SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

A GRAPHICAL PASSWORD AUTHENTICATION SYSTEM

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

BACHELOR OF COMPUTER ENGINEERING

BY

Sanket Adhav	B190494203
Vikas Bankar	B190494210
Abhishek Chavan	B190494216
Tushar Temgire	B190494275

Under The Guidance of

Prof.Pramode Dhamdhere

DEPARTMENT OF COMPUTER ENGINEERING
Parvatibai Genba Moze College of Engineering
WAGHOLI PUNE -412207

2022-23

Parvatibai Genba Moze College of Engineering

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

A GRAPHICAL PASSWORD AUTHENTICATION SYSTEM

Submitted by

Sanket Adhav	B190494203
Vikas Bankar	B190494210
Abhishek Chavan	B190494216
Tushar Temgire	B190494275

is a bonafide work carried out by Students under the supervision of Prof. and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering) Project.

Prof.Pramod Dhamdhere
Internal Guide/Co-ordinator
Dept. of Computer Engg.

Prof. Shrikant Dhamdhere
H.O.D
Dept. of Computer Engg.

.....
External Examinar

Dr. Navnath Narawade
Principal
PGMCOE Wagholi

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***'A GRAPHICAL PASSWORD AUTHENTICATION SYSTEM'****.*

We would like to take this opportunity to thank our internal guide ***Prof.Pramod Dhamdhere*** *for giving us all the help and guidance we needed. We are really grateful for her kind support. Their valuable suggestions were very helpful.*

We are also grateful to ***Prof. Shrikant Dhamdhere****, Head of Computer Engineering Department, PGMOZECOE, for her indispensable support, suggestions.*

Sanket Adhav
Vikas Bankar
Abhishek Chavan
Tushar Temgire
(B.E. Computer Engg.)

Abstract

Authentication is the method of giving persons access to system object based on user's uniqueness. If the code match, the process will be accomplished and user will get the approval to access the system. Text-based password scheme follows the guidelines such as at least 8 characters long, should combine upper case and lower-case and digits. User have problem to remember their complicated password over time due to the limitation of human brain, user tend to forget about their password. User tend to use the same password for all type of account. So, if one account is hacked, the possibility for other account to be hack is high. Other than that, choosing the simple textual-based password may increase its vulnerability for attacks or intrusions. Hence, graphical password authentication by using passpoints scheme has been introduced in this project. Graphical password authentication by using passpoints scheme is a model to identify the most likely regions for user to click in order to create graphical password. The operation of the purposed scheme is simple and easy to learn for user since they familiar with textual graphical password scheme. In conclusion, this graphical password scheme will make it easier for user to do their authentication process since it is easy to remember and hard to guess by others.

Keywords

- Data mining
- Performance analysis

INDEX

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Cyber Security protects computer systems, back-end systems, and end-user applications, as well as the data they hold, in the same way, physical security protects physical property and persons from criminal activity or accidental harm. Its goal is to keep cybercriminals, malicious insiders, and others from gaining access to, hurting, disrupting, or changing IT systems and applications.

Alphanumeric passwords are a dated, conventional, and popular type of authentication. Practically speaking, the typical approach is an unsafe system. For instance, if a user is not using a strong password, the attacker may pick an easily guessable password. For numerous devices or websites, a user may use the same password. All of these characteristics make regular users vulnerable. And one of the key security points where the user actively assumes responsibility for the security of their personal information is during authentication. When using an outdated, conventional password scheme, dictionary and brute force attacks are possibilities.

1.2 MOTIVATION OF THE PROJECT

Because usability decreases as password strength grows, the text-based solution cannot accomplish the goal. It ensures that the system's usability and security are both preserved without requiring us to give up either of these requirements.

1.3 PROBLEM STATEMENT

A graphical password authentication system

1.3.1 Organization of the report

- Chapter 1: In this chapter, introduction of the project i.e. what is need, relevance and what is actual project idea.
- Chapter 2: In this chapter, we briefly review the related work on mental disorder detection and their different techniques.

- Chapter 3: System Design describes proposed system approach and their advantages, System design diagrams.
- Chapter 4: in this chapter, conclusion of the project, Future Scope of the project, Applications and Limitations of the project.

1.4 PROJECT SCOPE

- This is system used for security purpose.

CHAPTER 2

LITERATURE SURVEY

William Stallings and Lawrie Brown., “Computer Security: Principle and Practices.”[1]. Interest in education in computer security and related topics has been growing at a dramatic rate in recent years. This interest has been spurred by a number of factors, two of which stand out: 1. As information systems, databases, and Internet-based distributed systems and communication have become pervasive in the commercial world, coupled with the increased intensity and sophistication of security-related attacks, organizations now recognize the need for a comprehensive security strategy. This strategy encompasses the use of specialized hardware and software and trained personnel to meet that need.

Susan Wiedenbecka,, Jim Watersa , Jean-Camille Birgetb , Alex Brodskiyc , Nasir Memon, “PassPoints: Design and longitudinal evaluation of a graphical password system”, [2] Finally, the actual evaluation of PassPoints reveals its advantages and disadvantages. Users using graphical passwords could make a good password fast and easily, but it took them longer and required more tries to memorise their passwords than users of alphabetic passwords. Over the course of six weeks, graphical users and alphanumeric users both retained their passwords, however graphical users continued to take longer to type their passwords. When it came to how simple, quick, and enjoyable their password system was, graphical users had perceptions that were comparable to those of alphanumeric users.

Robert Morris and Ken Thompson., “Password security: a case history”, [3] UNIX is arguably more secure than most systems when it comes to password security. In the absence of careful consideration from subject-matter specialists, using encrypted passwords seems to be adequately secure.

Making an attempt to hide even encrypted passwords is worthwhile. An ”external security code” is a requirement on some UNIX systems that users must enter when phoning in but before logging in. If this code is updated on a regular basis, someone with an outdated password will probably be unable to access it. Daniel V. Klein, “Foiling the Cracker: A Survey of, and Improvements to, Password Security ”, [4] Good fences make good neighbours,” as the saying goes. On a Unix system, many users also claim that they ”don’t need a solid password” since they ”don’t care who sees my files.” Unfortunately, keeping data unsecured is not the same as leaving accounts open to attack. The data stored in the unprotected files is all that is at risk in the latter scenario,

but the entire system is at risk in the former. Your home's front entrance is an invitation to the regrettably commonplace low-morale individuals, even if you only secure it with a cheap latch. The same is true for accounts that are susceptible to password cracking attacks.

Eugene H. Spafford., "Observing reusable password choices", [5] In this essay, the design of a password collector was presented. The collector has provided some interesting design issues despite being created to facilitate investigation of a new password screening technique. To securely save obtained passwords for future analysis, the collector employs a public-key technique. The instrumented systems are not under any apparent danger during the collection operation. The method employed could be adapted to other contexts and readily expanded to gather additional data.

Sigmund N. Porter. "A password extension for improvedhuman factors",[6] We utilise a reasonably big keyspace (64 bits) and a very long "passphrase" (up to 80 characters) to enhance both the difficulty of guessing passwords and also the simplicity of remembering passwords. The word is entered into the key and hashed, and it is then saved in encrypted form. One-way encryption is a necessary component of the hashing. Given the phrase's length, one would anticipate both the hashed and the original phrase to have a sizable keyspace. Since the owner finds meaning in the term, it ought should be simpler to remember.

XiaoyuanSuo, Ying Zhu, and G. Scott Owen,"Graphical passwords: A survey",[7]The use of graphical passwords as an alternative to conventional text-based passwords has gained popularity during the past ten years. We have undertaken a thorough analysis of the graphical password approaches that are currently in use in this work. The two types of graphical password methods currently in use are recognition-based and recall-based methods.

Antonella De Angeli, Lynne Coventry, Graham Johnson, and Karen Renaud,"Is a picture really worth athousand words? exploring the feasibility of graphicalauthentication systems",[8]In order to utilise visual memory for user authentication in self-service technologies, this research revealed a user-centered approach to the creation of cognometric mechanisms. Our experience has shown that designing an effective authentication method is a challenging process since it necessitates taking into account and balancing a number of crucial factors in order to achieve the

highest levels of security and usability. No ”miracle solution” exists, and the tension between these two goals occasionally seems insurmountable. This paper’s contribution is the definition of a few variables that may impact the usability and security of graphical authentication techniques.

