**HIS-ERP**

**HEALTHCARE APP**

**Project report of**

**Final Year, Semester-VI**

**Submitted in the partial fulfillment of the requirements for the degree of**

**Master of Computer Application**

**(MCA)**

**By**

**Tanvi Wagle**

**SAP ID: 70611018020**

**ROLL NO: A045**

**Name of the Company**

**Name of the Company Mentor**

**Under the guidance of**

**Prof. Krishna Samdani**



**Department of Computer Engineering**

**Mukesh Patel School of Technology Management & Engineering**

**NMIMS (Deemed-to-be University), Mumbai**

**Certificate of Completion**

This is to certify that the dissertation entitled **“\_\_\_\_\_\_\_\_\_\_\_”** carried out by\_\_\_\_\_\_\_\_\_\_\_\_\_, student of **Master of computer application** [2017-2018], of **Mukesh Patel School of Technology Management & Engineering, Mumbai**, is hereby accepted and approved as a credible work. Submitted in the partial fulfillment for the requirement of degree of MCA from **NMIMS (Deemed-to-be University), Mumbai,** It’s a bona –fide record of the work done by him/her under my supervision during his/her stay as a project trainee at **Company Name** From \_\_\_\_\_\_\_ to \_\_\_\_\_\_\_\_\_\_. The system has been implemented and running successfully. We do not have any objection on the content of the report.

**Sign & Name**

(Project Leader)

Company Name

**Shri Vile Parle Kelavani Mandal’s**

**NMIMS (Deemed-to-be University)**

**Mukesh Patel School of Technology Management & Engineering**

**JVPD Scheme, Bhaktivedanta swami Marg,**

**Vile Parle (w), Mumbai- 400 056.**

**This is to certify that the project report entitled**

**HIS-ERP, HEALTHCARE APP**

**Submitted by**

**TANVI WAGLE**

**has successfully completed the project report required for the partial fulfillment of Master of Computer Application(MCA) Degree as per the norms prescribed by NMIMS during the VI Semester of the academic year 2019-2020. The project report has been assessed and found to be satisfactory.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Company Mentor’s Sign Internal Mentor’s Sign**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Examiner 1 Sign Examiner 2 Sign**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**DEAN**

**DECLARATION**

I/We, **Tanvi Wagle** , hereby declare that the project work entitled  **“Project Name”** is an authenticated work carried out by me at **Company Name** Under the guidance of **Company Mentor Name** for the partial fulfillment of the award of the degree of **Master of Computer Application** and this work has not been submitted for similar purpose anywhere else except to **Mukesh Patel School of Technology Management & Engineering, Mumbai,** affiliated **to NMIMS (Deemed-to-be Univiersity),Mumbai.**

**Date:**

**Place: Sign &**

**Name of Students**

**ACKNOWLEDGEMENT**

**INDEX OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
|  | **INDUSTRY INTERNSHIP** |  |
| 1 | Introduction   * 1. About Organization   2. About Team and Project   3. Products and Services Provided | 1 |
| 2 | Introduction to Project  2.1 Project Profile  2.2 Introduction to system  2.3 Scope and Applicability of the Project | 2 |
| 3 | System Requirement Specification  3.1 Functional Requirements  3.2 Software Requirements | 3 |
| 4 | Implementation  4.1 Features/ Modules  4.2 Functional Design (Screen Shots)  4.3 Work Done on Project | 4 |
| 5 | Key Learnings from Internship | 7 |
| 6 | Challenges During Internship | 7 |
|  |  |  |
|  | **INTERNAL PROJECT** |  |
| 7 | Introduction to Project  7.1 Project Profile  7.2 Introduction to system  7.3 Scope and Applicability of the Project  7.4 Software Requirements | 8 |
| 8 | Implementation  8.1 Features/ Modules  8.1.1 Modules of the project  8.1.2 Features of the project  8.2 Code  8.3 Functional Design (Screen Shots)  8.3 Work Done on Project | 10 |
| 9 | Key Learnings |  |
| 10 | Challenges |  |
| 11 | Conclusion and Future Expansion |  |
| 12 | References |  |

**INDUSTRY INTERNSHIP**

1. **INTRODUCTION**

**1.1 ABOUT ORGANIZATION:**

* Infoway is a group of IT professionals, Chartered Accountants, Advocates, Govt. Certified Auditor - Co-operative Societies, having expertise in Real Estate Industry since 1993.
* They provide services to Builders for ERP Solutions, Digital Marketing, Property Marketing, RERA Consultancy, Society Management, Finance and Solar Power Management.

