MINI PROJECT – II

(2021-22)

REMOTE PASSWORD MANAGEMENT SYSTEM
PROJECT REPORT

Institute of Engineering & Technology

Team Members:

Achal Paliwal
(181500032)

Prashant Agrawal
(181500486)

Shashank Gupta
(181500653)

Supervised By-

Mrs. Harvinder Kaur
Technical Trainer
Department of Computer Engineering & Applications

GLA UNIVERSITY

MATHURA -281406, INDIA

April 2021

Contents

ACKNOWLEDGEMENT

We take this opportunity to thank all those who have helped us in completing the project successfully.

We would like to express our gratitude to **Mrs. Harvinder Kaur**, who as our guide/mentor provided us with every possible support and guidance throughout the development of project. This project would never have been completed without his encouragement and support.

Our heartiest thanks to Dr. (Prof). **Anand Singh Jalal**, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

Department of Computer Engineering and Applications

GLA University, 17 km. Stone NH#2, Mathura-Delhi Road, Chaumuha, Mathura – 281406 U.P (India)

Declaration

We hereby declare that the work which is being presented in the Mini Project 2 "**REMOTE PASSWORD MANAGEMENT SYSTEM"** in partial fulfilment of the requirements for Mini Project 2 viva voce, is an authentic record of our own work carried under the supervision of "**Mrs. Harvinder Kaur**".

Signature of Candidate:
Name of Candidate: Achal Paliwal
Roll. No.: 181500032

Name of Candidate: Prashant Agrawal
Roll. No.: 181500486

Name of Candidate: Shashank gupta
Roll. No.: 181500653

Abstract

The project named "**Remote Password Management System"**, is a Web-based application created by using front end and back-end technologies like Node.js, Express, EJS. The purpose of this project is to provide user a platform where he or she can store their social platform passwords and important notes so that it can be easily available by just one login from anywhere. To maintain their privacy of passwords and notes we have implemented AES Encryption technique for storing password in encrypted format in the database. And for security concerns of account the mail verification method is implemented, on login time user have to verify the OTP sent on his/her mail.

Chapter 1. Introduction

1.1 Motivation

The motivation behind this project is to provide better privacy and accessibility for a user. Because Now a days Every person has some important notes and so many IDs and passwords to remember and it is difficult to remember all of these. Therefore, through this project we are providing facilities to store passwords in encrypted format and notes with login protection and user can access these notes and password globally.

1.2 Overview

Today every person has generally many accounts on different online and offline platforms like Facebook, Google, Gmail, LinkedIn, Twitter and it is very difficult for a person to remember all the IDs and their passwords. Thorough this project we are providing facilitie to access all IDS, passwords and notes globally anytime anywhere by just one login with OTP.

In this project "*Remote Password Management System*" to maintain security and privacy we have implemented AES Encryption to store all password in encrypted format and to maintain the security email verification is used.

1.3 Objective

Objective of this project "REMOTE PASSWORD MANAGEMENT SYSTEM" is to provide Facility to a user to store IDS, password (Encrypted Format) of their social online platforms and make notes which can be easily access globally anytime anywhere with just login and to reduce the time taken in searching notes and password in offline or in disk.

Chapter 2. Software Requirement Analysis

2.1 Problem Statement

We want to provide Facility to a user to store IDS, password (Encrypted Format) and Make Notes which can be access globally anytime anywhere with just login and to reduce the time taken in searching notes and password in offline or in disk.

2.2 Modules

The project is based on several modules:

2.2.1 Product perspective

1. User Interface: The application will have a user-friendly and menu-based interface.

   Following frames will be provided.
2. A login frame for entering the username, the password will be provided. Access to main screen of the model.
3. A registration frame for Registration Purpose.
4. A login frame for entering the username, the password will be provided. Access to main screen of the model.
5. There is a frame for *Adding Password* and *Notes*.
6. There is a frame for displaying *Passwords* and *Notes* and a frame to change the account password.

2.2.2 Product Functions

The Website Name "*Remote Password and Management*" allow access only to authorized users or the user who have registered themselves already in it. A summary of the major functions that the model will perform:

a. Provide functionality to store passwords and to make notes online.

b. User can access theses notes et. all by just login and verifying by email verification.

c. User has to register only single time then he can access it by username and password.

2.2.2.1 User

- ✓ Can login and get registered
- ✓ Can Add Password and Notes.
- ✓ Can Check Notes and Passwords.

2.2.2.2 Administration

- ✓ Maintaining Server.
- ✓ Maintaining Functionality like ***AES Encryption***.

2.2.3 User Characteristics

a. Educational level: Users should be comfortable with the English language.

b. **Experience:** No prior experience is required to operate this website it is user friendly.

c. **Skills:** Users should have basic knowledge and should be comfortable using general purpose applications on computers.

2.3 Specific Requirements

These specific requirements describe the specific constraints impost on the requirements:

- **Hardware Requirement**
    - Processor - Intel i5
    - Operating System – Windows /8/10, Linux, Mac OS
    - RAM – 4GB (minimum)
    - Hard disk – 64 GB
    - Hardware Devices – Computer System
- **Tools Required**
    - Visual Studio
    - Node.js
    - Express.
    - MongoDB (To store data)
    - JavaScript, EJS(Front-End).
    - AES Encryption (For Encryption purpose)

2.4 External Interface Requirement

2.4.1 User Interfaces

- **Registration Screen-** Various fields available on this frame will be:

✓ Registration Email

✓ Password

✓ Confirm password

- **Login Screen-** Fields available on this screen are:

✓ Login Email

✓ Password

✓ OTP

- ✓ Forget password

- **Main Frame**- Various Fields are:

- ✓ Password

    1. Add password

    2. Delete password

- ✓ Dairy

    1. Add note

    2. Delete note

2.4.2 Hardware Interfaces

- Screen resolution of at least 800X600 is required for proper and complete viewing of screens. Higher resolution will be accepted.

2.4.3 Software Interfaces

- Any Windows/Linux/Mac based operating system.