Siva Janakiraman, Karunya Sri V S, ChathuryaPulluri, SundararamanRajagopalan, K Thenmozhi, and RengarajanAmirtharajan, “Numerical Password via Graphical Input – An Authentication System on Embedded Platform”,[9] The system suggested in this paper enables the user to use graphical passwords while storing and authenticating them using hexadecimal numerical passwords. It is not necessary to store the pixel values for the selected block when LFSR is used to randomise the image matrix blocks during the authentication phase. In this strategy, increasing the picture matrix’s dimension can increase the number of password combinations that are conceivable. As a result, the suggested technique works well for highly secure graphical password authentication for embedded devices with constrained memory resources.

Sung-Shiou Shen, Tsai-Hua Kang, Shen-Ho Lin, Wei Chien, “Random Graphic User Password Authentication Scheme in Mobile Devices”[10] the self-developing keypad lock app’s graphical user interface, which launches when the user tries to unlock the screen. To start, the placement of each digital button on the screen is chosen using a random number generator method that generates random numbers. The user must think about the password sequence and the shortest route based on the location of each of the digital buttons on the screen, for instance, if the user password is set to ”168”. The app programme gives 1-2 redundant toleration digitals analysing mechanisms for the convenience of users, even if it is designed to determine the quickest path and password sequence. In other words, the sequence ”1968” is the precise graphic user password.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

3.1.1 PROJECT SCOPE

This is system used for security purpose.

3.2 ASSUMPTIONS AND DEPENDENCIES

A number of factors that affect the requirements of the system are:

- The system the application is executing on will have the required resources available as necessary
- Another assumption is that the software and hardware components work in the same way as used while developing this project.

3.3 FUNCTIONAL REQUIREMENTS

3.3.1 System Feature 1(Functional Requirement).

1. Graphical Password Authentication system

3.3.2 User Interfaces

Home page
Login page
Generate Graphical Password
Verify Password

3.3.3 Hardware Interfaces

The entire software requires a completely equipped computer system including monitor, key-board, and other input output devices.

3.3.4 Software Interfaces

The system can use Microsoft as the operating system platform. System also makes use of certain GUI tools. To run this application we need java. To store data we need MySQL database.

3.4 NONFUNCTIONAL REQUIREMENTS

3.4.1 Performance Requirements

The performance of the system lies in the way it is handled. Every user must be given proper guidance regarding how to use the system. The other factor which affects the performance is the absence of any of the suggested requirements.

3.4.2 Safety Requirements

To ensure the safety of the system, perform regular monitoring of the system so as to trace the proper working of the system. An authenticated user is only able to access system.

3.4.3 Security Requirements

Any unauthorized user should be prevented from accessing the system. Password authentication can be introduced.

3.4.4 Software Quality Attributes

Accuracy: -

The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate. Result is organic results.

Reliability: -

The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information and Recommending location model.

3.5 SYSTEM REQUIREMENTS

3.5.1 Database Requirements

Dataset

3.5.2 Software Requirements(Platform Choice)

• Operating System - Windows

• Application Server - Apache Tomcat

• Front End - HTML, Bootstarp,CSS

• Language - Java.

• Database - My SQL

• IDE - Eclipse

3.5.3 Hardware Requirements

• Processor - Intel i3/i5/i7

• Speed - 3.1 GHz

• RAM - 4 GB(min)

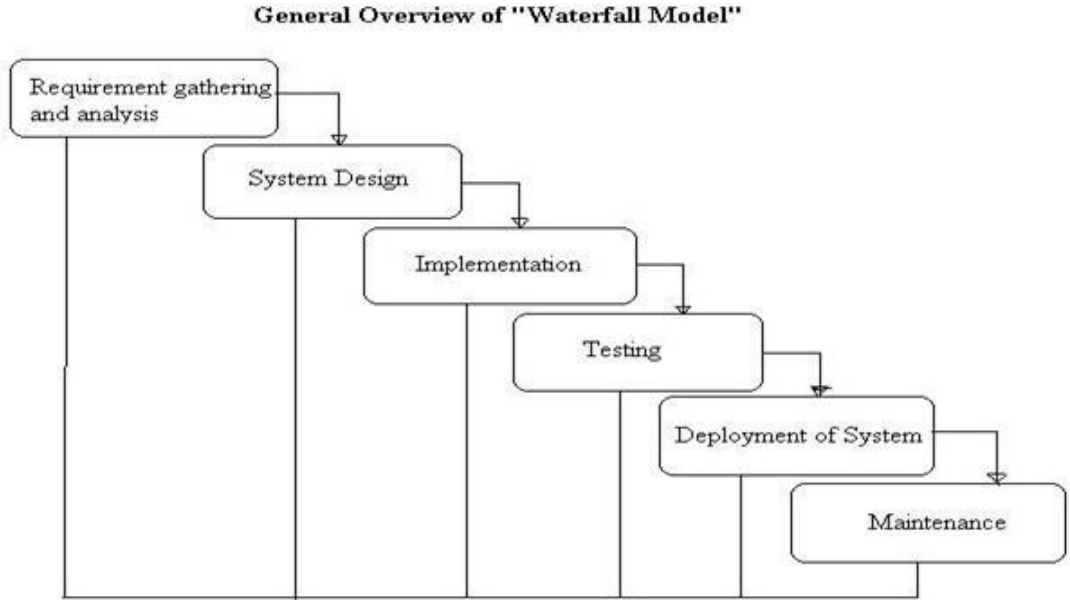
• Hard Disk - 20 GB

• Key Board - Standard Windows Keyboard

• Mouse - Two or Three Button Mouse

• Monitor - SVGA

3.6 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED



The Waterfall Model is sequential design process, often used in Software development processes, where progress is seen as flowing steadily download through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software developments. This one of the oldest model followed in software engineering. The process begins with the communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling, construction and deployment of the software.

There are 5 Phase of water fall model:

The Waterfall Model is sequential design process, often used in Software development processes, where progress is seen as flowing steadily download through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software developments. This one of the oldest model followed in software engineering. The process begins with the communication phase where the customer specifies the requirements and then progress through other phases like planning, modeling, construction and deployment of

the software.

There are 5 Phase of water fall model:

1. COMMUNICATION

In communication phase the major task performed is requirement gathering which helps in finding out exact need of customer. Once all the needs of the customer are gathered the next step is planning.

2. PLANNING

In planning major activities like planning for schedule, keeping tracks on the processes and the estimation related to the project are done. Planning is even used to find the types of risks involved throughout the projects. Planning describes how technical tasks are going to take place and what resources are needed and how to use them.

3. MODELING

This is one the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed. Different models for developing software are created depending on the requirements gathered in the first phase and the planning done in the second phase.

4. CONSTRUCTION

The actual coding of the software is done in this phase. This coding is done on the basis of the model designed in the modeling phase. So in this phase software is actually developed and tested.

5. DEPLOYMENT

In this last phase the product is actually rolled out or delivered installed at customer's end and support is given if required. A feedback is taken from the customer to ensure the quality of the product. From the last two decades Waterfall model has come under lot of criticism due to its efficiency issues. So let's discuss the advantages and disadvantages of waterfall model.

3.7 SYSTEM IMPLEMENTATION PLAN

Activity	I Week	II Week	III week	IV week	V Week	VI week	VII week	VIII week	IX week
	Aug 4	Aug 11	Aug 18	Aug 25	Sept 1	Sept 8	Sept 15	Sept 22	Sept 29
Initiate the project									
Communication									
Literature survey									
Define scope									
Develop SRS									
Plan the project									
Design mathematical model									
Feasibility Analysis									
Develop work breakdown structure									
Planning project schedule									
Design UML and other diagrams									
Design test plan									
Design risk management plan									

Activity	XI week	XII week	XIII Week	XIV week	XV week	XVI week	XVII week	XVIII week	XIX week	XX week	XXI week	XXII week
	Jan 5	Jan 15	Jan 19	Jan 26	Feb 2	Feb 9	Feb 16	Feb 23	Mar 2	Mar 9	Mar 16	April 25
Execute the project												
Build and test basic functional unit												
Build and test database with login and session maintenance facility												
Designing of 1rd and 2th module												
Testing of 1st and 2nd module												
Designing of 3rd and 4th module												
Testing of 3rd and 4th module												
Integration of all module												
Final Report and Presentation												

Table 3.1: System Implementation Plan

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The system architecture diagram is a visual representation of the system architecture. It shows the connections between the various components of the system and indicates what functions each component performs. The general system representation shows the major functions of the system and the relationships between the various system components.