**1.2 ABOUT TEAM AND PROJECT:**

* TEAM: INDIVIDUAL
* PROJECT NAME: HIS-ERP
  + MODULES: SALES, MARKETING, MATERIAL MANAGEMENT

**1.3 SERVICES AND PRODUCTS PROVIDED:**

Infoway provides IT Solutions, HIS-ERP, RERA, Solar Energy and Construction Finance.

1. **HIS-ERP PROJECT**

**2.1 Project Profile**

* **Project Title** **:** HIS-ERP
* **Project Definition:**  HIS-ERP is a user-friendly ERP solution for builders and developers. It helps the builders to keep track of the sales, materials used during construction, marketing, etc.
* **Developed for:** Infoway, Kandivali (West)
* **Project Coordinator :** Mr. Vijay Vyas
* **Project Guide:**  Prof. Krishna Samdani
* **Project Duration:** 5 weeks
* **Submitted to:** Mukesh Patel School of Technology Management and Engineering, Mumbai
* **Developed by :** Tanvi Wagle

**2.2 INTRODUCTION TO SYSTEM:**

* HIS-ERP is a user-friendly ERP solution for builders and developers.
* It helps the builders to keep track of the sales, materials used during construction, marketing, etc.
* Builders can access HIS-ERP 24x7 from anywhere and on any device.
* It also allows the user to download reports in pdf, excel sheet.

**2.3 SCOPE AND AVAILABILITY**

The main aim of this project is to help builders and developers to keep track of the materials used on the construction site, sell flats, keep track of the call details, help them to manage the documentation, access the information from any device 24x7. The website also helps the sales team to get information about the flat; information like if the flat is sold or not, if sold then for what price, etc.

1. **SYSTEM REQUIREMENT SPECIFICATION**

**3.1 FUNCTIONAL REQUIREMENTS**

The application is a Housing Industry Software created for Builders and Developers teams. It can help these users to keep track of the information about the materials used, sales done, keep call details, sales details, manage documentation, download pdf and excels sheets, etc.

