**CHAPTER - 1**

**INTRODUCTION**

**1.1 Project Overview :**

Loan sanctioning and credit scoring forms a multi-billion dollar industry -- in the US alone. With everyone from young students, entrepreneurs, and multi-million dollar companies turning to banks to seek financial support for their ventures, processing these applications creates a complex and cumbersome task for any banking institution. As of 2022, more than 20 million people in the US have active loans owing a collective debt of 178 billion dollars. Despite that, more than 20% of all applicants were denied loans. The loan approval or rejection has enormous ramifications for both the applicant and the bank, causing possible opportunity costs for both parties. Banks like Wells Fargo and Morgan Stanley have looked at the [use of AI](https://www.projectpro.io/article/artificial-intelligence-project-ideas/461) in determining lending risk and developing a loan prediction system in recent years to overcome human bias and delays in the application processing time.

**1.2 Purpose :**

To deal with the problem, we developed automatic loan prediction using machine learning techniques. We will train the machine with previous dataset. so machine can analyse and understand the process . Then machine will check for eligible applicant and give us result.

**CHAPTER - 2**

**LITERATURE SURVEY**

**2.1 Existing problem :**

Bank employees check the details of applicant manually and give the loan to eligible applicant. Checking the details of all applicants takes lot of time. The artificial neural network model for predict the credit risk of a bank. The Feed- forward back propagation neural network is used to forecast the credit default. The method in which two or more classifiers are combined together to produce a ensemble model for the better prediction. They used the bagging and boosting techniques and then used random forest technique. The process of classifiers is to improve the performance of the data and it gives better efficiency. In this work, the authors describe various ensemble techniques for binary classification and also for multi class classification. The new technique that is described by the authors for ensemble is COB which gives effective performance of classification but it also compromised with noise and outlier data of classification. Finally they concluded that the ensemble based algorithm improves the results for training data set.

**2.2 References :**

* Ashwini S. Kadam, Shraddha R Nikam, Ankita A. Aher, Gayatri V. Shelke, Amar S. Chandgude, 2021, “Prediction for Loan Approval using Machine Learning Algorithm”, No “Apr” / ”2021”.
* Sivasree M S, Rekha Sunny T, (2015), “Loan Credibility Prediction System Based on Decision Tree Algorithm”, No “September” / “2015”.
* Anuja Kadam, Pragati Namde, Sonal Shirke, Siddhesh Nandgaonkar, Dr.D.R Ingle, 2021, “Loan Credibility Prediction System using Data Mining Techniques” No “May” / “2021”.
* Pidikiti Supriya , Myneedi Pavani , Nagarapu Saisushma , Namburi Vimala Kumari , K Vikas, 2019, “Loan Prediction by using Machine Learning Models”, No “April” / “2019”.
* https://medium.com/swlh/lending-club-data-web-app-ada56ff64cee
* https://github.com/smartinternz02/SI-GuidedProject-48927-1652694502
* 7https://www.academia.edu/77162007/BANK\_LOAN\_PREDICTION\_USING MACHINE\_LEARNING

**2.3 Problem statement definition :**

• Tracking or checking the status is difficult.

• Prone to human errors.

• Time consumption is high.

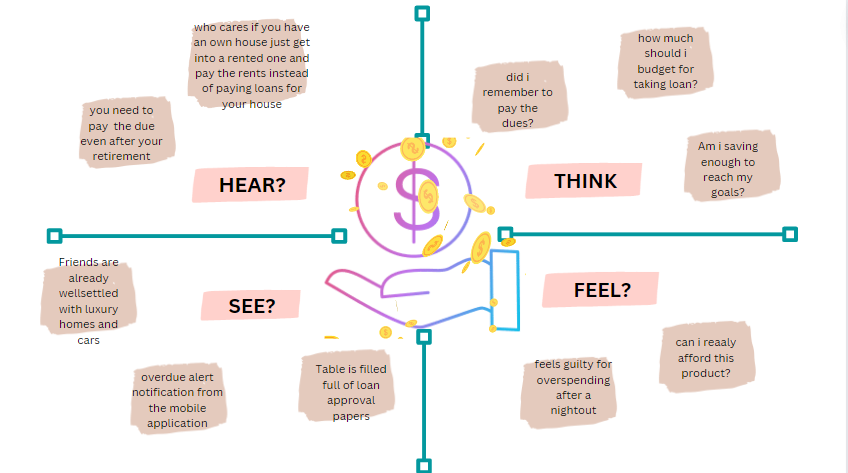
• Lot of paper works.