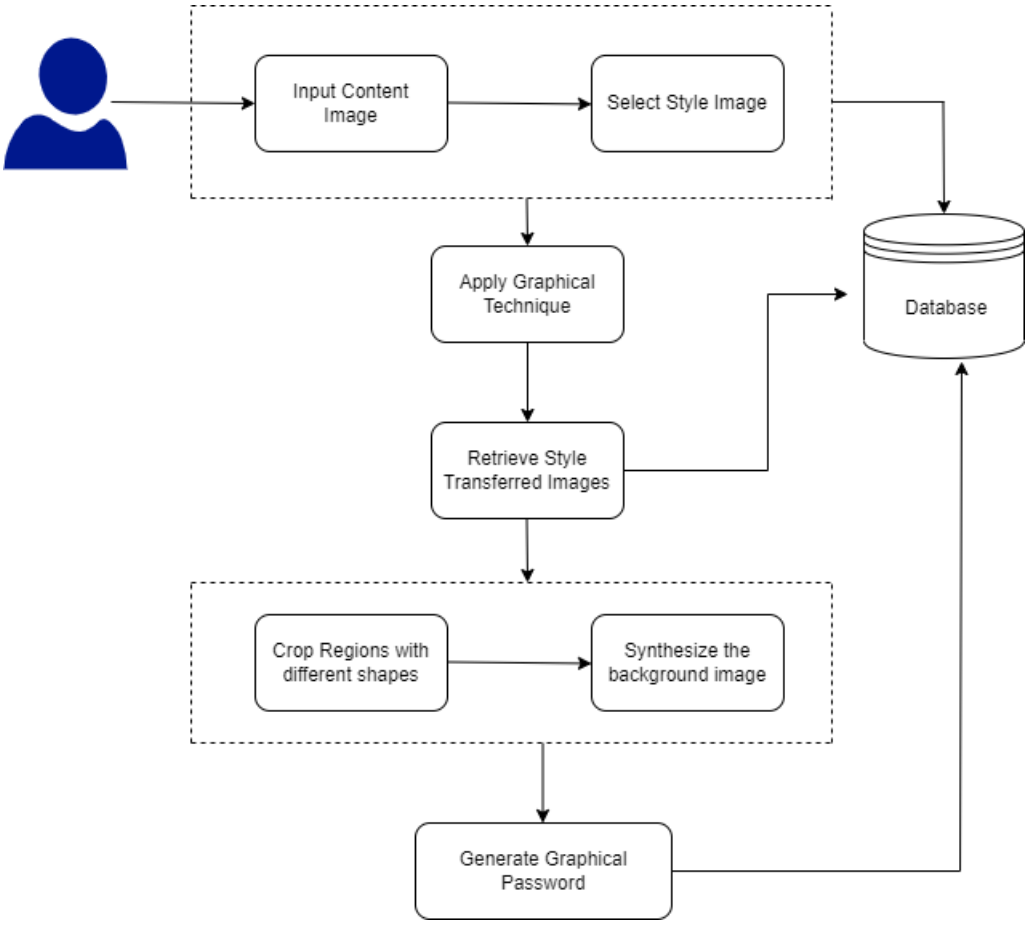


Figure 4.1: System Architecture

4.2 ALGORITHM

Algorithm1: Crop an Image(I,left,top,right,bottom)

Input: image I, rectangle with corners (left,top) and (right-1,bottom-1)

Output: cropped image I' of size new-width×new-height

new-width←right-left

new-height←bottom-top

I'←AllocateImage(new-width,new-height)

for(xˆ',yˆ')Iˆ' do

I^' (x^',y^')←I(x^'+left,y^'+top)
return I'

4.3 DATA FLOW DIAGRAMS:

4.3.1 Level 0 data flow diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information ows in the system.

3. Figure 4.1 shows level 0 DFD which shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information ow and the transformations that are applied as data moves from input to output.

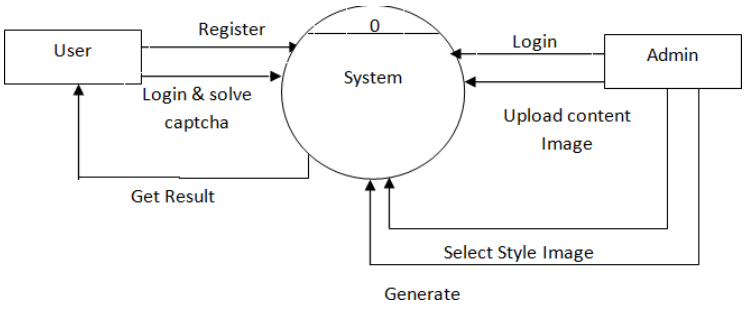


Figure 4.2: DFD Level 1

4.4 UML DIAGRAMS

4.4.1 Use case Diagram

A use case diagram is a graphical representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can show the different types of users of a system and the various ways in which they interact with the system. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionality use cases are prepared and actors are identified. The purposes of use case diagrams can be as follows:

• Used to gather requirements of a system.

• Used to get an outside view of a system.

• Identify external and internal factors influencing the system.

• Show the interaction among the actors.