**3.2 SOFTWARE REQUIREMENTS**

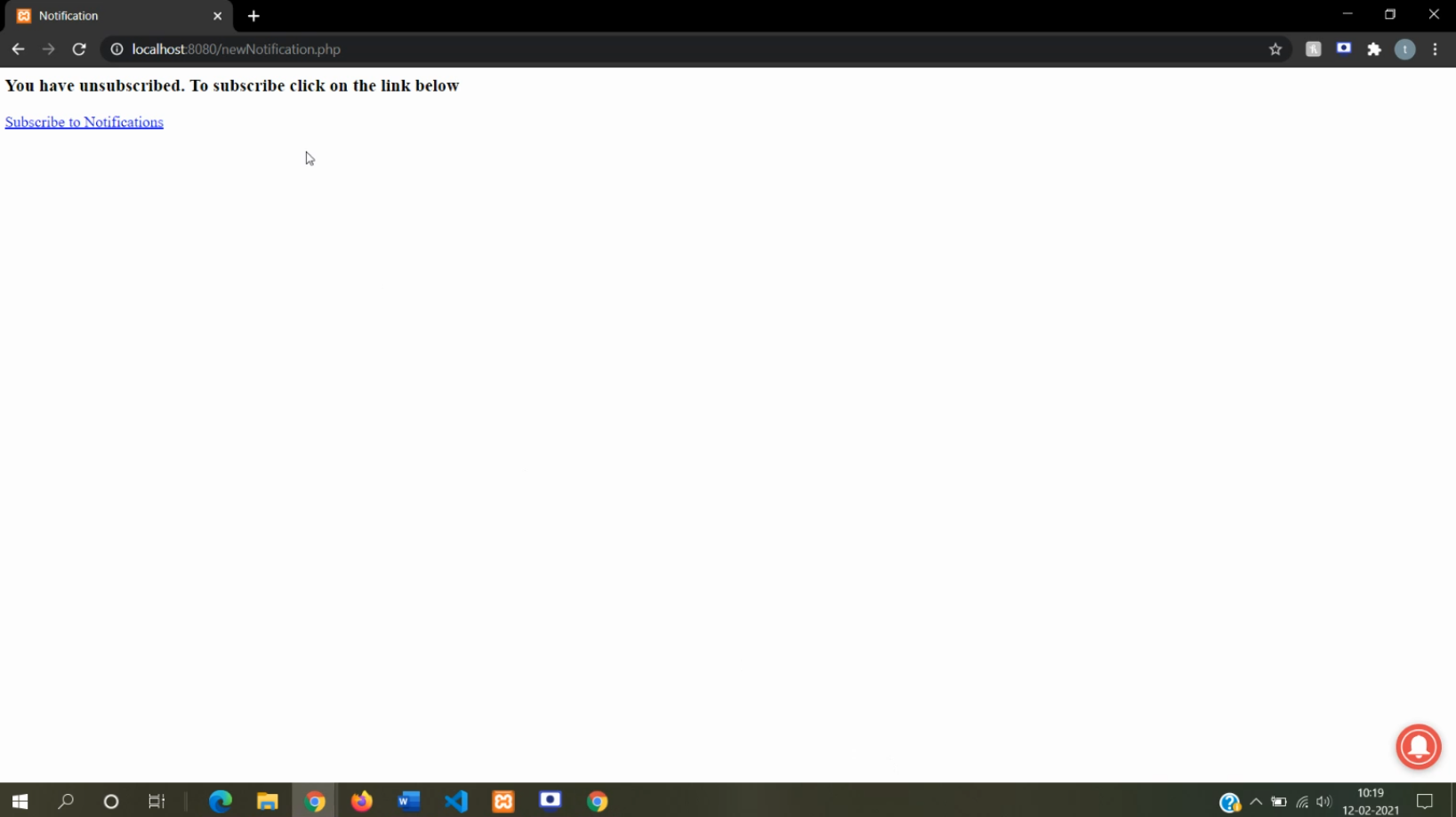
* HTML, CSS, JavaScript
* MySQL Workbench
* Codeigniter Framework
* OneSignal API

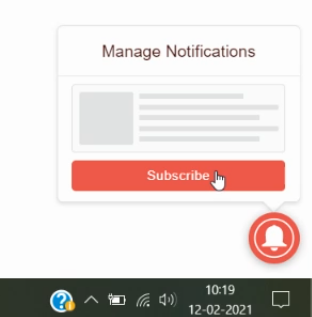
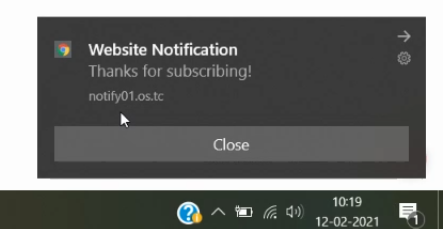
1. **IMPLEMENTATION**