2.5 Technologies used

2.5.1 NodeJS

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project! Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant. A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm. When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and

wasting CPU cycles waiting, Node.js will resume the operations when the response comes back

2.5.2 ExpressJS

Express is a fast, assertive, essential and moderate web framework of Node.js. You can assume express as a layer built on the top of the Node.js that helps manage a server and routes. It provides a robust set of features to develop web and mobile applications.

Let's see some of the core features of Express framework:

It can be used to design single-page, multi-page and hybrid web applications.

It allows to setup middlewares to respond to HTTP Requests.

It defines a routing table which is used to perform different actions based on HTTP method and URL.

It allows to dynamically render HTML Pages based on passing arguments to templates.

2.5.3 MongoDB

- MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time
- The document model maps to the objects in your application code, making data easy to work with
- Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze your data
- MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use
- MongoDB is free to use. Versions released prior to October 16, 2018 are published under the AGPL. All versions released after October 16, 2018, including patch fixes for prior versions, are published under the Server-Side Public License (SSPL) v1.

2.5.4 HTML

HTML or HTML 5 tutorial provides basic and advanced concepts of HTML. Our HTML tutorial is developed for beginners and professionals. In our tutorial, every topic is given step-by-step so that you can learn it in a very easy way. If you are new in learning HTML, then you can learn HTML from basic to a professional level and after learning HTML with CSS and JavaScript you will be able to create your own interactive and dynamic website. But Now We will focus on HTML only in this tutorial.

The major points of HTML are given below:

- HTML stands for HyperText Markup Language.
- HTML is used to create web pages and web applications.
- HTML is widely used language on the web.
- We can create a static website by HTML only.
- Technically, HTML is a Markup language rather than a programming language.

2.5.5 CSS

The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

2.5.6 JavaScript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

2.5.7 EJS

EJS or Embedded Javascript Templating is a templating engine used by Node.js. Template engine helps to create an HTML template with minimal code. Also, it can inject data into HTML template at the client side and produce the final HTML. EJS is a simple templating language which is used to generate HTML markup with plain JavaScript. It also helps to embed JavaScript to HTML pages. To begin with, using EJS as templating engine we need to install EJS using given command:

2.5.8 Bootstrap

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is among the most starred projects on GitHub, with more than 142,000 stars, behind freeCodeCamp and marginally behind Vue.js framework.

Chapter 3. Software Design

3.1 Data Flow Diagram

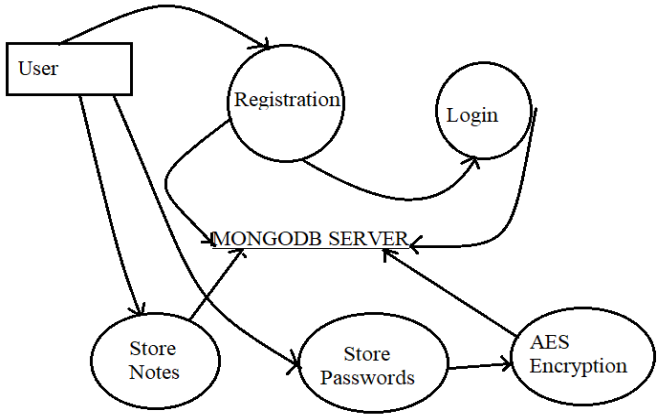


Fig.1 WorkFlow Diagram of "RPMS"

PART 1. Making *Login and Registration page* and adding functionality of email verification for security-related.

PART 2. Designing Internal Interface of website to add password and store notes.

PART 3. Implementation of Encryption Technique like AES to store theses password in encrypted format.

PART 4. If User want, he/she can change their account password again.

PART 5. Testing.