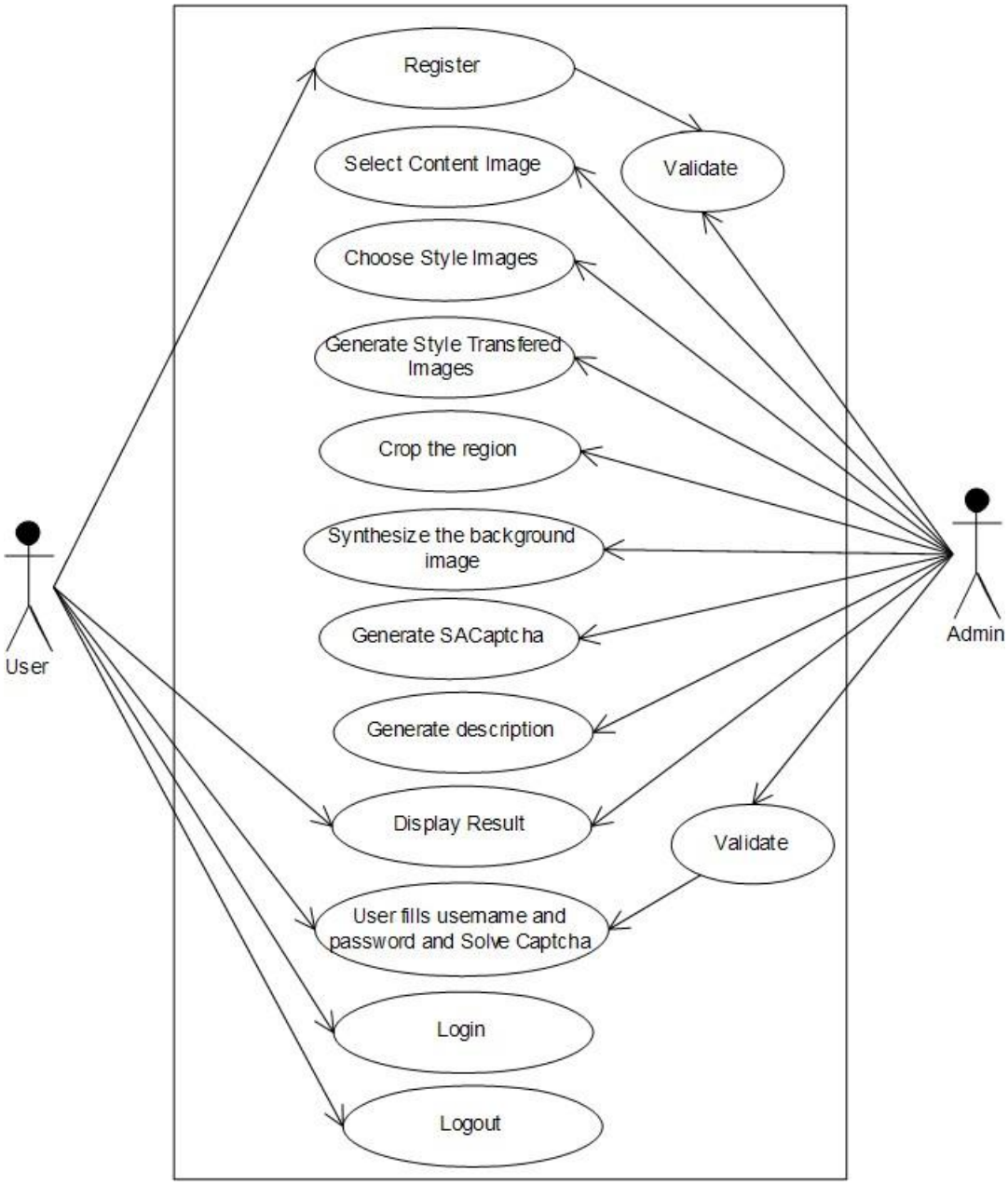


Figure 4.3: USE CASE

4.4.2 Component Diagram

A Component Diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

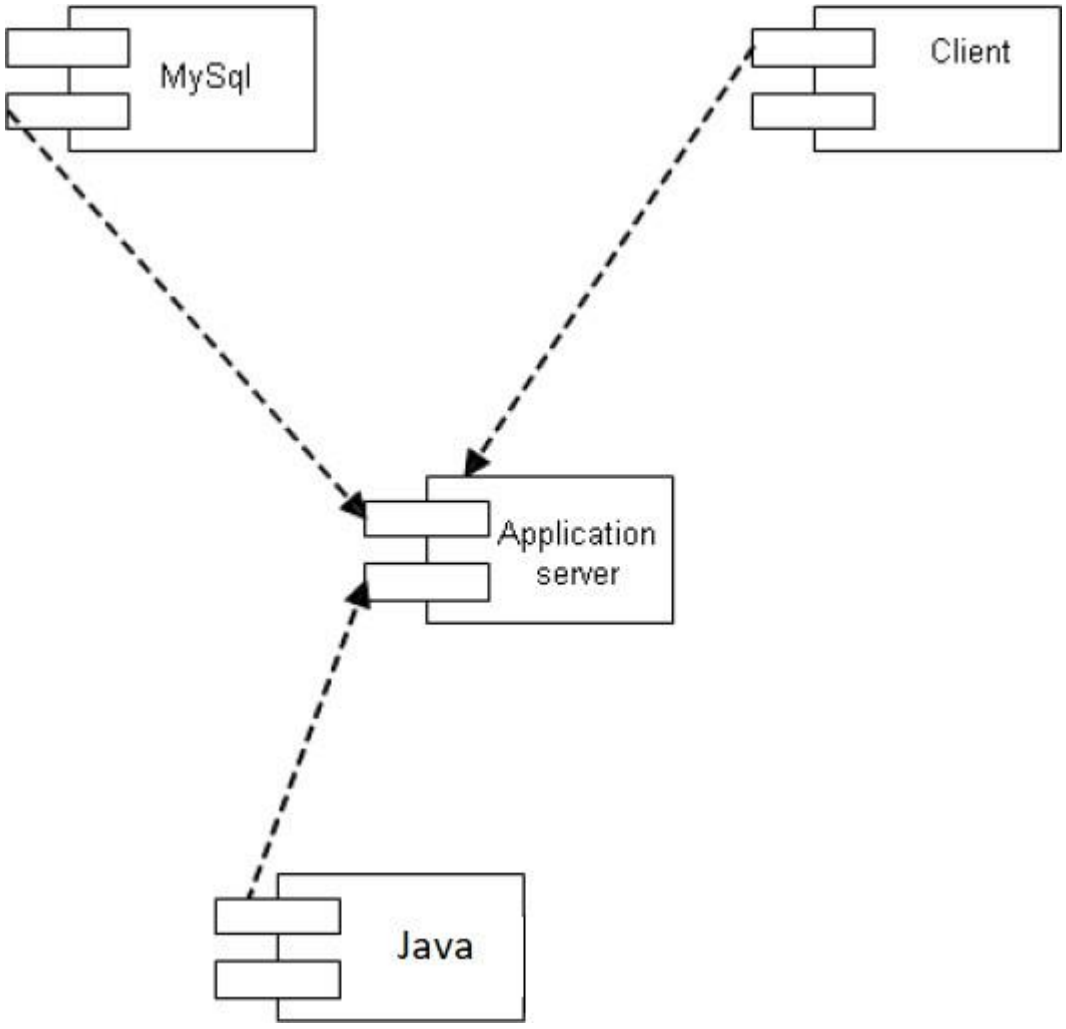


Figure 4.4: Component Diagram

4.4.3 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

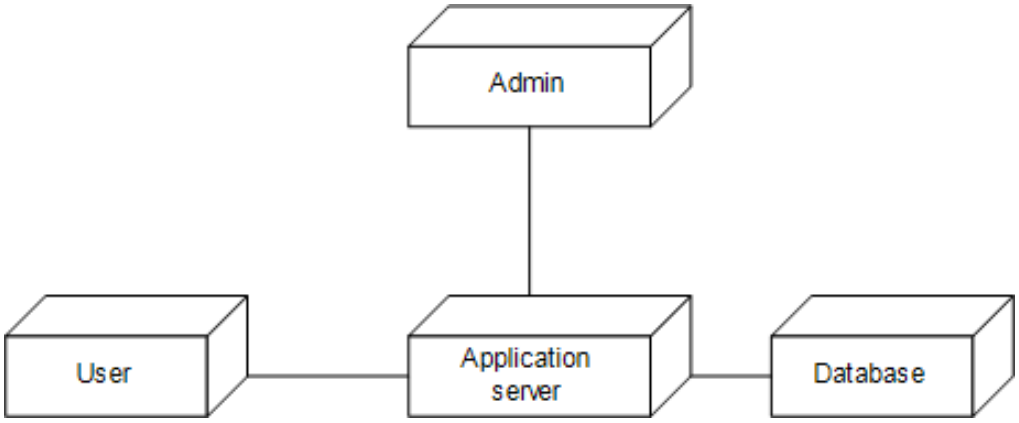


Figure 4.5: Deployment Diagram

CHAPTER 5

PROJECT PLAN

5.1 PROJECT ESTIMATES

5.1.1 Reconciled Estimates

5.1.1.1 Cost Estimates

Cost will estimate after completing the project that depend on time to complete the project. Also efforts required to complete.

5.1.1.2 Time Estimates

Time will depend on modules of project.Also project plan of execution.

5.1.2 Project Resources

1. Hardware Resources Required
   System: Intel 2.4 GHz. Hard Disk: 40 GB. Monitor: 15 VGA Color.

2. Software Resources Required
   Operating system: Windows. Coding Language: Python Database: SQL IDE: VS Code

5.2 RISK MANAGEMENT

5.2.1 Risk Identification

For risks identification, review of scope document, requirements specifications and schedule is done. Answers to questionnaire revealed some risks. Each risk is categorized as per the categories mentioned in [**?**]. Please refer table **??** for all the risks. You can refereed following risk identification questionnaire.

1. Have top software and customer managers formally committed to support the project?
Answer: Yes , Have top software and customer managers formally committed to support the project

2. Are end-users enthusiastically committed to the project and the system/product to be built?
Yse, end-users enthusiastically committed to the project and the system/product to be built

3. Are requirements fully understood by the software engineering team and its customers?
Yes, Are requirements fully understood by the software engineering team and its customers

5. Have customers been involved fully in the definition of requirements?

Yes, customers been involved fully in the definition of requirements

6. Are project requirements stable?

Answer: all project requirements are stable

9. Is the number of people on the project team adequate to do the job?

Yse, the number of people on the project team adequate to do the job

5.2.2 Risk Analysis

DESCRIPTION	IOW	High
Login detail	no	yes
Internet connection slow	yes	no

Table 5.1: Risk Analysis

Following are the details for each risk.

Risk ID 1

Risk Description Description 1

1.While login to system validation is there so user must follow rules and propoer enteries

Risk ID 2

1.While registering to system internet connection should be their for registration to get users data online

5.3 PROJECT SCHEDULE

5.3.1 Project Task set

Major Tasks in the Project stages are:

- Task 1: Requirement Analysis (Base Paper Explanation).
- Task 2: Project Specification (Paper Work).
- Task 3: Technology Study and Design.
- Task 4: Coding and Implementation (Module Development).