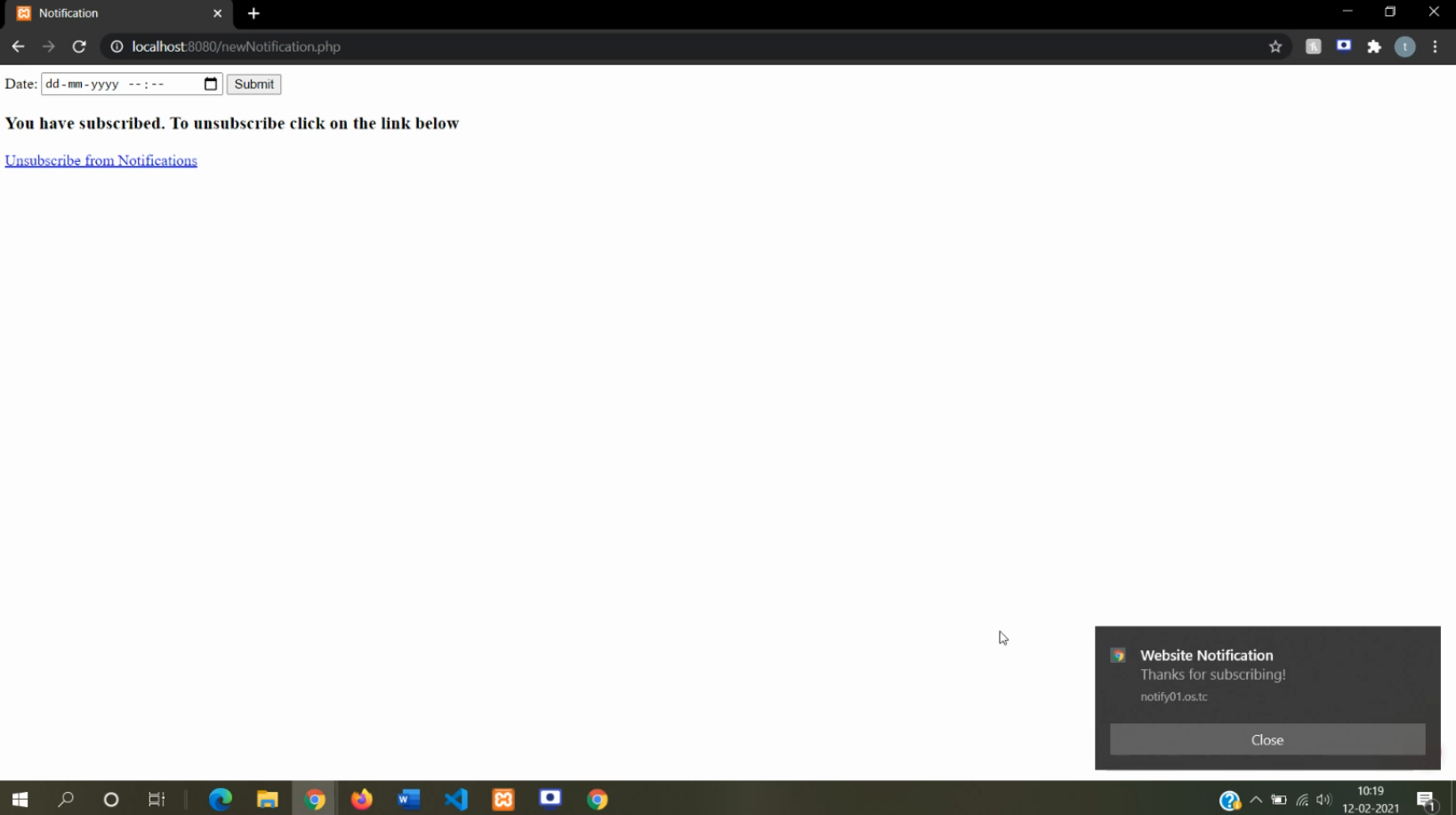
**4.1 FEATURES/ MODULES:**

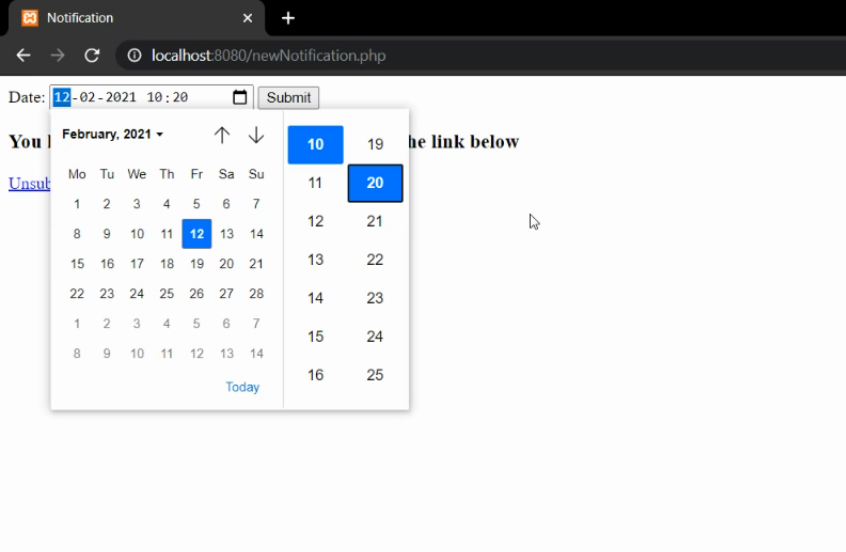
* **HIS-ERP SALES:**
  + This module is for sales team of the company. It stores information about premises sold in a building. Premises like Flat, shop, etc.
  + It shows building wise and flatwise details of the owner showing payment details (paid amount and due amount).
  + It shows the call logs and remarks of the follow ups. New followups can be added and it gives notification for the followup call.
* **HIS-ERP MARKETING**:
  + This module helps for marketing of project. Allows the marketing team keep track of interested people by entering call time, remarks, next call time for reference.
  + The followup will give notification at next call time.
* **HIS\_ERP MATERIAL MANAGEMENT:**
  + This module helps the builder to keep track of the material used, material left in stock, material ordered.
  + The report can be printed or downloaded as pdf. The report generated can be datewise or individual record.

