THIRUVALLUVAR GOVERNMENT ARTS COLLEGE
RASIPURAM – 637 408

Team Leader: Sangeetha S (20UCS2322)

Team Member: Thenmozhi S (20UCS2325)

Team Member: Priya S (20UCS2321)

Team Member: Nithya T (20UCS2320)

1 INTRODUCTION

1.1 Overview

Personal loan approval is a process where financial institutions or banks evaluate a borrower's financial status and creditworthiness to decide whether or not to approve their loan application. Predicting personal loan approval using machine learning involves building a model that can analyze data related to borrowers' demographics, financial history, credit score, and other relevant factors to predict whether or not their loan application will be approved.

Machine learning algorithms, such as logistic regression, decision trees, random forests, and gradient boosting, can be used to analyze historical loan data and identify patterns and relationships that can help predict future loan approvals. By leveraging these algorithms, lenders can make faster and more accurate decisions, resulting in a more efficient and reliable lending process.

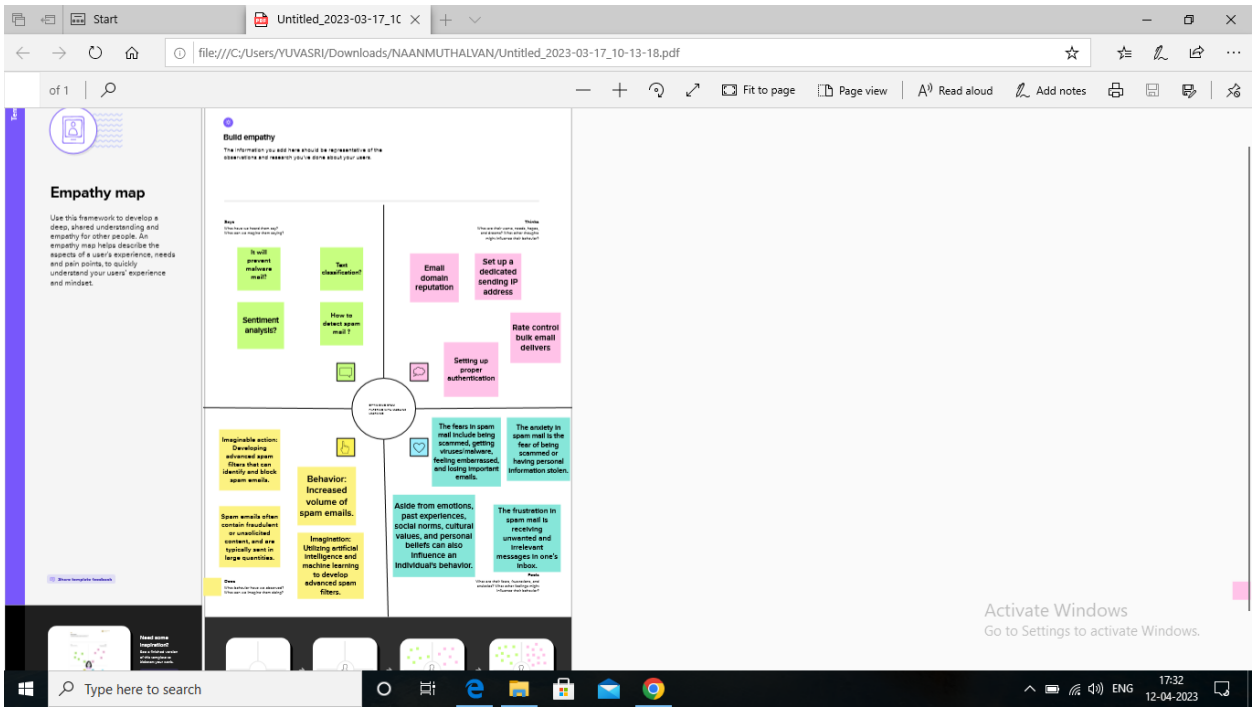
1.2 Purpose

The purpose of predicting personal loan approval using machine learning is to streamline and automate the loan approval process for lenders while improving the accuracy and efficiency of loan decisions.