5.3.2 Task Network

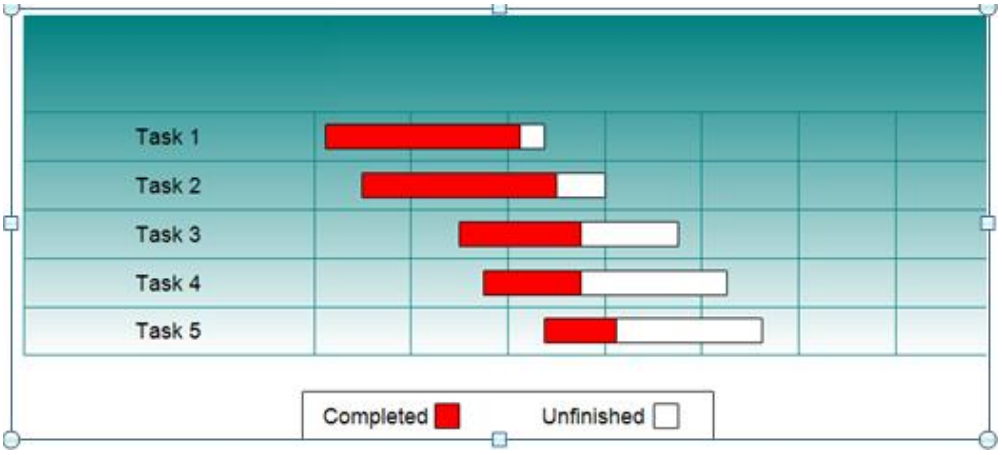


Figure 5.1: Task network

5.3.3 Timeline Chart

Activity	I week	II week	III week	IV week	V Week	VI week	VII week	VIII week	IX week
	Aug 4	Aug 11	Aug 18	Aug 25	Sept 1	Sept 8	Sept 15	Sept 22	Sept 29
Initiate the project									
Communication									
Literature survey									
Define scope									
Develop SRS									
Plan the project									
Design mathematical model									
Feasibility Analysis									
Develop work breakdown structure									
Planning project schedule									
Design UML and other diagrams									
Design test plan									
Design risk management plan									

Figure 5.2: Timeline Chart

5.3.4 Timeline Chart

Activity	XI week	XII week	XIII week	XIV week	XV week	XVI week	XVII week	XVIII week	XIX week	XX week	XXI week	XXII week
	Jan 5	Jan 15	Jan 19	Jan 26	Feb 2	Feb 9	Feb 16	Feb 23	Mar 2	Mar 9	Mar 16	April 25
Execute the project												
Build and test basic functional unit												
Build and test database with login and session maintenance facility												
Build and test Bluetooth mode												
Build and test security features												

Figure 5.3: Timeline Chart

5.4 TEAM ORGANIZATION

5.4.1 Team Structure

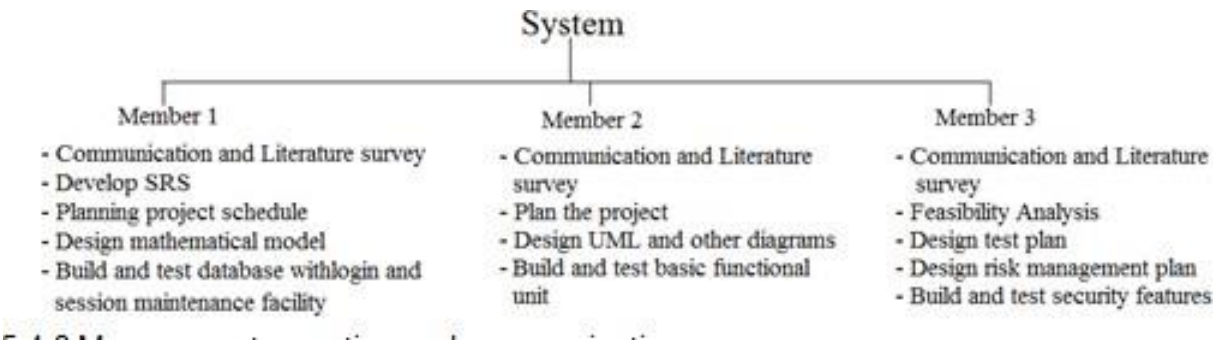


Figure 5.4: Team structure

5.4.2 Management reporting and communication

Sr No.	Month	Description
1	June	Discussion with guide regarding domain. Searching for IEEE paper for domain.
2	July	Short listing of IEEE papers within domain. Selection of IEEE paper.
3	August	Deciding Project name. Submission of Synopsis.
4	September	Requirement analysis. Designing of models.
5	October	Report preparation. Stage-I report submission.

Table 5.2: Management reporting and communication

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 TOOLS AND TECHNOLOGIES USED

6.1.1 Technology Description

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes the machine language of the Java Virtual Machine1 (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

6.1.2 Hardware Specification

- Processor - I3,I5
- Speed - 3.8 GHz
- RAM - 4GB
- Hard Disk - 1 TB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - LCD(Liquid Crystal Display)

6.1.3 Software Specifications

- Operating System: Windows 10
- Programming Language: Java
- Backend: Mysql 5.0
- IDE : Eclipse

CHAPTER 7

SOFTWARE TESTING

7.1 TYPES OF TESTINGS

7.1.1 Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.1.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.1.3 Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or

special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.1.4 System Testing

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

7.1.5 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

7.1.6 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.1.7 Unit Testing :

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

7.1.8 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task

of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

7.1.9 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2 TEST CASES AND TEST RESULTS

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Module-ID:-01

Modules to be tested:-Registration

1. Enter the case insensitive Username click on Submit button.

Expected: It should display error.

2. Enter the case sensitive Username click on Submit button.

Expected: It should accept.

3. Enter the case insensitive Password click on Submit button.

Expected: It should display error.

4. Enter the case sensitive Password click on Submit button.

Expected: It should accept.

5. Enter the case insensitive Mobile Number click on Submit button.

Expected: It should display error.

6. Enter the case sensitive Mobile Number click on Submit button.

Expected: It should accept.

7. Enter the wrong address and click on Submit button.

Expected: It should display error.

8. Enter the correct address and click on Submit button.

Expected: It should accept.

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
101	Enter the case insensitive Username click on Submit button.	Username	Error comes	Error Should come	P
102	Enter the case sensitive Username click on Submit button.	Username	Accept	Accept Username	P
201	Enter the case insensitive Password click on Submit button.	Password	Error comes	Error Should come	P
202	Enter the case sensitive Password click on Submit button	Password	Accept	Accept	P
301	Enter the case insensitive Mobile Number click on Submit button	Mobile Number	Error comes	Error Should come	P
302	Enter the case sensitive Mobile Number click on Submit button.	Mobile Number	Accept	Accept	P

Table 7.1: Test Cases

Module-ID:-2

Modules to be tested:- Login

1. Enter the correct username and wrong password click on Submit button.

Expected: It should display error.

2. Enter the wrong username and correct password and click on Submit button.

Expected: It should display error.

3. Enter the correct username and password and click on Login button.

Expected: It should display welcome page.

4. After login with valid credentials click on back button.

Expected: The page should be expired.

5. After login with valid credentials copy the URL and paste in another browser.

Expected: It should not display the user's welcome page.

6. Check the password with Lower case and upper case.

Expected: Password should be case sensitive.

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
001	Enter the correct username and wrong password click on Login button.	Username Password	Error comes	Error Should come	P
002	Enter the wrong username and correct password click on Login button,	Username Password	Error comes	Error Should come	P
003	Enter the correct username and password and click on Login button.	Username Password	Accept	Accept	P