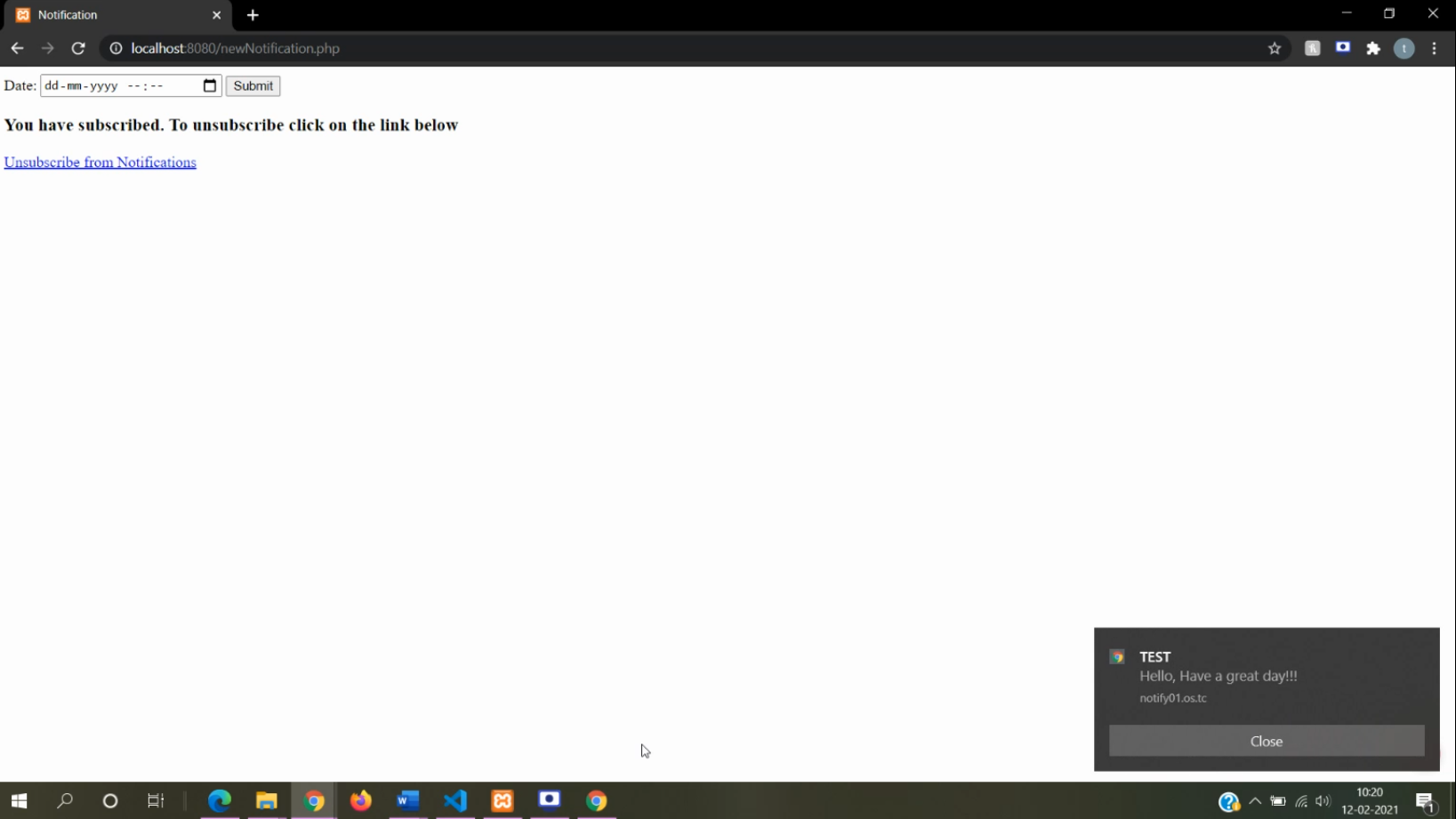
**4.2 FUNCTIONAL DESIGN (SCREEN SHOTS)**