3.2UML Diagrams

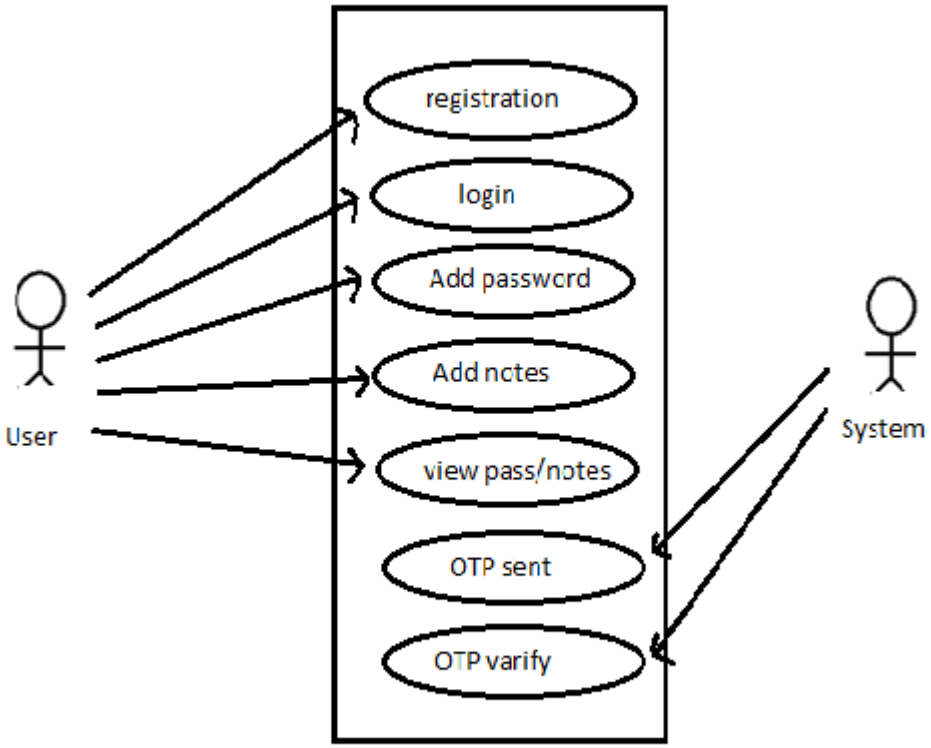


Fig-2

3.2.2 Sequence Diagram

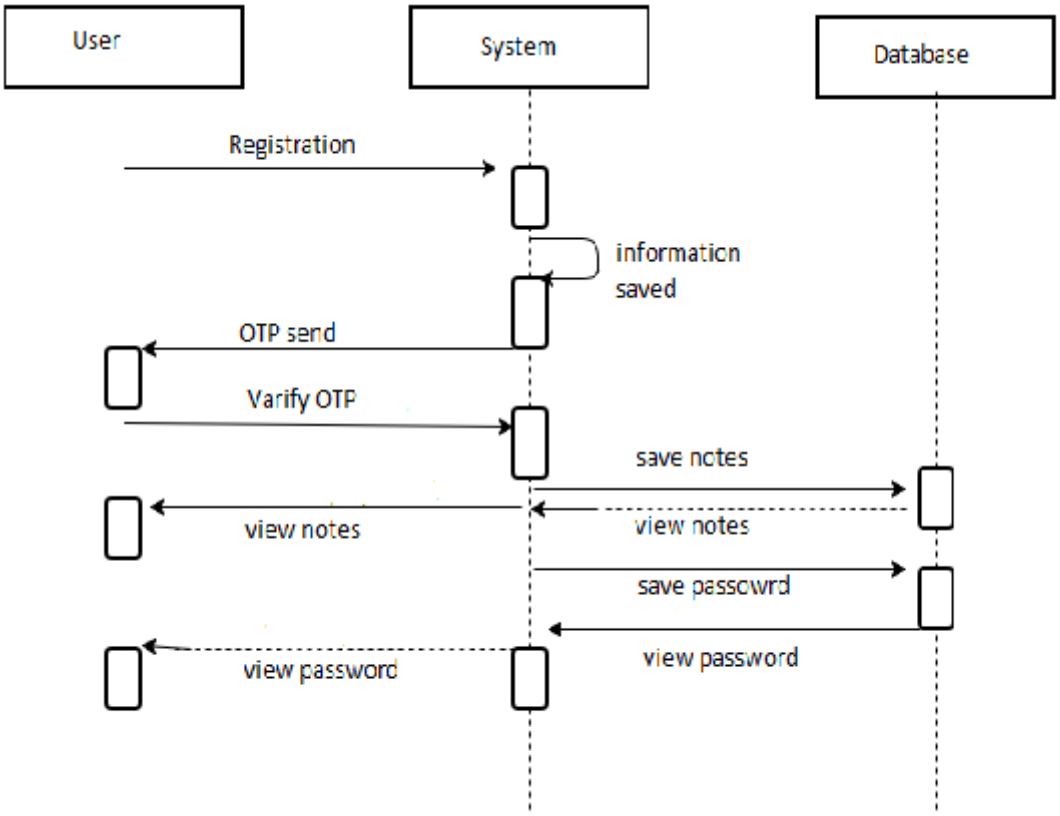
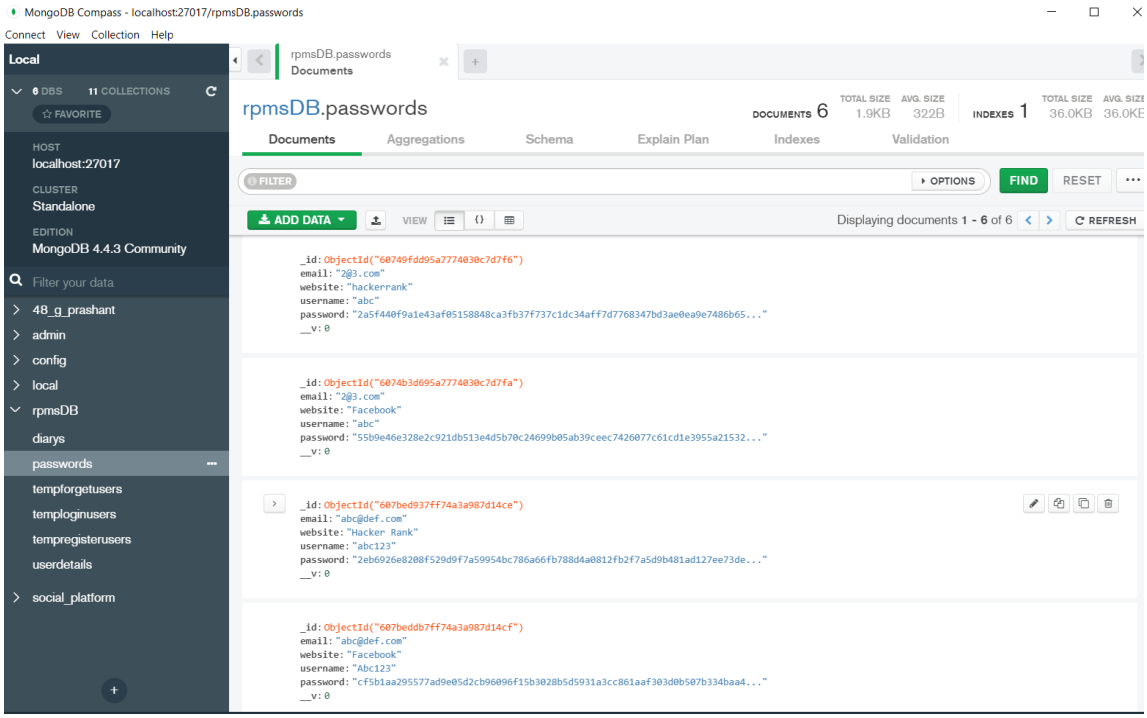
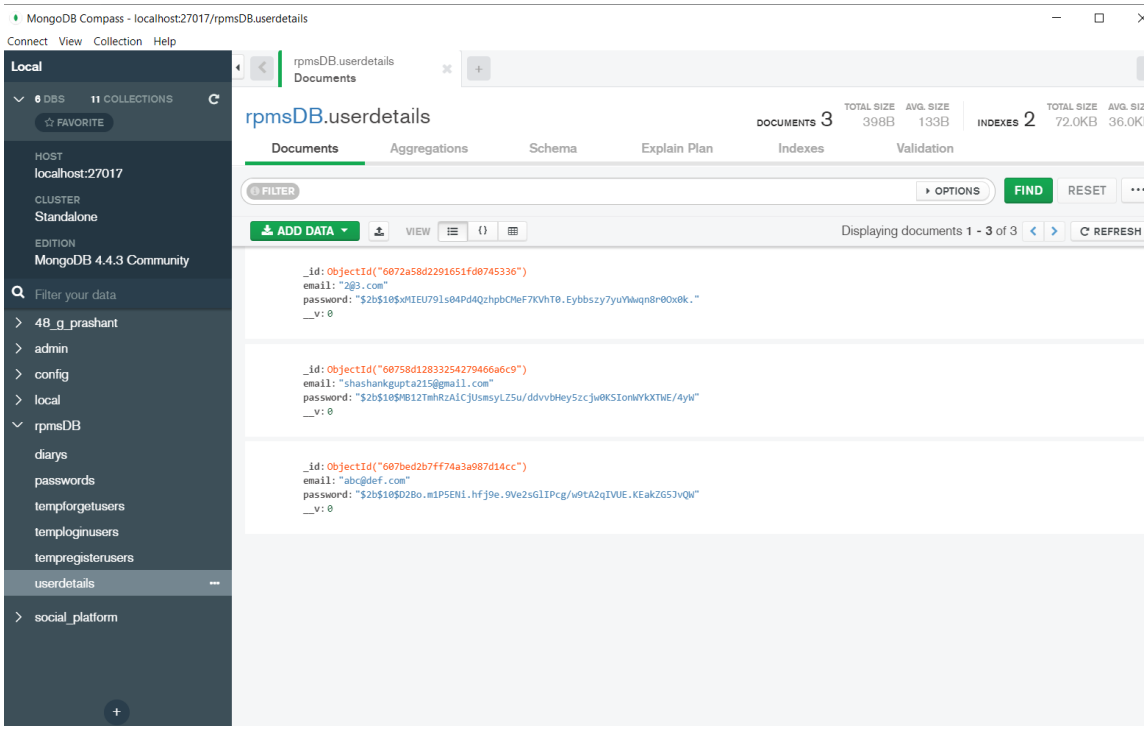