Table 7.2: Test Cases

CHAPTER 8

RESULTS

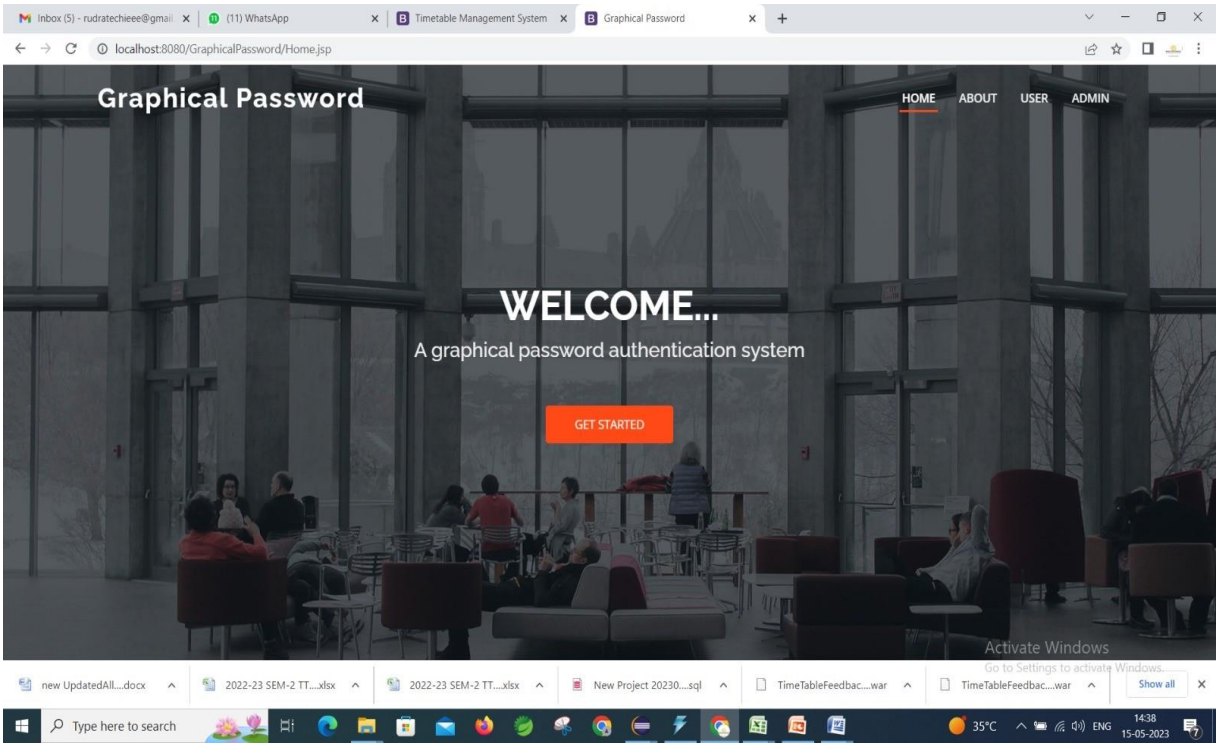


Figure 8.1: Welcome Page

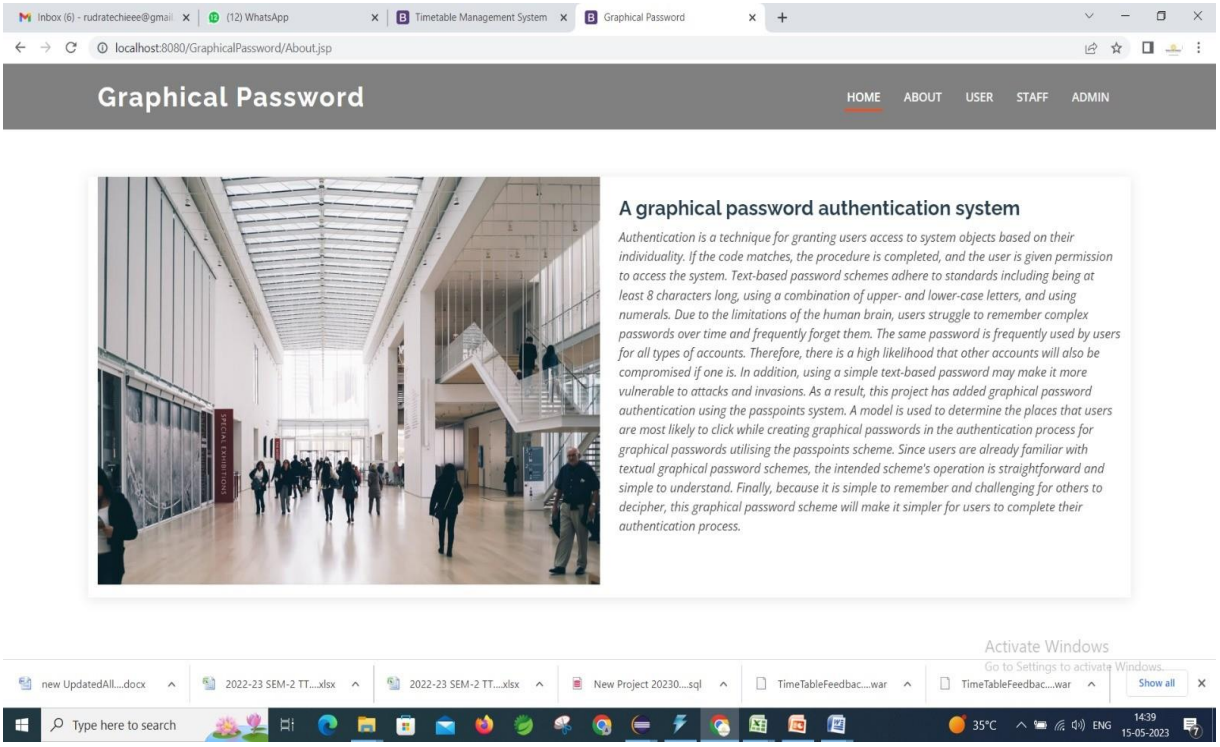


Figure 8.2: Home Page

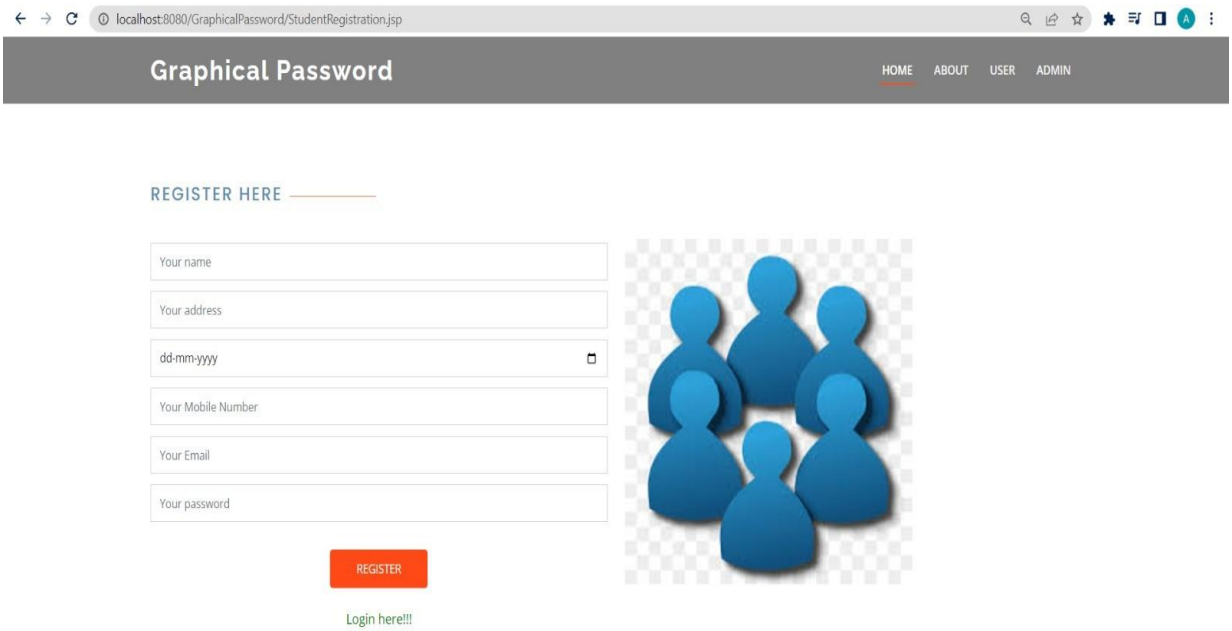


Figure 8.3: User Register Page

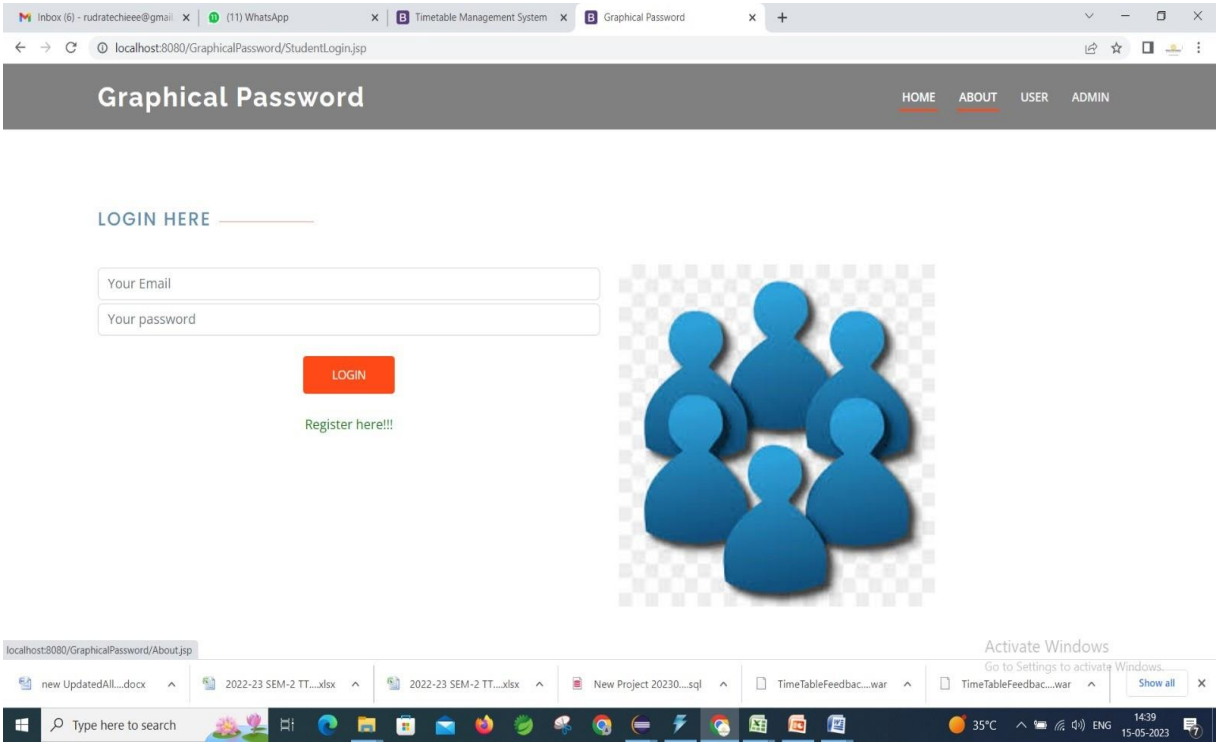


Figure 8.4: User Login Page

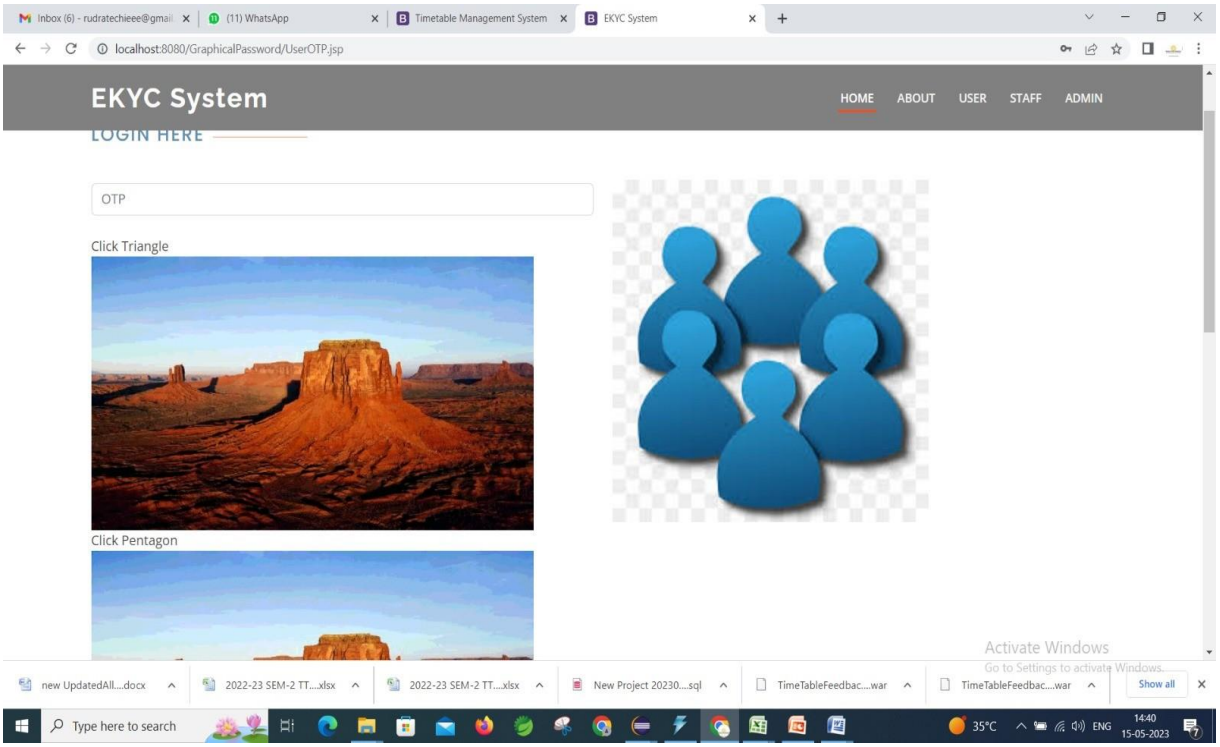


Figure 8.5: Login With Clude Click Point Page

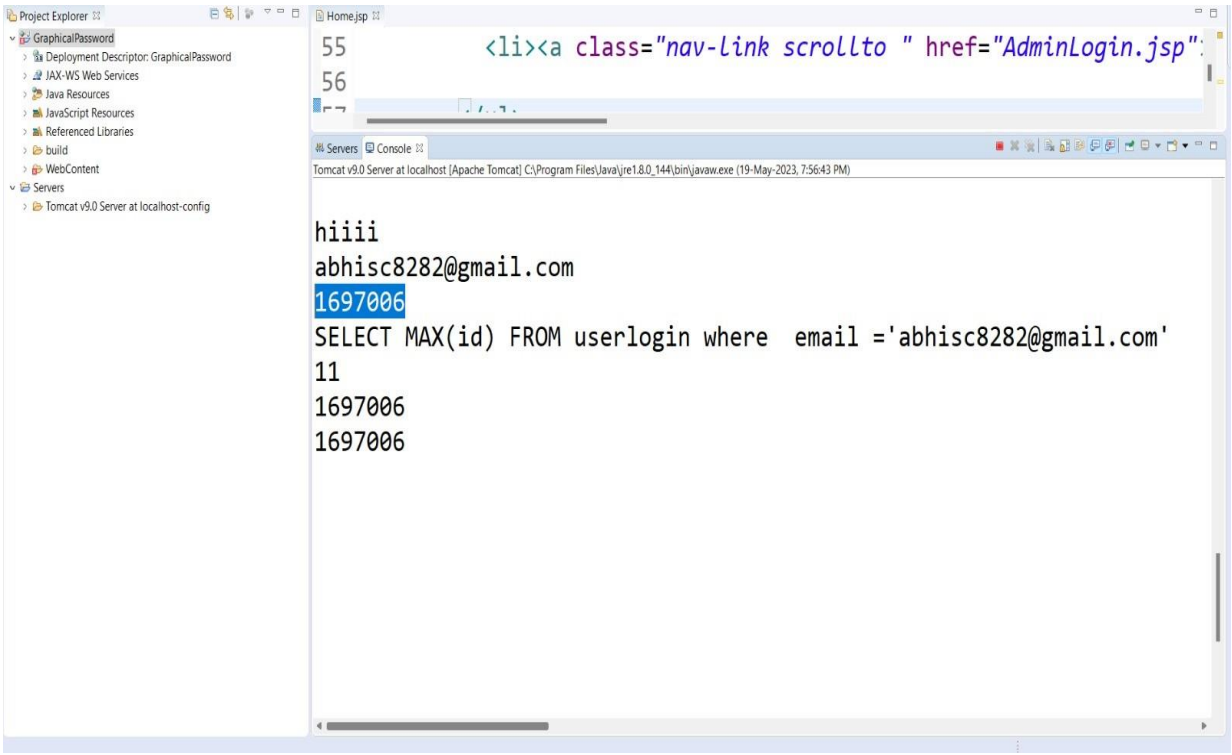


Figure 8.6: Otp Image

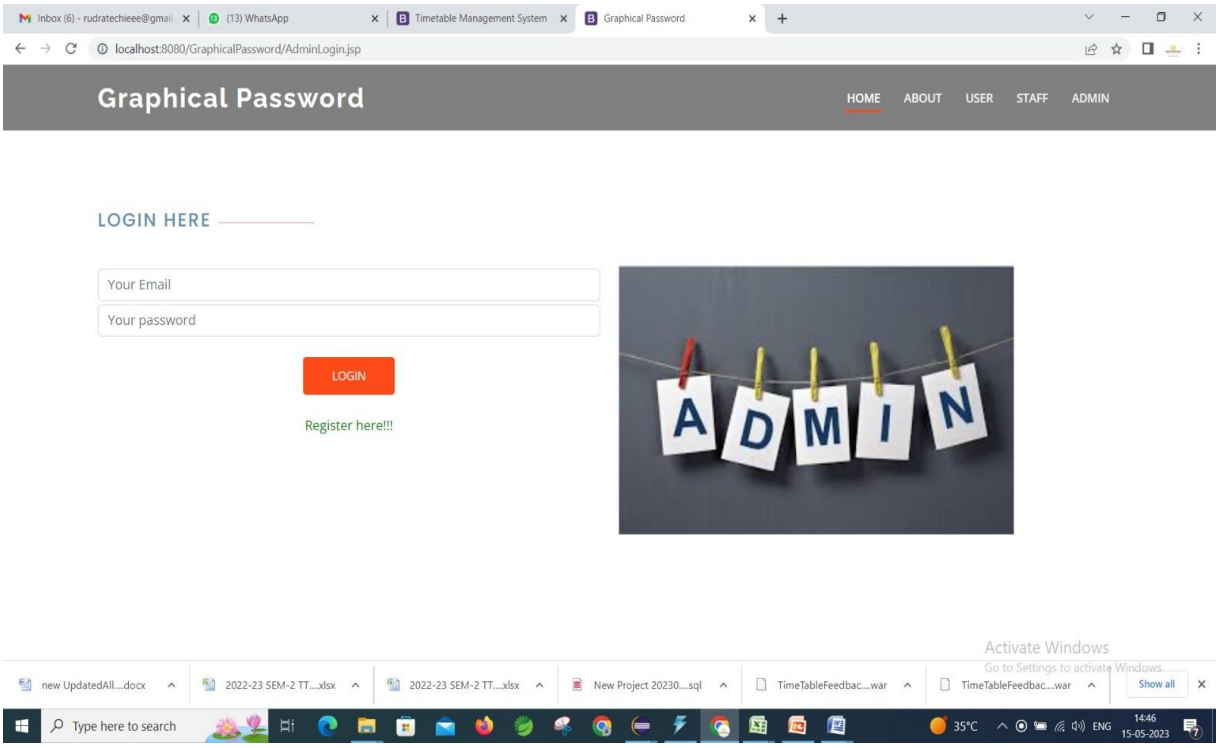


Figure 8.7: Admin Login Page

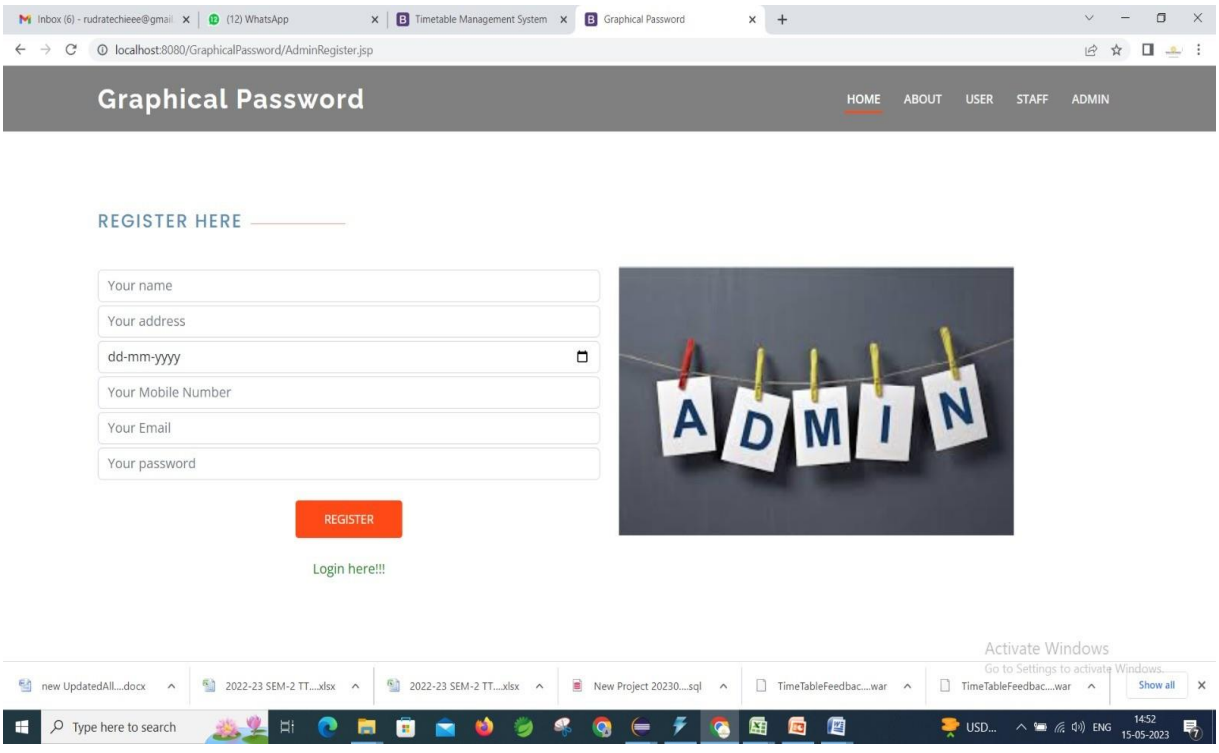


Figure 8.8: Admin Register Page

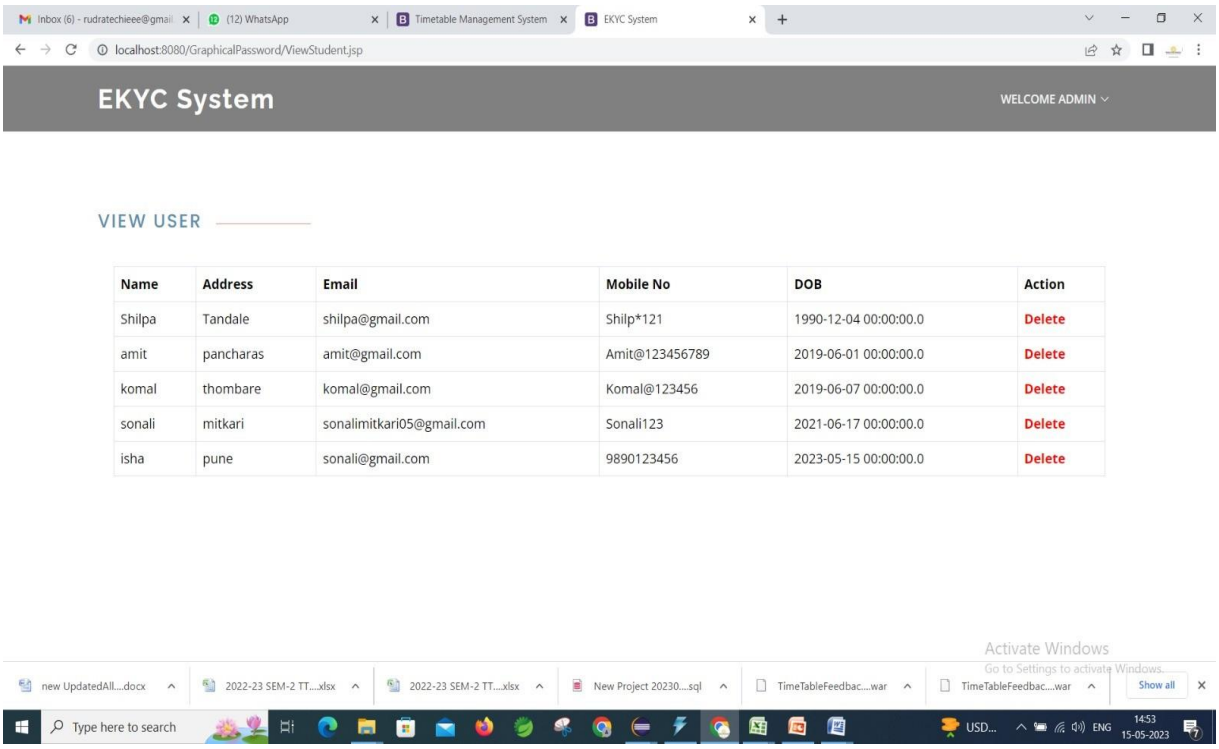


Figure 8.9: Admin Page

CHAPTER 9

SUMMARY AND CONCLUSION

Summary and Conclusion

User authentication is a fundamental component in most computer security contexts. In this extended abstract, we proposed a simple graphical