****

** **

****

****

****

**4.3 WORK DONE ON PROJECT**

* **HIS-ERP SALES:**

**FOR WEBSITE:**

* + Added tabs for followup and receipt and made changes in the UI.
  + Receipts tab shows the details of paid amount by the owner.
  + Followup tab shows the call details made by the sales team. The sales team can add new followup record. It also gives push notifications on the next call time with the details of the owner name, flat number, premises number and the remarks. The team can delete, edit followup.
  + Added new functionality called Recovery Followup
  + It shows the call details that are to be made Today, Tomorrow and the calls that are pending.

**FOR FLUTTER APP:**

* + Made login page, included validation, connectivity to database using REST API.
  + Made dashboard charts.
* **HIS-ERP MARKETING:**
  + Added push notifications functionality to the followup calls.
* **HIS\_ERP MATERIAL MANAGEMENT:**
  + Made UI changes in entire module.
  + Added edit, delete icons on manage inventory like inward, outward, indent, purchase order.
  + Made UI changes in report printing.

1. **Key Learnings from Internship**

* Development using codeigniter for websites.
* Flutter development for sales applications
* Add push notifications to website using OneSignal

1. **CHALLENGES**

* Learning new technologies and understanding the existing project to add new modules.
* Getting familiar to codeigniter and flutter.
* Adding push notifications.

**INTERNAL PROJECT**

1. **INTRODUCTION TO PROJECT**

**7.1 PROJECT PROFILE**

* **Project Title:** HEALTHCARE APP
* **Project Definition:**  Healthcare app is a website which uses Machine Learning models to predict whether the user has disease or not based on the inputs given by the user.
* **Project Guide:**  Prof. Krishna Samdani
* **Project Duration:** 12 weeks
* **Submitted to:** Mukesh Patel School of Technology Management and Engineering, Mumbai
* **Developed by :** Tanvi Wagle

**7.2** **INTRODUCTION TO SYSTEM**

Healthcare app is a website that uses Machine learning model which will help the users to predict if they suffer from any disease like heart disease, liver disease based on the inputs given by them.

The website shows some analytical graphs which will give information like: how many people suffer from the disease?

This website gives information about the symptoms for the disease and also it gives preventive measures that one can follow.

**7.3 SCOPE AND APPLICABILITY OF THE PROJECT**

The main objective of this project is to reduce the workload of doctors by using machine learning model to predict if any user is suffering from certain disease or not. Information about the symptoms of the disease and what preventive measures one can take is given on the website.

**7.4 SOFTWARE REQUIREMENTS**

* Python 3.7
* Django framework
* HTML 5, CSS 3, Bootstrap 4, JavaScript
* Chart.js JavaScript Library
* Python Libraries
  + Numpy
  + Pandas
  + Matplotlib
  + Seaborn
  + Sklearn
  + Imblearn
  + Pickle

IDE: Visual Studio Code, Jupyter Notebook

1. **Implementation**

**8.1 FEATURES/ MODULES**

**8.1.1 Modules of Project:**

* **Models**:
  + **Heart disease prediction:**
    - Heart disease is the leading cause of death for people of most racial and ethnic groups.
    - This model will predict heart disease based on different parameters like age, gender, cholesterol level, blood pressure level, etc.
  + **Liver disease prediction:**
    - Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs.
    - This model will predict liver disease based on different parameters like age, gender, total bilirubin, direct bilirubin, etc.
  + **Brain stroke prediction:**
    - According to the World Health Organization (WHO) stroke is the 2nd leading cause of death globally, responsible for approximately 11% of total deaths.
    - This model will predict whether a patient is likely to get stroke based on the input parameters like gender, age, various diseases, and smoking status.
* **Symptom/ Preventions for diseases:**
  + It will show information about symptoms for the disease as well as the preventive measures that can be followed.