• Poor customer service due to lack of manpower

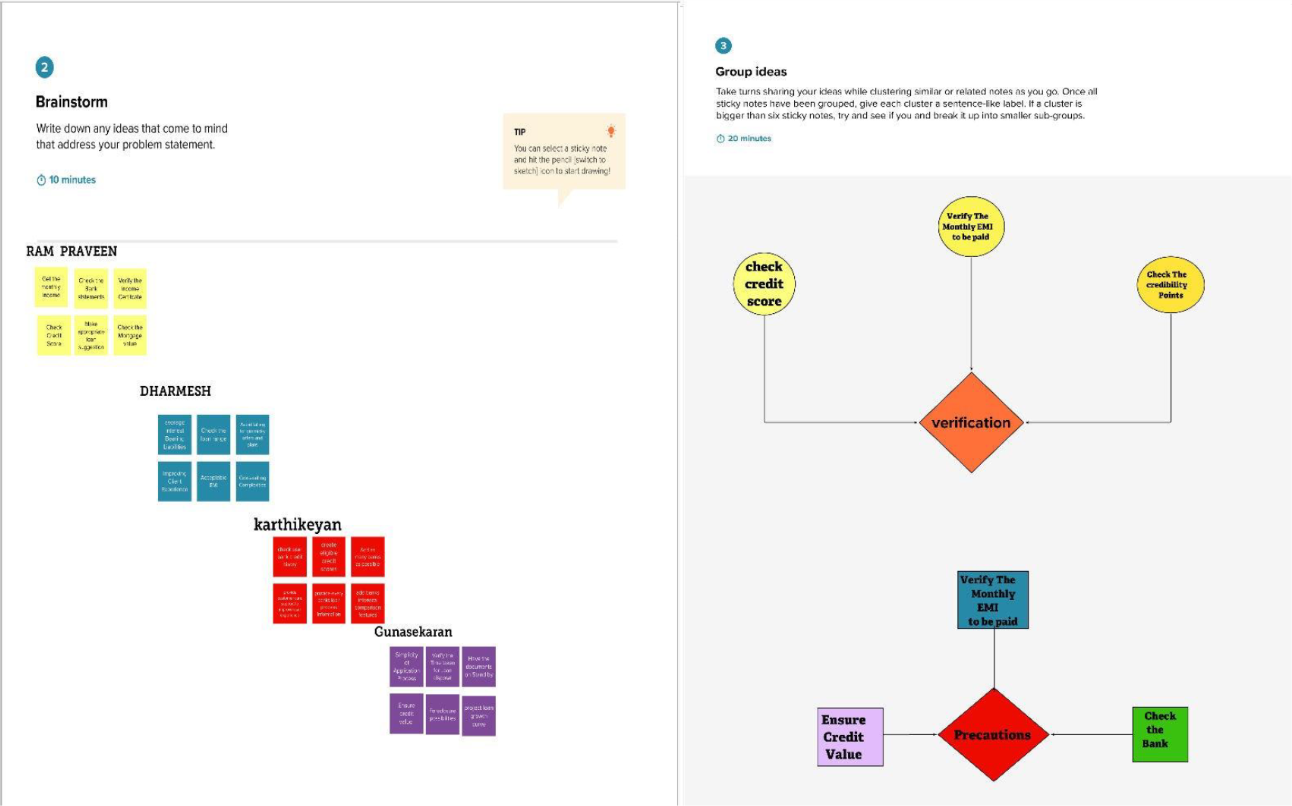
**CHAPTER - 3**

**IDEATION AND PROPOSED SOLUTION**

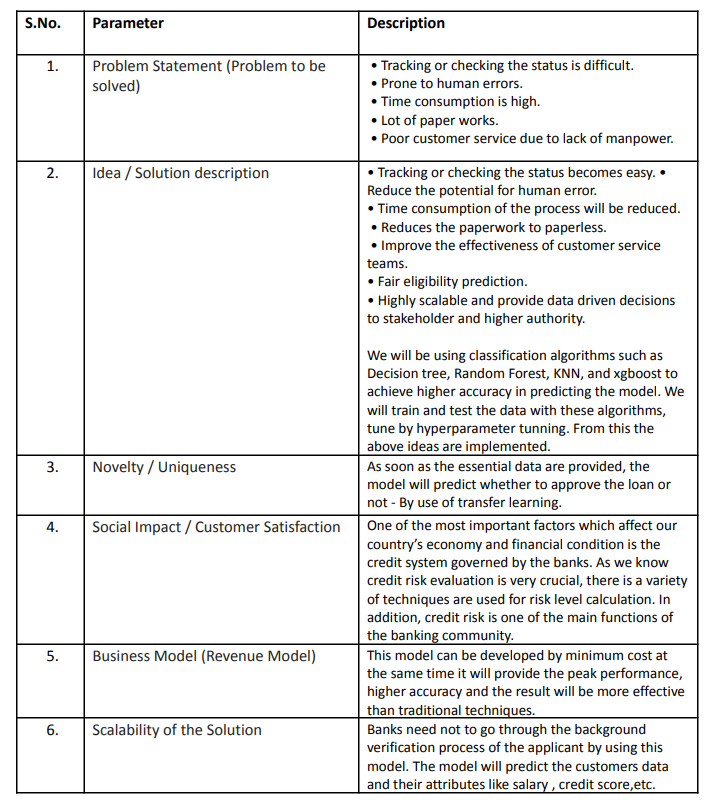