Traditionally, lenders have used manual underwriting processes that rely on human judgment and are often time-consuming and error-prone

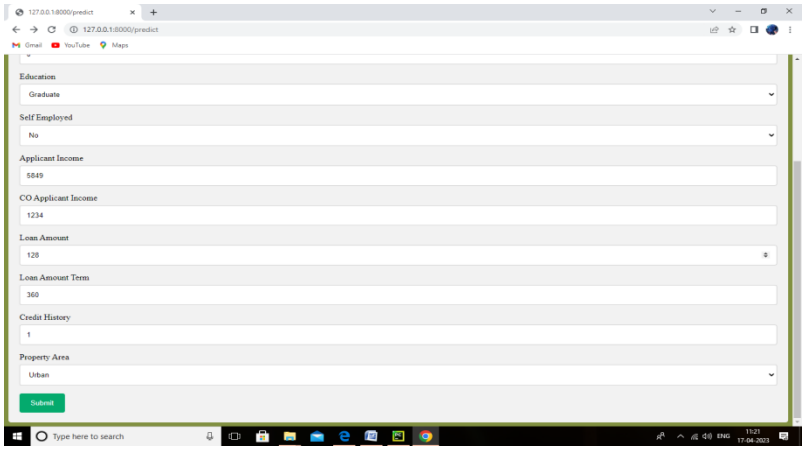
2 problem Definition & design Thinking

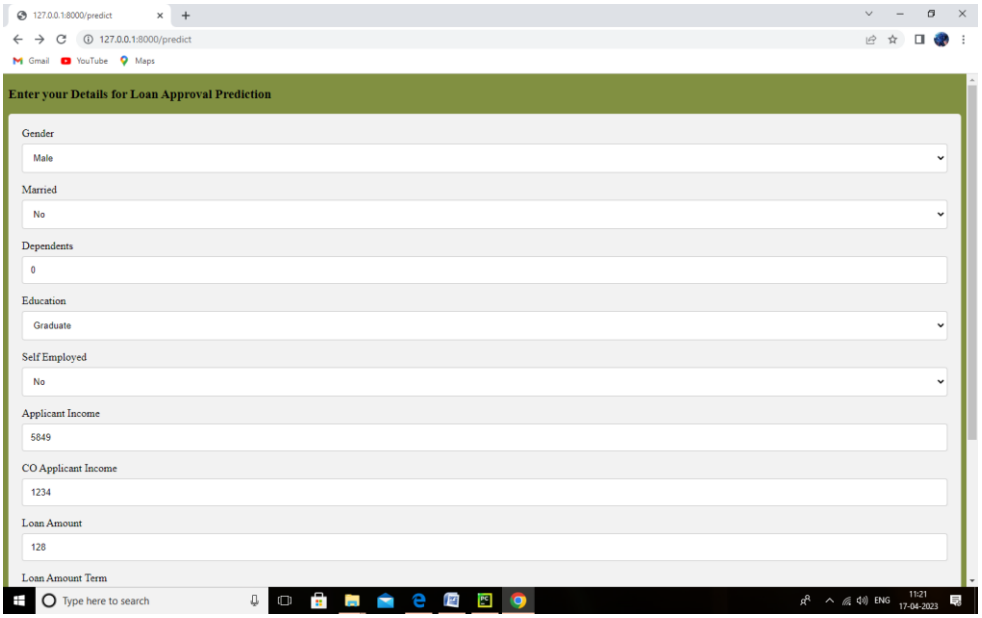
2.1 Empathy Map

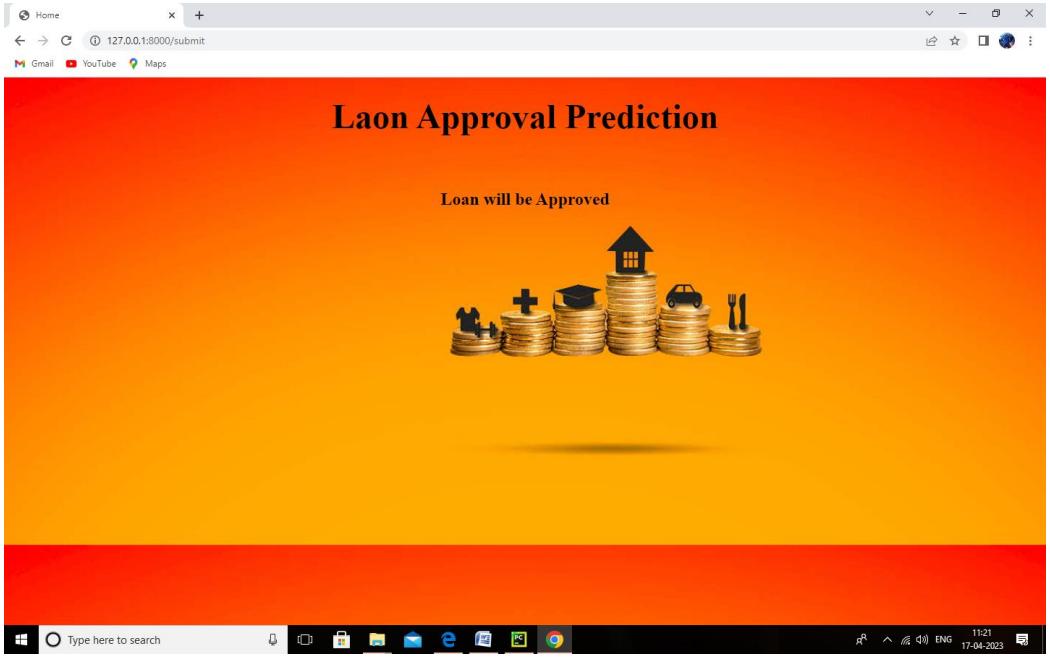


2.2 Ideation & Brainstorming Map

3.RESULT




127.0.0.1:8000/predict
Gmail
YouTube
Maps
Enter your Details for Loan Approval Prediction
Gender
Male
Married
No
Dependents
0
Education
Graduate
Self Employed
No
Applicant Income
5849
CO Applicant Income
1234
Loan Amount
128
Loan Amount Term
Type here to search
ENG
11:21
17-04-2023


Home
127.0.0.1:8000/submit
Gmail
YouTube
Maps
Laon Approval Prediction
Loan will be Approved
Type here to search
ENG
11:21
17-04-2023

3 ADVANTAGES &DISADVANTAGES

Advantages:-

1.Improved accuracy: Machine learning algorithms can analyze vast amounts of data and identify patterns and relationships that may not be apparent to human underwriters. This can lead to more accurate loan decisions and reduce the risk of default.

2.Speed and efficiency: Machine learning algorithms can quickly analyze borrower data and make loan decisions, allowing lenders to process loan applications faster and more efficiently.

Disadvantages:-

1. Biases and discrimination: Machine learning algorithms can perpetuate biases and discrimination if not designed properly. Lenders must ensure that their algorithms are fair and unbiased and do not discriminate against certain groups of borrowers.
2. .Lack of transparency: Machine learning algorithms can be complex and difficult to understand, making it challenging for borrowers to understand why their loan application was denied or approved**.**

4 APPLICATION

1. Data collection: Lenders need to collect data related to borrowers' demographics, financial history, credit score, and other relevant factors. This data can be obtained from a variety of sources, such as credit bureaus, financial institutions, and government agencies.
2. Data cleaning and preprocessing: Once data is collected, it needs to be cleaned and preprocessed to remove any errors or inconsistencies and make it suitable for analysis.

6 CONCLUSION

1. In conclusion, predicting personal loan approval using machine learning is a powerful tool that can improve the accuracy and efficiency of loan decisions for lenders. Machine learning algorithms can analyze vast amounts of borrower data and identify patterns and relationships that can help predict loan approvals more accurately and quickly than traditional manual underwriting processes.

2. However, there are also challenges associated with predicting personal loan approval using machine learning, such as biases and discrimination, lack of transparency, limited data, and the need for human oversight.

7 FUTURE SCOPE

Improved accuracy and efficiency: As machine learning algorithms continue to improve and evolve, they will become more accurate and efficient in predicting loan approvals, reducing the risk of default and improving the overall efficiency of the lending process.

Integration with blockchain technology: Blockchain technology can provide a secure and transparent way to store and share borrower data, which can enhance the accuracy and efficiency of machine learning algorithms.