**8.1.2 Features of Project:**

* Machine Learning model will predict if the user has disease or not based on the inputs given by the user.
* View analytical graphs based on the disease.
* View symptoms for any disease.
* View preventive measures for diseases.

**8.2 CODE**

**Code for Machine learning Models:**

**Heart\_Disease Model:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import pickle

from sklearn.linear\_model import LinearRegression, LogisticRegression

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support, mean\_squared\_error, roc\_auc\_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import BaggingClassifier, RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

# Reading dataset

df\_train = pd.read\_csv(r'..\Datasets\heart.csv')

# Rename columns for better readability

df\_train.columns = ['age', 'sex', 'chest\_pain\_type', 'resting\_blood\_pressure', 'cholesterol', 'fasting\_blood\_pressure',

'rest\_ecg', 'max\_heart\_rate\_achieved', 'exercise\_induced\_angina', 'st\_depression', 'slope',

'num\_major\_vessels', 'thal', 'target']

# One hot encoding

df\_train = pd.get\_dummies(data = df\_train, columns = ['sex', 'chest\_pain\_type', 'fasting\_blood\_pressure', 'rest\_ecg',

'exercise\_induced\_angina', 'slope', 'num\_major\_vessels', 'thal'], dtype= 'int')

# Training model

X = df\_train.drop('target', axis = 1)

y = df\_train['target']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state = 5)

# Logistic Regression

logistic = LogisticRegression(max\_iter=10000)

logistic.fit(X\_train, y\_train)

pred = logistic.predict(X\_test)

#Calculating accuracy

score = np.around(accuracy\_score(y\_test, pred), decimals = 3) \*100

# Tuple of model and score

model\_score = (logistic, score)

# Dumping model and score

pickle.dump(model\_score, open('Heart\_Disease.sav', 'wb'))

**Liver\_Disease Model:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import math

from imblearn.combine import SMOTETomek

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support, mean\_squared\_error, roc\_auc\_score

from sklearn.tree import DecisionTreeClassifier

# Reading Dataset

df\_train = pd.read\_csv(r'..\Datasets\indian\_liver\_patient.csv')

# There are 4 null values in one column of the dataset.

# Adding mean value to the null values

mean\_ratio = df\_train['Albumin\_and\_Globulin\_Ratio'].mean()

def fill\_ratio(df\_train, mean\_ratio):

if math.isnan(df\_train['Albumin\_and\_Globulin\_Ratio']):

return mean\_ratio

else:

return df\_train['Albumin\_and\_Globulin\_Ratio']

df\_train['Albumin\_and\_Globulin\_Ratio'] = df\_train.apply(fill\_ratio, axis = 1, args = (mean\_ratio, ))

# Renaming column for better readability

df\_train.rename(columns = {'Dataset': 'Response'}, inplace = True)

# converting categorical values to numerical values

df\_train = pd.get\_dummies(df\_train, columns = ['Gender'], dtype= 'int')

# Model

X = df\_train.drop('Response', axis = 1)

y = df\_train['Response']

# Oversampling the dataset

smk = SMOTETomek(random\_state = 1, sampling\_strategy = 'minority')

x\_res, y\_res = smk.fit\_resample(X, y)

# Scaling

standardScaler = StandardScaler()

scale\_columns = ['Age', 'Total\_Bilirubin', 'Direct\_Bilirubin', 'Alkaline\_Phosphotase', 'Alamine\_Aminotransferase',

'Aspartate\_Aminotransferase', 'Total\_Protiens', 'Albumin', 'Albumin\_and\_Globulin\_Ratio']

df\_train[scale\_columns] = standardScaler.fit\_transform(df\_train[scale\_columns])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x\_res, y\_res, test\_size = 0.3, random\_state = 50)

# Decision Tree

dt\_clf = DecisionTreeClassifier()

dt\_clf.fit(X\_train, y\_train)

pred = dt\_clf.predict(X\_test)

#Calculating accuracy

score = np.around(accuracy\_score(y\_test, pred), decimals = 3) \*100

# Tuple of model and score

model\_score = (dt\_clf, score)