**3.1 Empathy Map Canvas :**

****

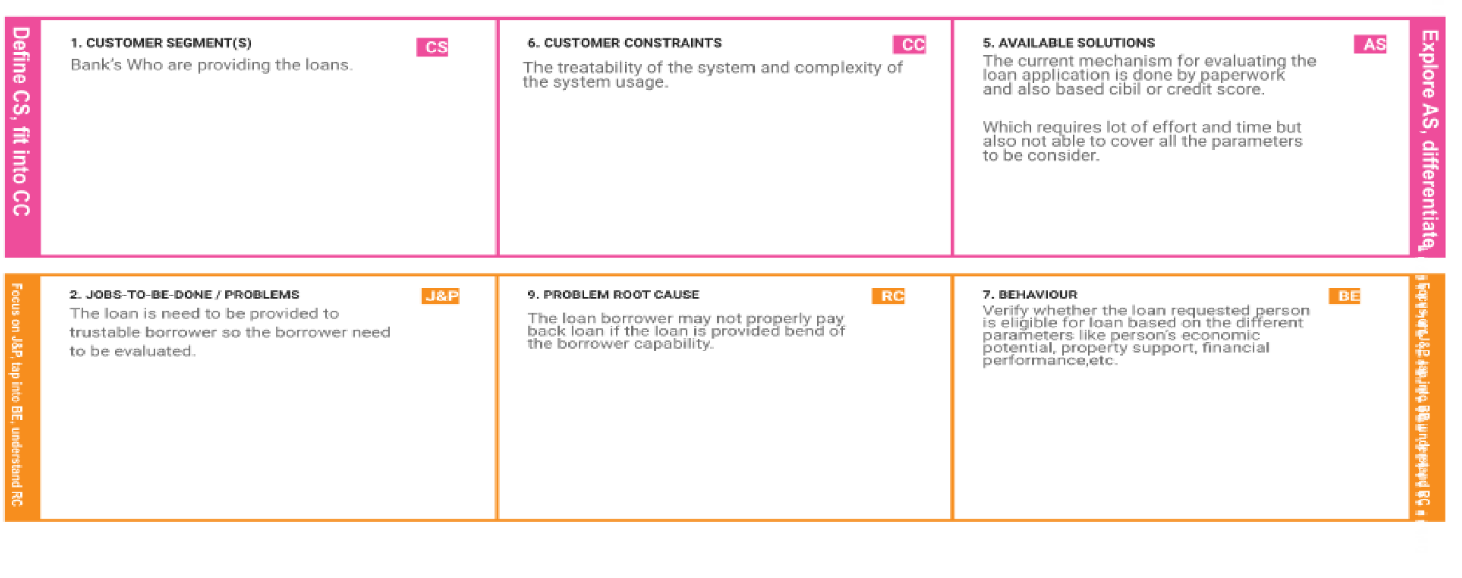
**3.2 Ideation and Brainstorming :**

****

**3.3 Proposed Solution :**

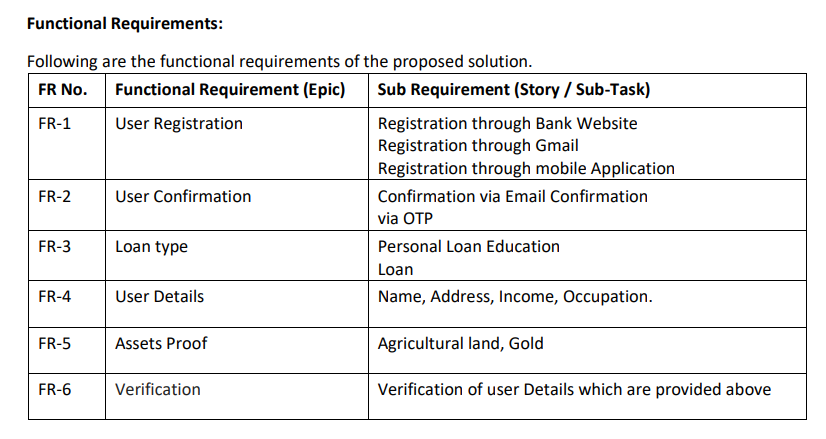
****

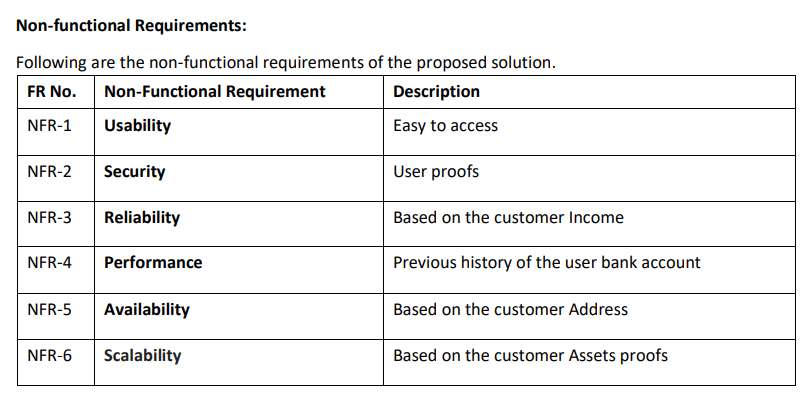