Fig. 3

3.3.2 Database



Img. 1



Img.2 (Database)

Chapter 4. Software Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under testing. Software testing is a process of executing a program or application with intent of finding the software bugs. It can also be stated as the process of validating and verifying that a software program or application.

Various test cases are as follows:

4.1 Testing of login session:

Sr. No	Test Cases	Feature	Description	Steps to Execute	Test Data/Input
1.	TC-1	User Interface	Check all the test boxes, radio buttons, buttons etc.	1.Click on Radio buttons, buttons and dropdown list	
2.	TC-2	Required Fields	Check the required fields by not filling any data	1.Do not enter any value in the fields 2.Click on Signup button	NIL
3.	TC-3	Required Fields	Check required fields by filling the data	1.Enter the valid values in required fields 2.Click on Signup button	NIL

Table No.-1

4.2 Testing of main session:

Test ID	Testing Method	Page	Purpose	Input	Expected Output	Actual Output	Result
T1	Equivalence Class Testing	Registration Page	To check name	A-Z and a-z	Error message	Error Message	Pass
T2	Equivalence Class Testing	Registration Page	To check name	1-9 or @, $, etc.	Error message	No Error message	Fail
T3	Equivalence Class Testing	Registration Page	To check Mobile number	10 digits	Error message	Error message	Pass
T4	Equivalence Class Testing	Registration Page	To check mobile number	11 digits	No Error message	No Error message	Fail
T5	Equivalence Class Testing	Registration Page	To check City	A-Z or a-z	Error message	Error message	Pass
T6	Equivalence Class Testing	Registration Page	To check State	A-z or a-z	Error message	Error message	Pass
T7	Equivalence Class Testing	Registration Page	To check Matric%	0-9	Error message	Error Message	Pass
T8	Equivalence Class Testing	Registration Page	To check 10+2%	0-9	Error Message	Error message	Pass

T9	**Equivalence Class Testing**	**Registration Page**	**To check Email**	**a-z+0-9+@+a-z+. +a-z**	**Error message**	**Error message**	**Pass**
T1	**Equivalence Class Testing**	**Registration Page**	**To check Institute**	**A-Z or a-z**	**Error message**	**Error message**	**Pass**
T1	**Equivalence Class Testing**	**Login Page**	**To check Username**	**a-z+0-9+@+a-z+. +a-z**	**Error message**	**Error message**	**Pass**
T1	**Equivalence Class Testing**	**Login page**	**To check**	**0-9**	**Error message**	**Error message**	**Pass**

Table No.-2

Chapter 5. Implementation and User Interface

5.1 This is the Home page of the project which shows a menu bar showing various buttons and a screen to show the disease as output.

