
sns.countplot(df['ApplicantIncome'],hue=df['CoapplicantIncome'])
```

```
sns.countplot(df['LoanAmount'],hue=df['Property_Area'])
```

```
sns.countplot(df['Education'],hue=df['Self_Employed'])
```

```
sns.barplot(df.ApplicantIncome,df.CoapplicantIncome)
```

```
sns.countplot(df['LoanAmount'],hue=df['Loan_Amount_Term'])
```

```
plt.scatter(df.ApplicantIncome,df.LoanAmount)
```

```
sns.countplot(df['Dependents'],hue=df['Gender'])
```

MULTI VARIATE ANALYSIS:

```
sns.heatmap(df.corr(),annot=True)
```

```
plt.plot(df.LoanAmount,df.Loan_Amount_Term,df.ApplicantIncome)
```

```
df.plot.line()
```

```
df.hist()
```

```
plt.plot(df.LoanAmount,df.ApplicantIncome,df.CoapplicantIncome)
```

```
plt.plot(df.Loan_Amount_Term,df.ApplicantIncome,df.CoapplicantIncome)
```


DESCRIPTIVE ANALYSIS:

```
df.describe()
```

```
df.std()
```

```
df.mean()
```

```
df.mode()
```

DATA PRE-PROCESING

CHECK FOR NULL VALUES:

```
df.isnull().any()
```

```
df.isnull().sum()
```

```
df['LoanAmount']=df['LoanAmount'].fillna(df['LoanAmount'].mean())
```

```
df['Loan_Amount_Term']=df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mean())
```

```
df['Credit_History']=df['Credit_History'].fillna(df['Credit_History'].mean())
```

```
df['Gender']=df['Gender'].fillna(df['Gender'].mode()[0])

df['Married']=df['Married'].fillna(df['Married'].mode()[0])

df['Dependents']=df['Dependents'].fillna(df['Dependents'].mode()[0])

df['Self_Employed']=df['Self_Employed'].fillna(df['Self_Employed'].mode()[0])
```

```
df.isnull().any()
```

```
df.isnull().sum()
```

HANDLING CATEGORICAL VALUES:

```
df.head()
```

```
le=LabelEncoder()
```

```
df.Gender=le.fit_transform(df.Gender)

df.Married=le.fit_transform(df.Married)

df.Education=le.fit_transform(df.Education)

df.Self_Employed=le.fit_transform(df.Self_Employed)

df.Property_Area=le.fit_transform(df.Property_Area)
```

```
df.Loan_Status=le.fit_transform(df.Loan_Status)
```

```
df.Dependents=le.fit_transform(df.Dependents)
```

```
df.head()
```

SPLITTING INTO DEPENDENT AND DEPENDENT DATA:

```
df.head()
```

```
x=df.iloc[:, :-1]
```

```
y=df.Loan_Status
```

```
x.head()
```

```
y.head()
```

SCALING THE DATA:

```
scaler = MaxAbsScaler()
```

```
x_sc=scaler.fit_transform(x)
```

```
x_sc
```

BALANCING THE DATASET:

```
sns.countplot(df.Loan_Status)
```

```
rus=RandomUnderSampler(sampling_strategy=1)

x_res,y_res=rus.fit_resample(x,y)

ax=y_res.value_counts().plot.pie(autopct='%%.2f')

_=ax.set_title("under-sampling")
```

SPLITTING DATA INTO TRAIN AND TEST;;

```
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=10)
```

```
xtrain.head()
```

```
xtest.head()
```

```
ytrain.head()
```

```
ytest.head()
```

```
xtrain.shape
```

```
xtest.shape
```

```
ytrain.shape
```

```
ytest.shape
```

MODEL BUILDING

DECISION TREE MODEL:

```
dmodel=DecisionTreeClassifier(random_state=100)
```

```
dmodel.fit(x_res,y_res)
```

```
ypredd=dmodel.predict(xtest)
```

```
ypred2d=dmodel.predict(xtrain)
```

RANDOM FOREST MODEL:

```
Rmodel=RandomForestClassifier(n_estimators=5000,max_depth=80,max_features='log2')
```

```
Rmodel.fit(x_res,y_res)
```

```
ypredR=Rmodel.predict(xtest)
```

```
ypred2R=Rmodel.predict(xtrain)
```

KNN MODEL:

```
kmodel=KNeighborsClassifier()
```

```
kmodel.fit(x_res,y_res)
```

```
ypredk=kmodel.predict(xtest)
```

```
ypred2k=kmodel.predict(xtrain)
```

XGBOOST MODEL:

```
xmodel=XGBClassifier(eval_metric='mlogloss',n_estimators=100,random_state=100)
```

```
xmodel.fit(x_res,y_res)
```

```
ypredx=xmodel.predict(xtest)
```

```
ypred2x=xmodel.predict(xtrain)
```

COMPARE THE MODEL:

```
print("Decision Tree Model Testing Accuracy")

print(accuracy_score(ytest,ypredd))

print("Decision Tree Model Training Accuracy")

print(accuracy_score(ytrain,ypred2d))
```

```
print("Random Forest Model Testing Accuracy")

print(accuracy_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy_score(ytrain,ypred2R))
```

```
print("KNN Model Testing Accuracy")

print(accuracy_score(ytest,ypredk))
```

```
print("KNN Model Training Accuracy")

print(accuracy_score(ytrain,ypred2k))
```

```
print("Xgboost Model Testing Accuracy")

print(accuracy_score(ytest,ypredx))

print("Xgboost Model Training Accuracy")

print(accuracy_score(ytrain,ypred2x))
```

EVALUATING PERFORMANCE OF THE MODEL AND SAVING THE MODEL:

```
## RANDOM FOREST MODEL IS SELECTED

print("Random Forest Model Testing Accuracy")

print(accuracy_score(ytest,ypredR))

print("Random Forest Model Training Accuracy")

print(accuracy_score(ytrain,ypred2R))
```

```
y=Rmodel.predict([[1,1,1,1,0,4583.0,1508.0,128.0,360.0,1.0,0]])

print(y)

y1=Rmodel.predict([[1,0,0,0,0,5849,0.0,146.412162,360.0,1.0,2]])

print(y1)

y2=Rmodel.predict([[1,0,0,1,0,678,987,90,24,1.0,2]])
```



```
print(y2)
```

```
f1_score(ypredR,ytest,average='weighted')
```

```
pd.crosstab(ytest,ypredR)
```

```
print(classification_report(ypredR,ytest))
```

IBM-DEPLOYMENT:

```
!pip install -U ibm-watson-machine-learning
```

```
from ibm_watson_machine_learning import APIClient
```

```
import json
```

```
wml_credentials = {  
  
    "apikey": "pYsrmaRyzz3LufvFxHnD2hbld7dSoiqu4iIsV0Rbk8Ry",  
  
    "url": "https://us-south.ml.cloud.ibm.com"  
  
}
```

```
wml_client = APIClient(wml_credentials)
```

```
wml_client.spaces.list()
```

```
SPACE_ID= "fc1e7624-773f-4e76-b9cb-d6a9a4e5a1a9"
```

```
wml_client.set.default_space(SPACE_ID)
```

```
wml_client.software_specifications.list(500)
```

```
import sklearn
```

```
sklearn.__version__
```

```
MODEL_NAME = 'Loan'
```

```
DEPLOYMENT_NAME = 'Approval'
```

```
DEMO_MODEL = Rmodel
```

```
# Set Python Version
```

```
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

```
# Setup model meta
```

```
model_props = {
```

```
wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,  
  
wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',  
  
wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid  
  
}
```

```
#Save model  
  
model_details = wml_client.repository.store_model(  
  
    model=DEMO_MODEL,  
  
    meta_props=model_props,  
  
    training_data=xtrain,  
  
    training_target=ytrain  
  
)
```

```
model_details
```

```
model_id = wml_client.repository.get_model_id(model_details)  
  
model_id
```

```
# Set meta  
  
deployment_props = {
```

```
wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,  
  
wml_client.deployments.ConfigurationMetaNames.ONLINE: {}  
  
}
```

```
# Deploy  
  
deployment = wml_client.deployments.create(  
  
    artifact_uid=model_id,  
  
    meta_props=deployment_props  
  
)
```

GitHub and Project Demo Link

GitHub Link: <https://github.com/IBM-EPBL/IBM-Project-53949-1661582224>

Demo Link:

https://drive.google.com/drive/folders/1kXITg0ILB_PTSVOSzW5CFNwBpGuIpIZk