**3.4 Problem Solution Fit :**

****

**CHAPTER - 4**

**REQUIREMENT ANALYSIS**

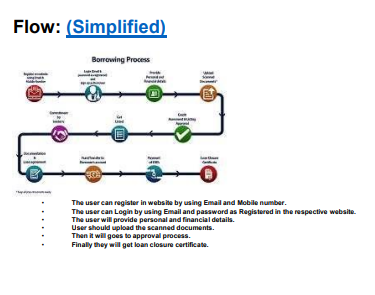
****

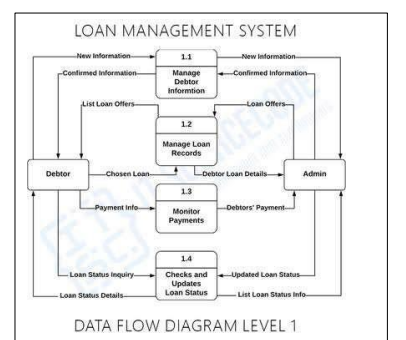
****

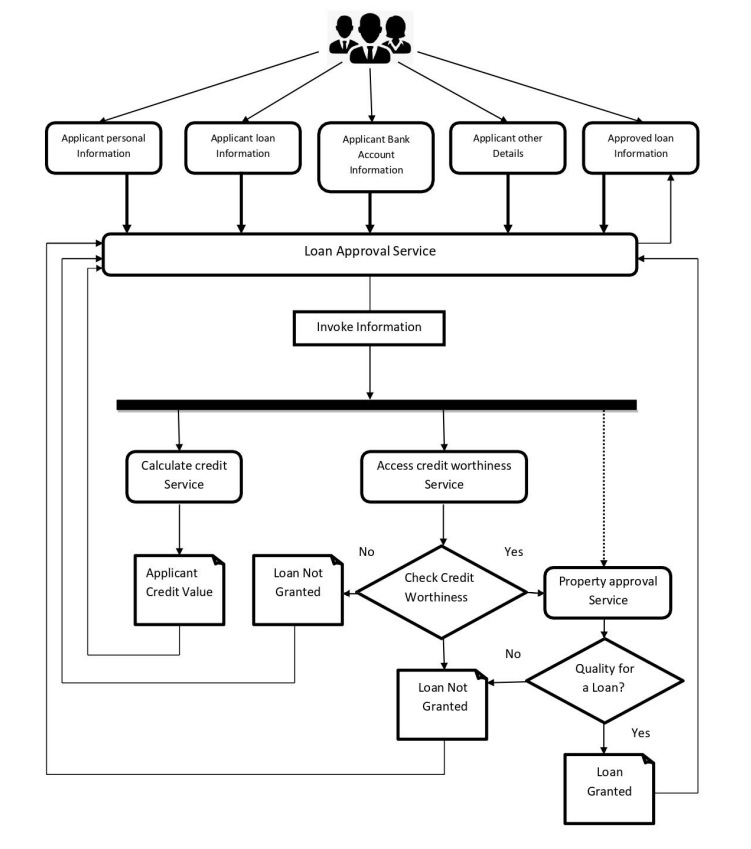