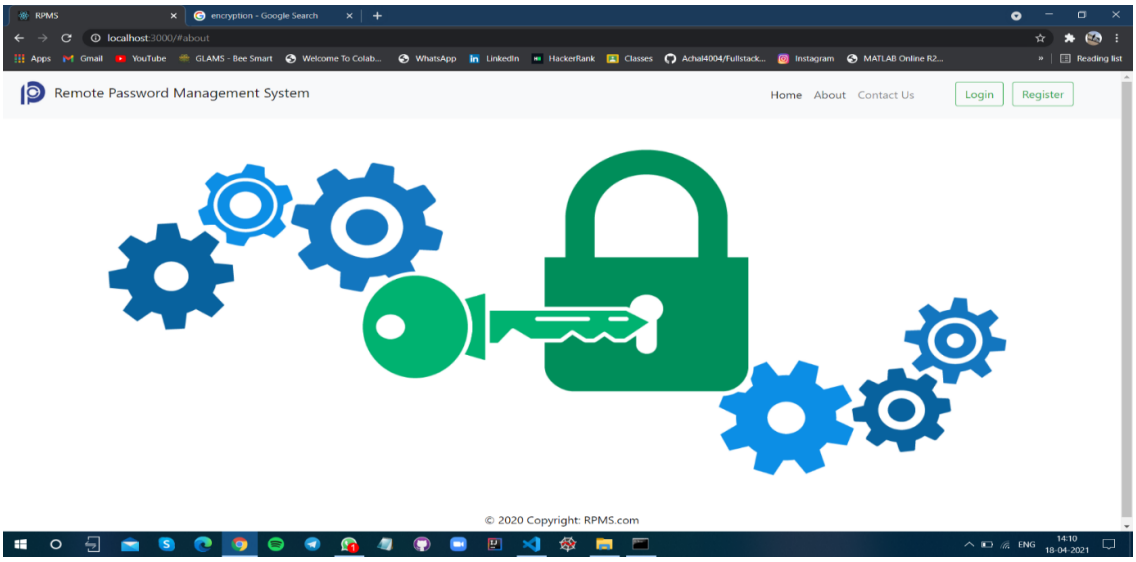


Fig. 3

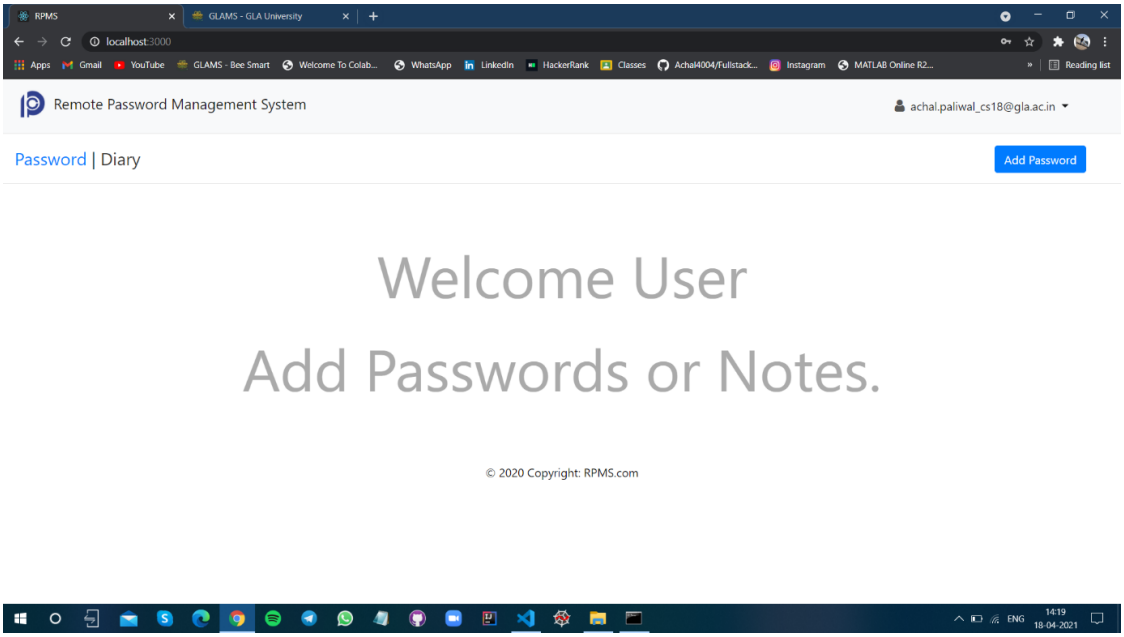


Fig. 4

5.2 - Registration Page

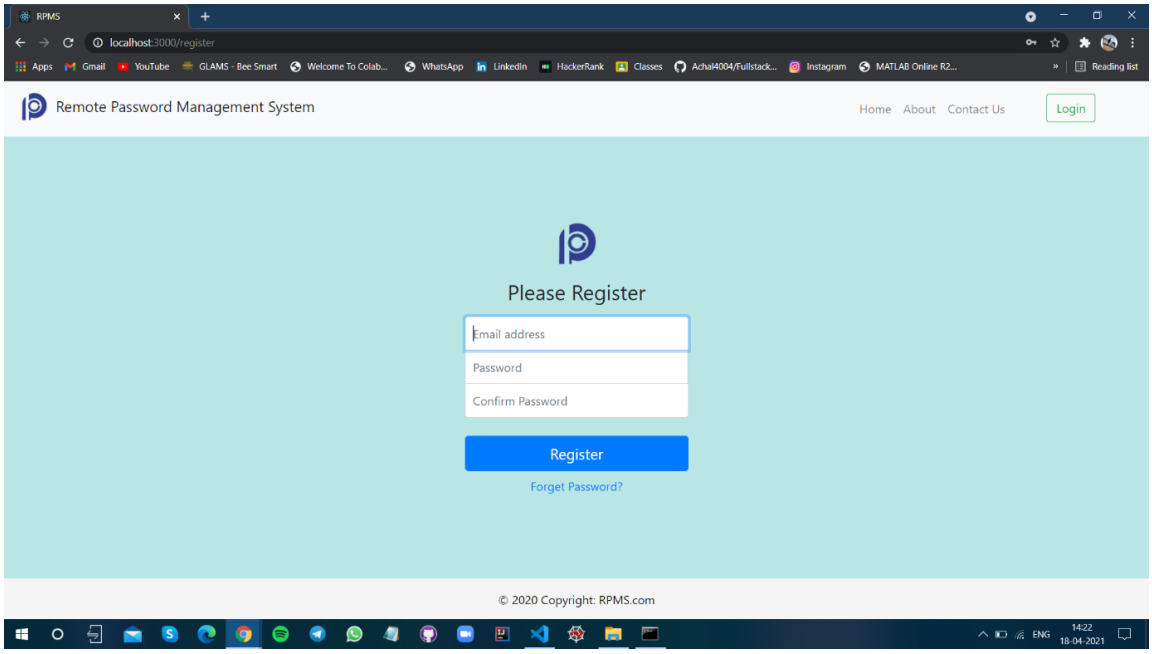


Fig. 5

5.2 - Login Page

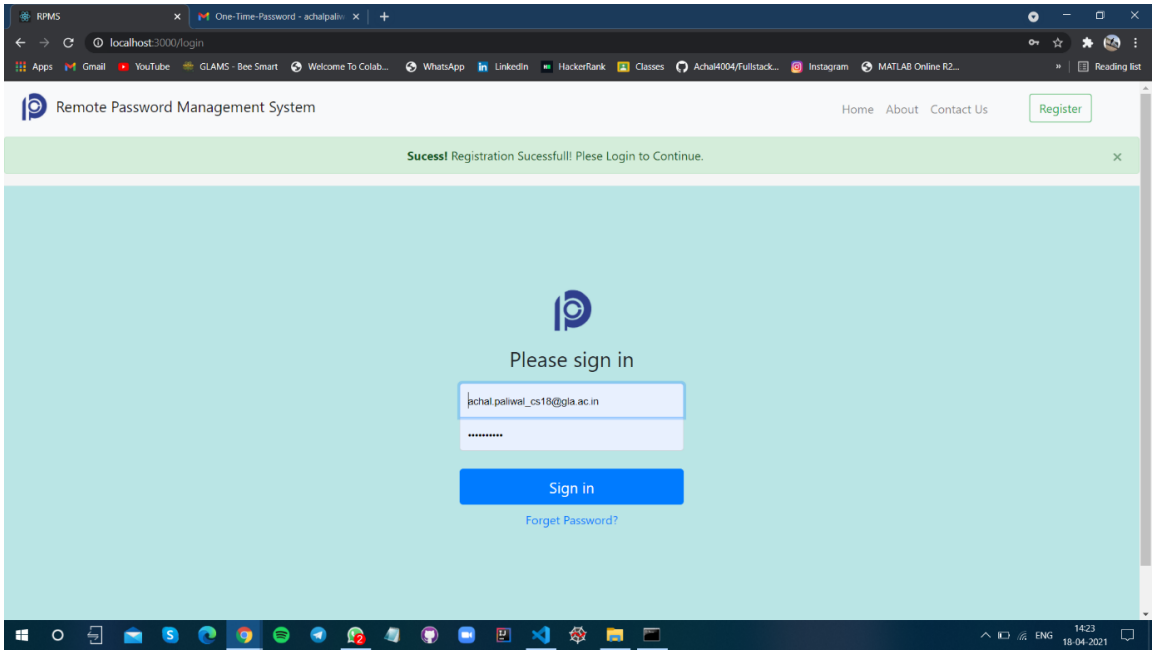


Fig. 6

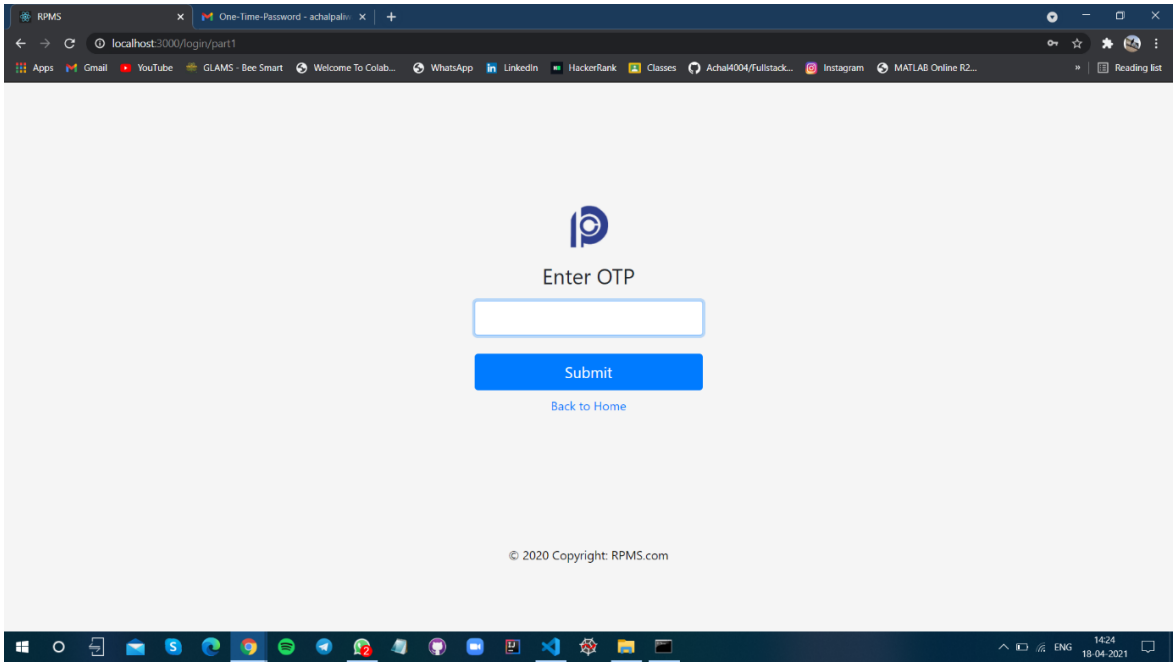


Fig. 7

5.4- Detail Pages

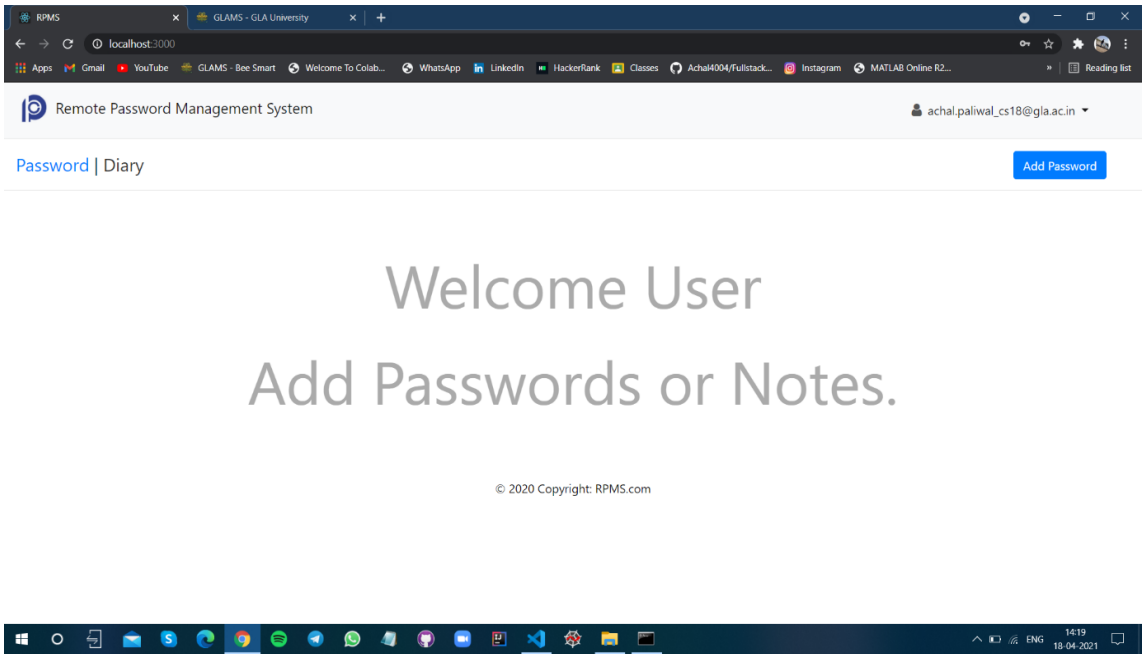