8 APPENDIX

A Source code

```
import pandas as pd

import numpy as np

import pickle

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

import sklearn

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier
```

```
from sklearn.neighbors import KNeighborsClassifier

from sklearn.model_selection import RandomizedSearchCV

import imblearn

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

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

import pandas as pd

import numpy as np

data = pd.read_csv('/content/train_u6lujuX_CVtuZ9i.csv')

data.head()

data.info()

data.isnull().sum()
data['Gender']=data['Gender'].fillna(data['Gender'].mod
e()[0])
data['Dependents']=data['Dependents'].str.replace('+','
')
```

```
data['Dependents']=data['Dependents'].fillna(data['Depe
ndents'].mode()[0])
data['Self_Employed']=data['Self_Employed'].fillna(data
['Self_Employed'].mode()[0])
data['LoanAmount']=data['LoanAmount'].fillna(data['Loan
Amount'].mode()[0])
data['Loan_Amount_Term']=data['Loan_Amount_Term'].filln
a(data['Loan_Amount_Term'].mode()[0])
data['Credit_History']=data['Credit_History'].fillna(da
ta['Credit_History'].mode()[0])
```

```
data.isnull().sum()
```

```
data['Gender']=data['Gender'].fillna(data['Gender'].mod
e()[0])
data['Married']=data['Married'].fillna(data['Married'].
mode()[0])
data['Dependents']=data['Dependents'].str.replace('+','
')
data['Dependents']=data['Dependents'].fillna(data['Depe
ndents'].mode()[0])
data['Self_Employed']=data['Self_Employed'].fillna(data
['Self_Employed'].mode()[0])

data['LoanAmount']=data['LoanAmount'].fillna(data['Loan
Amount'].mode()[0])
data['Loan_Amount_Term']=data['Loan_Amount_Term'].filln
a(data['Loan_Amount_Term'].mode()[0])
data['Credit_History']=data['Credit_History'].fillna(da
ta['Credit_History'].mode()[0])
data.info()
data['Gender'].unique()
```

```
data['Gender']=data['Gender'].replace({'Male':1,'Female
':0})
data['Married'].unique()
data['Married']=data['Married'].replace({'Yes':1,'No':0
})

data['Loan_Status']=data['Loan_Status'].replace({'N':0,
'Y':1})

data['Dependents'].unique()
data['Dependents']=data['Dependents'].astype(int)

data['Education'].unique()
data['Education']=data['Education'].replace({'Graduate'
:1,'Not Graduate':2})
data['Self_Employed'].unique()
data['Self_Employed']=data['Self_Employed'].replace({'Y
es':1,'No':0})
data['Property_Area'].unique()
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data['Property_Area']=le.fit_transform(data['Property_A
rea'])
data['Property_Area']
data.info()
data.describe()
plt.figure(figsize=(12,5))

plt.subplot(121)

sns.distplot(data['ApplicantIncome'], color='r')

plt.subplot(122)
```

```
sns.distplot(data['Credit_History'])

plt.show()
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(x=data['Married'], hue=data['Gender'])
plt.subplot(132)
sns.countplot(x=data['Self_Employed'], hue=data['Educat
ion'])
plt.subplot(133)
sns.countplot(x=data['Property_Area'], hue=data['Loan_A
mount_Term'])
data.head()
import seaborn as sns

sns.swarmplot(x=data['Gender'],y=data['ApplicantIncome'
],hue=data['Loan_Status'])
x=data.drop(['Loan_Status','Loan_ID'],axis=1)
y=data['Loan_Status']
x_train, x_test, y_train, y_test = train_test_split(x,
y, test_size=0.33, random_state=42)
x_train
from sklearn.tree import DecisionTreeClassifier
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
y_pred=dtc.predict(x_test)
```

```
print('***DecisionTreeClassifier***')
print('confusion matrix')
print(confusion_matrix(y_test,y_pred))
print('classification report')
print(classification_report(y_test,y_pred))
import pickle
pickle.dump(dtc,open('rdf.pkl','wb'))
```