**CHAPTER - 5**

**PROJECT DESIGN**

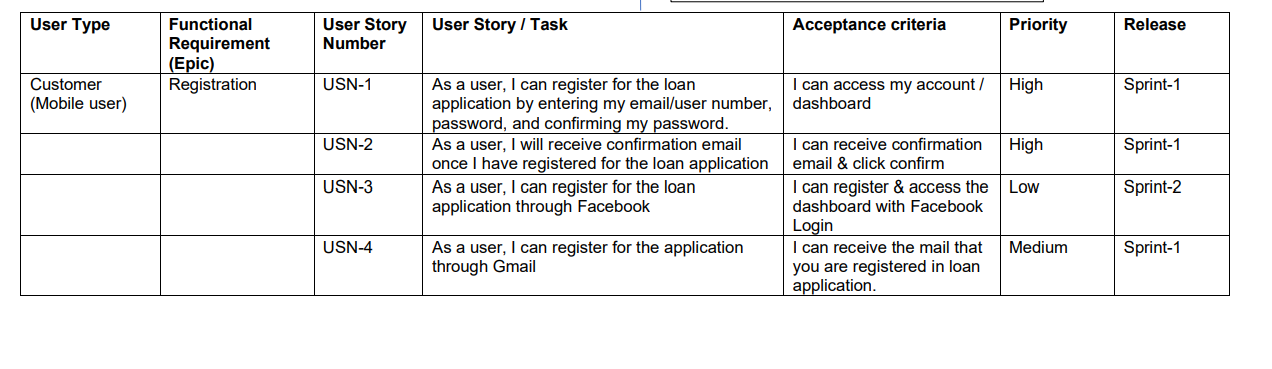
**5.1 Data Flow Diagrams :**

****

****

**5.2 Solution and Technical Architecture :**

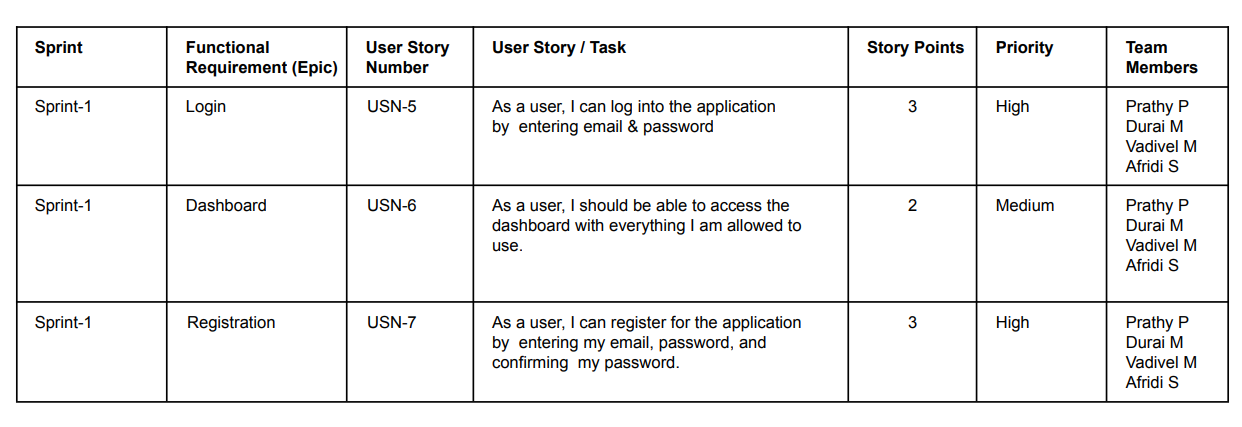