# Dumping model and score

pickle.dump(model\_score, open('Liver\_Disease.sav', 'wb'))

**Brain\_Stroke Model:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import math

from sklearn.linear\_model import LinearRegression, LogisticRegression

from sklearn.svm import SVC

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, precision\_recall\_fscore\_support, mean\_squared\_error, roc\_auc\_score

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import BaggingClassifier, RandomForestClassifier

from sklearn.neighbors import KNeighborsClassifier

# Read dataset

df\_train = pd.read\_csv(r'..\Datasets\healthcare-dataset-stroke-data.csv')

# Removing irrelevant columns

df\_train.drop(columns = 'id', axis = 1, inplace = True)

# Handling null values

mean\_bmi = df\_train.groupby(['gender', 'age']).mean()['bmi']

mean\_bmi = np.around(mean\_bmi, decimals = 3)

def fill\_bmi(df\_train, mean\_bmi):

if math.isnan(df\_train['bmi']):

return mean\_bmi[df\_train['gender']][df\_train['age']]

else:

return df\_train['bmi']

df\_train['bmi'] = df\_train.apply(fill\_bmi, axis = 1, args = (mean\_bmi, ))

df\_train['bmi'].iloc[2030] = mean\_bmi['Female'][0.48]

df\_train.drop(index = 3116, inplace = True)

# One hot encoding

df\_objects = df\_train.select\_dtypes(include = 'object')

df\_train = pd.get\_dummies(df\_train, columns= df\_objects.columns, dtype = 'int')

# Model

X = df\_train.drop('stroke', axis = 1)

y = df\_train['stroke']

# Oversampling

smk = SMOTETomek(random\_state = 2, sampling\_strategy = 'minority')

x\_res, y\_res = smk.fit\_resample(X, y)

# train test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x\_res, y\_res, test\_size = 0.3, random\_state = 17)

#Logistic Regression

logistic = LogisticRegression(max\_iter=10000)

logistic.fit(X\_train, y\_train)

pred = logistic.predict(X\_test)

#Calculating accuracy

score = np.around(accuracy\_score(y\_test, pred), decimals = 3) \*100

# Tuple of model and score

model\_score = (logistic, score)

# Dumping model and score

pickle.dump(model\_score, open('Stroke\_Prediction.sav', 'wb'))

**8.3 FUNCTIONAL DESIGN (SCREEN SHOTS)**

**8.4 WORK DONE ON PROJECT**

1. Search and download of datasets.
2. Data preprocessing.
3. Exploratory Data Analysis.
4. Selecting model based on score
5. Making user interface for the website
6. Integrating model into website
7. Adding charts using chart.js
8. Add Symptoms/ Preventions
9. **Key Learnings**

* Learning various machine learning models, EDA.
* Learning Django as web framework for python.
* Integration of model with User interface.
* Add chart on website using chart.js and how to change its configurations

1. **Challenges**

* Balancing the dataset and optimize the model.
* Add graphs on the website using chart.js

1. **Conclusion and Future Expansion**

**Conclusion:**

This project is very crucial as it is related to medical. One wrong prediction can mislead a patient. It is very important to optimize the model. While making this project, I have learnt about different machine learning models and to calculate their scores and how to optimize the model. Also, I have about different python libraries like imblearn, sklearn, seaborn, etc.

**Future Expansion:**

I would like to integrate concept of image processing in this project based on which various other disease could be predicted.

1. **References**

* Datasets:
  + <https://www.kaggle.com/uciml/indian-liver-patient-records>

Upload Date: 2017-09-20

* + <https://www.kaggle.com/fedesoriano/stroke-prediction-dataset>

Upload Date: 2021-01-27

* + <https://www.kaggle.com/ronitf/heart-disease-uci>

Upload Date: 2018-06-25

* Courses completed for this project:
  + Python and Django Bootcamp:

<https://www.udemy.com/certificate/UC-15549f73-e326-4d86-a325-da196f784ed9/>

* + Data Science and Machine Learning Bootcamp: <https://www.udemy.com/certificate/UC-f6d4fa22-680e-4e67-8496-6e35fe077058/>
* <https://towardsdatascience.com/creating-a-machine-learning-based-web-application-using-django-5444e0053a09>
* https://www.chartjs.org/docs/latest/