app.py

```
import numpy as np

import pickle

import pandas

import os

from flask import Flask, request, render_template


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

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

scale = pickle.load(open('scale1.pkl', 'rb'))
```

```
@app.route('/')  # rendering the html template

def home():

    return render_template('home.html')  # to access
the home page.



@app.route('/predict', methods=["POST", "GET"])  #
rendering the html template

def predict():

    return render_template("input.html")



@app.route('/submit', methods=["POST", "GET"])  # route
to show the predictions in a web UI

def submit():

    #  reading the inputs given by the user

    input_feature = [int(x) for x in
request.form.values()]

    # input_feature = np.transpose(input_feature)

    input_feature = [np.array(input_feature)]

    print(input_feature)
```

```
    names = ['Gender', 'Married', 'Dependents',
'Education', 'Self_Employed', 'ApplicantIncome',

             'CoapplicantIncome', 'LoanAmount',
'Loan_Amount_Term', 'Credit_History', 'Property_Area']

    data = pandas.DataFrame(input_feature,
columns=names)

    print(data)



    # data_scaled = scale.fit_transform(data)

    # data = pandas.DataFrame(,columns=names)



    # predictions using the loaded model file

    prediction = model.predict(data)

    print(prediction)

    prediction = int(prediction)

    print(type(prediction))



    if (prediction == 0):

        return render_template("output.html",
result="Loan will not be Approved")

    else:
```

```
        return render_template("output.html",
result="Loan will  be Approved")

    # showing the prediction results in a UI




if __name__ == "__main__":

    app.run(host='0.0.0.0', port=8000, debug=True)  #
running the app

    port = int(os.environ.get('PORT', 5000))

    app.run(debug=False)
```

home.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
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+rP48ckxl
pbzKgwra6" crossorigin="anonymous">

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



    <title>loan Prediction</title>

  </head>

  <body>

<!-- This example requires Tailwind CSS v2.0+ -->

<div class="relative bg-white overflow-hidden">

  <div class="max-w-7xl mx-auto">

    <div class="relative z-10 pb-8 bg-white sm:pb-16
md:pb-20 lg:max-w-2xl lg:w-full lg:pb-28 xl:pb-32">

      <svg class="hidden lg:block absolute right-0
inset-y-0 h-full w-48 text-white transform translate-x-
1/2" fill="currentColor" viewBox="0 0 100 100"
preserveAspectRatio="none" aria-hidden="true">

        <polygon points="50,0 100,0 50,100 0,100" />
```

```
      </svg>


      <div class="relative pt-6 px-4 sm:px-6 lg:px-8">

        <nav class="relative flex items-center justify-
between sm:h-10 lg:justify-start" aria-label="Global">

          <div class="flex items-center flex-grow flex-
shrink-0 lg:flex-grow-0">

            <div class="flex items-center justify-
between w-full md:w-auto">

              <a href="#">

                <span class="sr-only">Workflow</span>

                <img class="h-8 w-auto sm:h-10" src="">

              </a>

              <div class="-mr-2 flex items-center
md:hidden">

                <button type="button" class="bg-white
rounded-md p-2 inline-flex items-center justify-center
text-gray-400 hover:text-gray-500 hover:bg-gray-100
focus:outline-none focus:ring-2 focus:ring-inset
focus:ring-indigo-500" aria-expanded="false">

                  <span class="sr-only">Open main
menu</span>

                  <!-- Heroicon name: outline/menu -->
```

```
                    <svg class="h-6 w-6"
xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke="currentColor" aria-
hidden="true">

                      <path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M4 6h16M4
12h16M4 18h16" />

                    </svg>

                  </button>

                </div>

              </div>

            </div>

            <div class="hidden md:block md:ml-10 md:pr-4
md:space-x-8">

              <a href="#" class="font-medium text-gray-
500 hover:text-gray-900">Home</a>



              <a href="#" class="font-medium text-gray-
500 hover:text-gray-900">Prediction</a>



              <a href="#" class="font-medium text-gray-
500 hover:text-gray-900">About us</a>
```

```
            <a href="#" class="font-medium text-gray-
500 hover:text-gray-900">contact</a>


          </div>

        </nav>

      </div>


      <!--

        Mobile menu, show/hide based on menu open
state.