**5.3 User stories:**

****

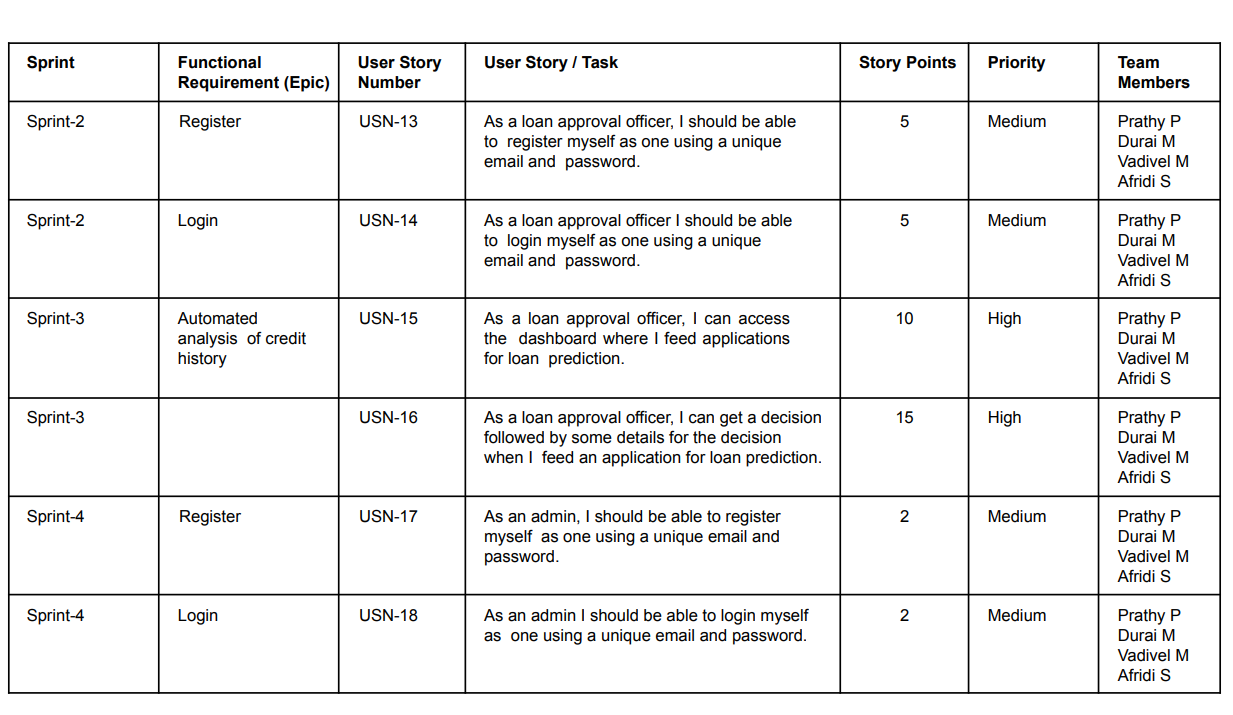
**CHAPTER - 6**

**PROJECT PLANNING AND SCHEDULING**

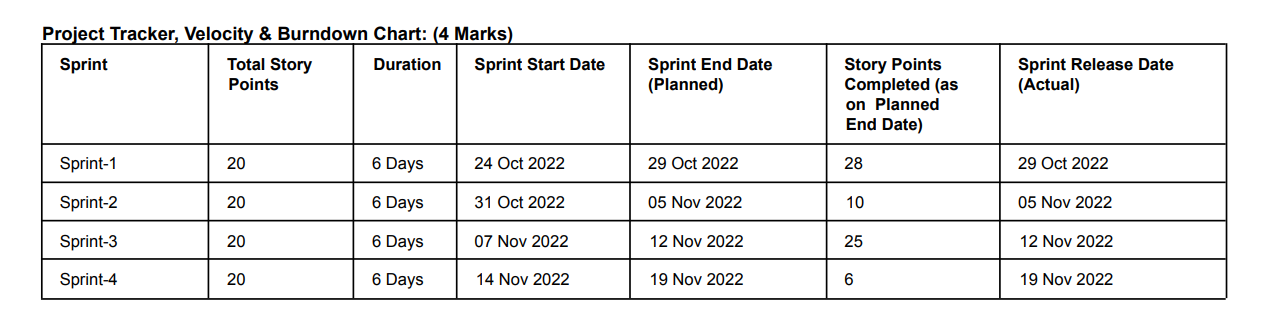
**6.1 Sprint Planning and Estimation :**