password authentication system. The system combines graphical and text-based passwords trying to achieve the best of both worlds. It also provides multi-factor authentication in a friendly intuitive system. We described the system operation with some examples, and highlighted important aspects of the system.

CHAPTER 10

REFERENCES

References

1. William Stallings and Lawrie Brown. Computer Security: Principle and Practices. Pearson Education, 2008.

2. Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. Passpoints: design and longitudinal evaluation of a graphical password system. International Journal of Human-Computer Studies, 63:102–127, July 2005.

3. Robert Morris and Ken Thompson. Password security: a case history. Communications of the ACM, 22:594– 597, November 1979.

4. Daniel V. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In Proceedings of the 2nd USENIX UNIX Security Workshop, 1990.

5. Eugene H. Spafford. Observing reusable password choices. In Proceedings of the 3rd Security Symposium. Usenix, pages 299–312, 1992.

6. Sigmund N. Porter. A password extension for improved human factors. Computers Security, 1(1):54 – 56, 1982.

7. Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. Graphical passwords: A survey. In Proceedings of Annual Computer Security Applications Conference, pages 463–472, 2005.

8. Antonella De Angeli, Lynne Coventry, Graham Johnson, and Karen Renaud. Is a picture really worth a thousand words? exploring the feasibility of graphical authentication systems. International Journal of Human-Computer Studies, 63:128–152, July 2005.

9. Real User Corporation. The science behind passfaces, June 2004.

10. G. E. Blonder. Graphical password. U.S. Patent 5559961, Lucent Technologies, Inc. (Murray Hill, NJ), August 1995.