        Entering: "duration-150 ease-out"

          From: "opacity-0 scale-95"

          To: "opacity-100 scale-100"

        Leaving: "duration-100 ease-in"

          From: "opacity-100 scale-100"

          To: "opacity-0 scale-95"

      -->

        <div class="absolute top-0 inset-x-0 p-2
transition transform origin-top-right md:hidden">
```

```
          <div class="rounded-lg shadow-md bg-white
ring-1 ring-black ring-opacity-5 overflow-hidden">

            <div class="px-5 pt-4 flex items-center
justify-between">

              <div>

                <img class="h-8 w-auto"
src="https://tailwindui.com/img/logos/workflow-mark-
indigo-600.svg" alt="">

              </div>

              <div class="-mr-2">

                <button type="button" class="bg-white
rounded-md p-2 inline-flex items-center justify-center
text-gray-400 hover:text-gray-500 hover:bg-gray-100
focus:outline-none focus:ring-2 focus:ring-inset
focus:ring-indigo-500">

                  <span class="sr-only">Close main
menu</span>

                  <!-- Heroicon name: outline/x -->

                  <svg class="h-6 w-6"
xmlns="http://www.w3.org/2000/svg" fill="none"
viewBox="0 0 24 24" stroke="currentColor" aria-
hidden="true">

                    <path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M6 18L18
6M6 6l12 12" />
```

```
                    </svg>

                  </button>

                </div>

              </div>

            <div class="px-2 pt-2 pb-3 space-y-1">

              <a href="#" class="block px-3 py-2 rounded-
md text-base font-medium text-gray-700 hover:text-gray-
900 hover:bg-gray-50">Home</a>



              <a href="#" class="block px-3 py-2 rounded-
md text-base font-medium text-gray-700 hover:text-gray-
900 hover:bg-gray-50">prediction</a>



              <a href="#" class="block px-3 py-2 rounded-
md text-base font-medium text-gray-700 hover:text-gray-
900 hover:bg-gray-50">about us</a>



              <a href="#" class="block px-3 py-2 rounded-
md text-base font-medium text-gray-700 hover:text-gray-
900 hover:bg-gray-50">contact</a>

            </div>



          </div>
```

```
    </div>


    <main class="mt-10 mx-auto max-w-7xl px-4 sm:mt-
12 sm:px-6 md:mt-16 lg:mt-20 lg:px-8 xl:mt-28">

      <div class="sm:text-center lg:text-left">

        <h1 class="text-4xl tracking-tight font-
extrabold text-gray-900 sm:text-5xl md:text-6xl">

          <span class="block xl:inline">Loan
Prediction</span>

          <span class="block text-indigo-600
xl:inline">Machine Learning </span>

        </h1>

        <p class="mt-3 text-base text-gray-500 sm:mt-
5 sm:text-lg sm:max-w-xl sm:mx-auto md:mt-5 md:text-xl
lg:mx-0">

          Lorem ipsum dolor sit amet consectetur
adipisicing elit. Excepturi ad perspiciatis dolores,
deleniti culpa odit dolorem harum dolore ex amet.

        </p>

        <div class="mt-5 sm:mt-8 sm:flex sm:justify-
center lg:justify-start">

          <div class="rounded-md shadow">

            <a href="/predict" class="w-full flex
items-center justify-center px-8 py-3 border border-
```

```
transparent text-base font-medium rounded-md text-white
bg-indigo-600 hover:bg-indigo-700 md:py-4 md:text-lg
md:px-10">

                Prediction

              </a>

            </div>



          </div>

        </div>

      </main>

    </div>

  </div>

  <div class="lg:absolute lg:inset-y-0 lg:right-0 lg:w-
1/2">

    <img class="h-56 w-full object-cover sm:h-72 md:h-
96 lg:w-full lg:h-full"
src="https://images.unsplash.com/photo-1551434678-
e076c223a692?ixlib=rb-
1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=
2850&q=80" alt="">

  </div>

</div>
```

```
    <!-- Option 1: Bootstrap Bundle with Popper -->

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
JEW9xMcG8R+pH31jmWH6WWP0WintQrMb4s7ZOdauHnUtxwoG2vI5DkL
tS3qm9Ekf" crossorigin="anonymous"></script>