Fig. 8

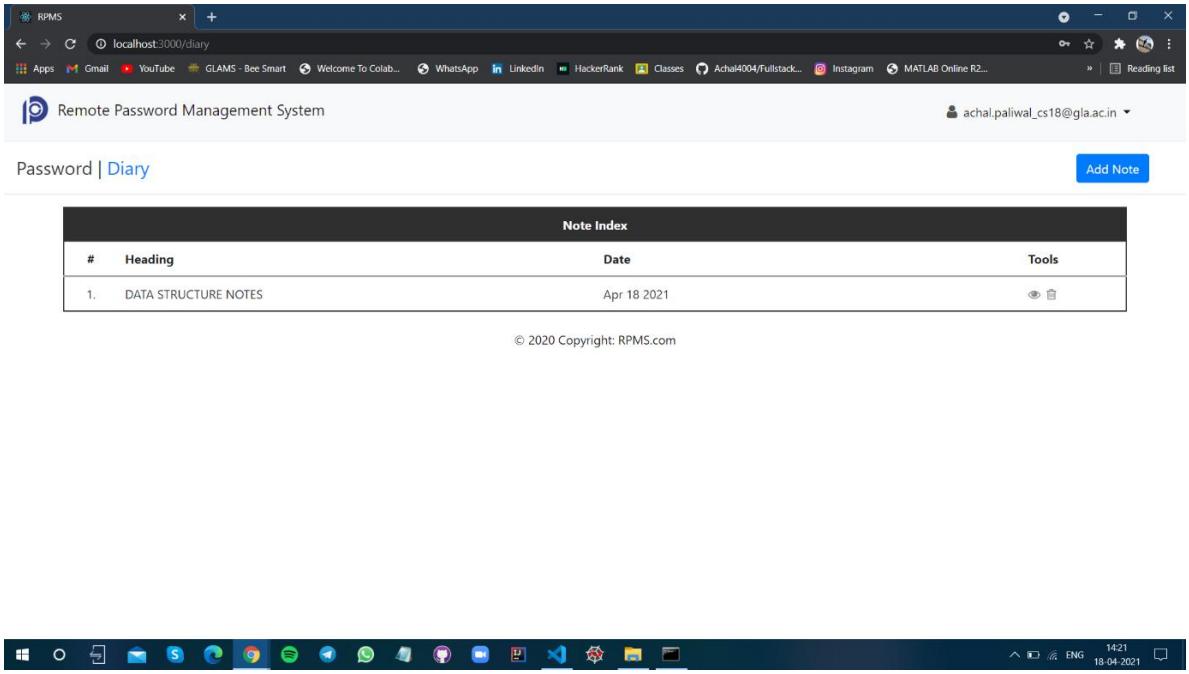

RPMS
localhost:3000/diary
Remote Password Management System
achal.paliwal_cs18@gla.ac.in
Password | Diary
Add Note
Note Index
Heading Date Tools
1. DATA STRUCTURE NOTES Apr 18 2021
© 2020 Copyright: RPMS.com

Fig. 9

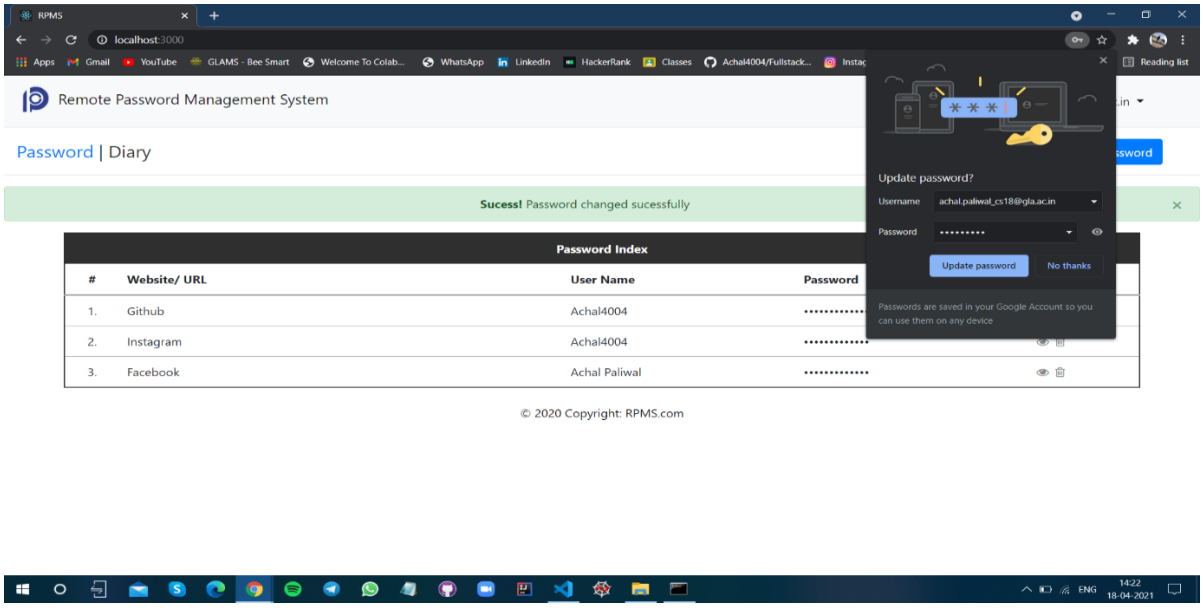

RPMS
localhost:3000
Remote Password Management System
Password | Diary
Sucess! Password changed sucessfully
Update password?
Username achal.paliwal_cs18@gla.ac.in
Password
Update password
No thanks
Passwords are saved in your Google Account so you can use them on any device
Password Index
Website/ URL User Name Password
1. Github Achal4004
2. Instagram Achal4004
3. Facebook Achal Paliwal
© 2020 Copyright: RPMS.com