****

**6.2 Sprint Delivery Schedule :**

****

**6.3 Reports from JIRA :**



**CHAPTER - 7**

**CODING AND SOLUTIONING**

**7.1 Feature 1**

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;

}

**7.2 Feature 2**

@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);

}

**CHAPTER - 8**

**TESTING**

**8.1 Test Cases :**

* Check the working condition of **Temperature** sensor
* Verify the working of **Humidity** sensor
* Check the working condition of **soil moisture** sensor
* Check the [**Performance**](https://www.testrigtechnologies.com/service/performance-testing/) **of the app** on the different internet networks
* Check whether the application is working for **real-time updates**
* Check **response** when a Login Button is pressed

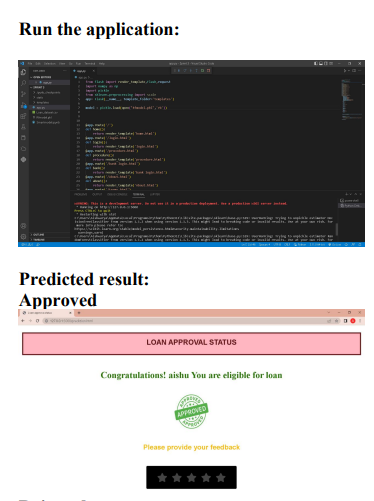
**8.2 User Acceptance Testing :**

User acceptance testing, also called end-user, user acceptability testing, or beta testing, is the process of testing software by the clients or users to see if the product is acceptable for release or not. The testers are familiar with the software’s business requirements, so they can adequately gauge the product’s readiness.

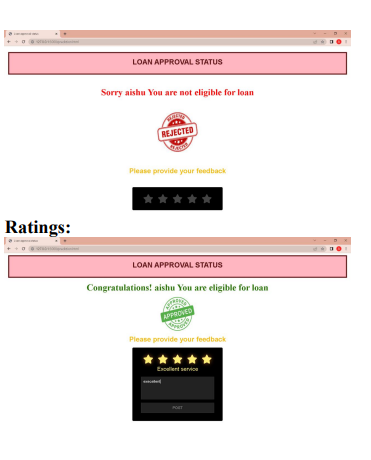
**CHAPTER - 9**

**RESULTS**

**9.1 Performance Metrices :**

****

**Rejection:**

****

**CHAPTER - 10**

**ADVANTAGES AND DISADVANTAGES**

**Advantages :**

The Loan Prediction System can automatically calculate the weight of each feature taking part in loan processing and on new test data the same features are processed with respect to their associated weight **.**

**Disadvantages :**

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

**CHAPTER - 11**

**CONCLUSION**

Machine learning algorithms play a significant role in predicting the risks of bank

loans and decision support systems. The choice of the algorithm used to make the decision(whether the borrower will default), which is the key to addressing decision management when issuing a loan. In this paper, the performance of machine learning algorithms has been tested and their performance compared to standard measurements used on a dataset that includes 1000 loans and their repayment status. Finally, the results showed the possibility of using the proposed algorithms for this purpose with acceptable accuracy rates and superiority of the neural networks for this purpose.

**CHAPTER - 12**

**FUTURE SCOPE**

The system is trained on old training dataset so future software can be made such that new testing data should also take part in training data after some fixed time.

**CHAPTER - 13**

**APPENDIX**

**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)**

**Github Link:**

https://github.com/IBM-EPBL/IBM-Project-19332-1659696311

**Project Demo Link :**

[**https://we.tl/t-hDTq7wJErB?src=dnl**](https://we.tl/t-hDTq7wJErB?src=dnl)