  </body>

</html>
```

Input.html

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/html">

<head>


     <style>

     /* Style inputs with type="text", select elements
and textareas */
```

```
    input[type=number], select, textarea {
      width: 100%; /* Full width */
      padding: 12px; /* Some padding */
      border: 1px solid #ccc; /* Gray border */
      border-radius: 4px; /* Rounded borders */
      box-sizing: border-box; /* Make sure that padding and width stays in place */
      margin-top: 6px; /* Add a top margin */
      margin-bottom: 16px; /* Bottom margin */
      resize: vertical /* Allow the user to vertically resize the textarea (not horizontally) */
    }

    /* Style the submit button with a specific background color etc */
    input[type=submit] {
      background-color: #04AA6D;
      color: white;
      padding: 12px 20px;
      border: none;
      border-radius: 4px;
```

```
      cursor: pointer;
    }

    /* When moving the mouse over the submit button,
add a darker green color */
    input[type=submit]:hover {
      background-color: #45a049;
    }

    /* Add a background color and some padding around
the form */
    .container {
      border-radius: 5px;
      background-color: #f2f2f2;
      padding: 20px;
    }

   body {
      background-size: cover;
```

```
      background-image: url('https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcR70TDTAUcRk7Q7W2nK-
aIqsmoTN6VruMi0mA&usqp=CAU');

    }

    </style>

</head>

<body>


<h3>Enter your Details for Loan Approval
Prediction</h3>


<div class="container">


    <form action = '/submit', method = 'post'>


    <label for="Gender">Gender</label>

    <select id="Gender" name="Gender">

      <option value=0>Male</option>

      <option value=1>Female</option>

    </select>
```

```
    <label for="Married">Married</label>

    <select id="Married" name="Married">

      <option value=1>Yes</option>

      <option value=0>No</option>

    </select>




    <label for="Dependents">Dependents</label>

    <input type="number" id="Dependents" min = 0 max =
10 name="Dependents" placeholder="No of Dependents on
you......">




    <label for="Education">Education</label>

    <select id="Education" name="Education">

      <option value=1>Graduate</option>

      <option value=0>Not Graduate </option>

    </select>
```

```
     <label for="Self_Employed">Self Employed</label>
    <select id="Self_Employed" name="Self_Employed">
      <option value=1>Yes</option>
      <option value=0>No</option>
    </select>

    <label for="ApplicantIncome">Applicant
Income</label>
    <input type="Number" min = 1000
id="ApplicantIncome" name="Applicant Income"
placeholder="Your Income...">

    <label for="CoapplicantIncome">CO Applicant
Income</label>
    <input type="Number" min = 100
id="CoapplicantIncome" name="Co Applicant Income"
placeholder="Your Co Applicant Income...">

    <label for="LoanAmount">Loan Amount</label>
    <input type="Number" min = 0 id="LoanAmount"
name="Loan Amount" placeholder="Enter the Loan Amount
...">
```

```
    <label for="Loan_Amount_Term">Loan Amount
Term</label>

    <input type="Number" min = 30 max = 15000
id="Loan_Amount_Term" name="Loan Amount Term"
placeholder="Enter the Term Loan Amount in days ...">



    <label for="Credit_History">Credit History</label>

    <input type="Number" min = 0 max = 5
id="Credit_History" name="Credit History"
placeholder="Enter the Your Previous Credit History
...">



     <label for="Property_Area">Property Area</label>

    <select id="Property_Area" name="Property Area">

      <option value=2>Urban</option>

      <option value=0>Rural</option>

      <option value=1>Semi Urban</option>

    </select>



     <input type ='submit' value = 'Submit'>
```

```
</div>


</body>

</html>
```

Output.html

```
<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<style>

body

{

    background-image: url("https://www.dbs.com/in/iwov-resources/media/images/learn/banners/what-is-a-personal-loan-1404x630.jpg");
```

```
    background-size: cover;
}
.pd{
padding-bottom:45%;}
}
</style>
</head>
<body>
<form action="/submit" method="post">
<br>
<center><b class="pd"><font color="black" size="15"
font-family="Comic Sans MS" >Laon Approval
Prediction</font></b></center><br><br>
<div>
<br>
<center>
<h2><font color="black"> {{result}} </h2>
</center>
</div>
</form>
```

```
</body>

</html>
```