Fig. 10

5.5- Forget password

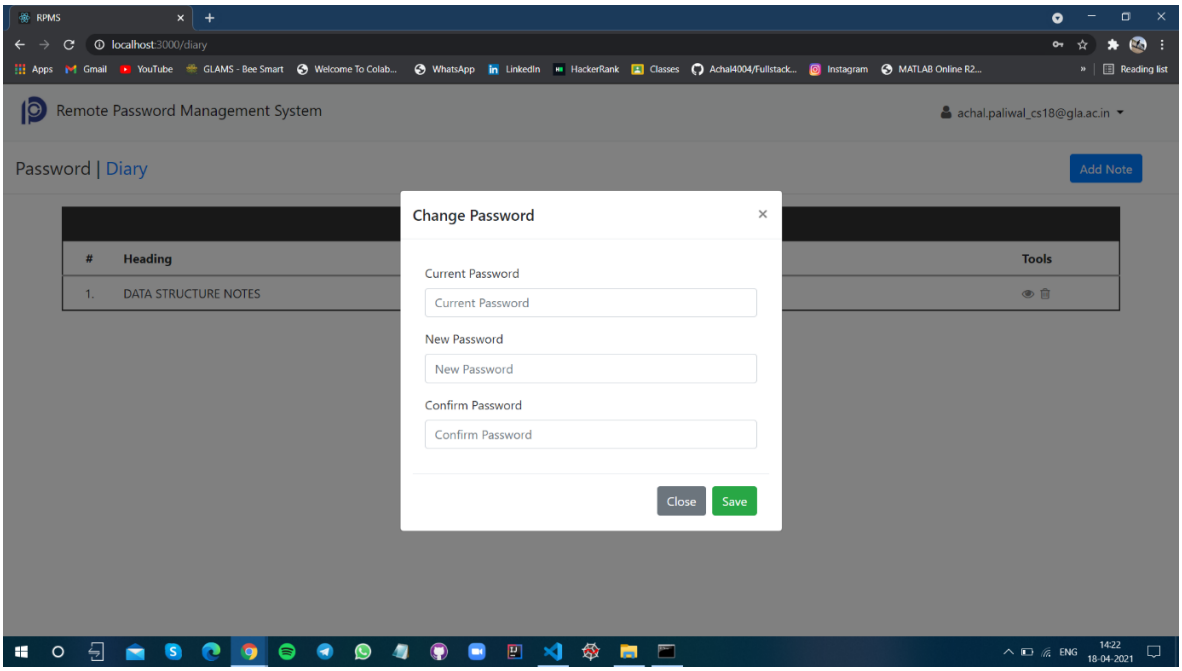


Fig.-11

Chapter 6 Appendices

Implementation /Coding Part of project:

```
require("dotenv").config()
const express = require("express");
const mongoose = require('mongoose');
const session = require('express-session');
const passport = require("passport");
const cookieParser=require("cookie-parser");
const flash=require("connect-flash");
const homeRoute=require('./routes/homeRoute');
const registerRoute=require('./routes/registerRoute');
const loginRoute=require('./routes/loginRoute');
const forgetRoute=require('./routes/forgetRoute');

const app=express();

app.set("view engine","ejs");
app.use(express.urlencoded({extended:true}));
app.use(express.static("public"));
app.use(cookieParser(process.env.SESSION_COOKIES_SECRET_KEY));
app.use(session({ secret: process.env.SESSION_COOKIES_SECRET_KEY,resave: false,saveUni
nitialized: false,}));
app.use(passport.initialize());
app.use(passport.session());
app.use(flash());

app.use(function(req,res,next){
    res.locals.sorry_msg=req.flash("sorry_msg");
    res.locals.success_msg=req.flash('success_msg');
    res.locals.error_msg=req.flash('error_msg');
    res.locals.error=req.flash('error');
    next();
});

mongoose
.connect(process.env.MONGODB_URIS, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useFindAndModify: false,
  useCreateIndex: true,
})
.then(() => console.log("MongoDB Connection Successfull...!!"))
.catch(() => console.log("Error, Not Connected with MongoDB"));
```

```
require('./controller/passport')(passport);
app.use('/',homeRoute);
app.use('/register',registerRoute);
app.use('/login',loginRoute);
app.use('/forget',forgetRoute);

app.all('*',(req,res)=> {
    res.render('error404');
});

const PORT = process.env.PORT || 3000
app.listen(PORT,function() {
    console.log("server is listening at port "+PORT);
})
```

Chapter 7 Progress

Part 1 is completed

PART 1: Registration Page and Login Page

- Designing Registration Page for user.
- Designing Login Page for User.

Part 2 is completed

PART 2: Email Verification

- The email for otp verification on login time will be sent to the user

Part 3 is completed

PART 3: Internal Dashboard and Interface

- Designing Page to add password and notes.
- Designing page to change and see these password and notes.

Part 4 is completed.

PART 4: Selection of Encryption technique

- Experiment with various algorithms.
- AES (Advanced Encryption Standard) is used.

Part 5 is completed.

PART 5: Testing

- Testing model to measure performance

Bibliography/ References

The following references were used in this project:

1. https: www.beta-labs.org
2. https://www.geeksforgeeks.org
3. https://www.Youtube.com
4. https://www.wikipedia.org
5. https://www.educative.io/