CONTENTS

1.INTRODUCTION

1.1 Synopsis

1.2 Over View of the project

2. SYSTEM CONFIGURATION

2.1 Hardware specification

2.2 Software specification

2.2.1 About the software

3. SYSTEM ANALYSIS

3.1 Existing system

3.2 Proposed system

4. SYSTEM DESIGN

4.1 Table design

4.2 Input & output design

5. SYSTEM TESTING

6. CONCLUSION

7. APPENDIX

7.1 Data flow diagram

7.2 Source code

7.3 Screen layout

8. BIBLIOGRAPHY

1.Introduction

1.1 Synopsis

General health examination is an integral part of healthcare in many countries. Identifying the participants at risk is important for early warning and preventive intervention. The fundamental challenge of learning a classification model for risk prediction lies in the unlabeled data that constitutes the majority of the collected dataset. Particularly, the unlabeled data describes the participants in health examinations whose health conditions can vary greatly from healthy to very-ill. There is no ground truth for differentiating their states of health. In this paper, we propose a graph-based, semi-supervised learning algorithm called SHG-Health (Semi-supervised Heterogeneous Graph on Health) for risk predictions to classify a progressively developing situation with the majority of the data unlabeled. An efficient iterative algorithm is designed and the proof of convergence is given. Extensive experiments based on both real health examination datasets and synthetic datasets are performed to show the effectiveness and efficiency of our method.

1.2 Over View of the project

It explores a Heterogeneous graph based on Health Examination Records called **HeteroHER** graph, where examination items in different categories are modelled as different types of nodes and their temporal relationships as links.To tackle large unlabeled data, SHG-Health features a semi-supervised learning method that utilizes both labeled and unlabeled instances. In addition, it is able to learn an additional K +1 “unknown” class for the participants who do not belong to the K known high-risk disease classes.

We present the SHG-Health algorithm to handle a challenging multi-class classification problem with substantial unlabeled cases which may or may not belong to the known classes. This work pioneers in risk prediction based on health examination records in the presence of large unlabeled data.

2. SYSTEM CONFIGURATION

2.1 Hardware specification

- System : Intel Core i3
- Generation : 7th Gen
- Hard Disk : 1000 GB
- Monitor : 15’’ LED
- Input Devices : Keyboard, Mouse
- Ram : 8GB

2.2 Software specification

- Operating system: Windows 10
- Coding Language: ASP.NET, C#.NET
- Tool: Visual Studio 2010
- Database: SQL SERVER 2008

2.2.1 About the software

ASP.NET

INTRODUCTION

With the advent of ASP.NET we see a shift from traditional scripting to the beginning of full-fledged programming online. VBScript isn't the only option anymore, as programmers can now employ the full power that lies behind both Visual Basic (VB) and C within their ASP.NET assemblies.

There is no denying the widespread acceptance that .NET received from the developer community. It's proven itself to be a well-developed framework with solid ideas on how the programming world should continue to change. The introduction of a software solution that enables anyone to code in any language that is compatible with the framework is groundbreaking to say the least.

Here we will take a look at how Active Server Pages (ASP) itself began just a couple of years ago and how it has captivated programmers ever since. It has had some problems, of course, but the .NET architecture seems to have found solutions to many preexisting programming problems. There have also been changes with how ASP works with the server and client, to provide the user with the information that you want to provide.

WHY ASP WAS NEEDED

Not all Web developers have the programming skills needed to write ISAPI applications, and because ISAPI requires the compilation of programs, there are extra steps in producing an ISAPI-based site that slow development down. Novice and intermediate programmers found the need to learn an industrial strength language, such C++, and compile even the simplest of their page logic into .dll files a real barrier.

Another problem facing development teams in the mid '90s was the fact that a Web site is a mixture of Hypertext Markup Language (HTML) and logic. They needed a way to mix the programmer's code with the designer's page-layout HTML. and designs without one messing up the other. There were many solutions to this problem, ranging from custom template systems to Sever Side Include (SSI) statements that told the server to execute code based on special HTML comment tags.

NET?

With NET, Microsoft is formalizing a vision of an Internet made up of an infinite number of interoperable Web applications or services, which will operate in concert to form a global exchange network. The .NET Framework is really a strategy to tie disparate platforms and devices together, moving data around in a far more efficient manner than it is currently.

NET is Microsoft's platform for Web Services Web Services allow applications to communicate and share data over the Internet, regardless of operating system or programming language.

The NET Framework is a high productivity, standards based, multi-language application execution environment that handles the essential "housekeeping" chores and eases deployment and management. It provides an application execution environment that manages memory, addresses, versioning issues, and improves the reliability. scalability, and security of applications. The .NET Framework consists of several parts, including the Common Language Runtime and ASP.NET.

NET CLIENTS

NET Clients are PCs, laptops, workstations, phones, handheld computers, Tablet PCs, game consoles, and other smart devices. All of these devices will have the ability to consume Web Services. .NET Clients use software that supports Web Services, and enable you to access your data regardless of location or type. The NET client software Microsoft will offer includes Windows 7, Window 8, and Windows 10. These applications will power PCs, laptops, workstations, smart phones, handheld computers, and Tablet PCs.

NET Servers

The NET Servers, including the Windows 10 server family, make up Microsoft NET's server infrastructure for developing, deploying, and managing Web Services Designed with performance in mind, the NET Servers will provide enterprises with the resources required to integrate their systems, applications, and partners via Web Services. The NET Enterprise Servers are SQL Server 2008 to store, retrieves, and analyzes relational data. Application Center 2008 to deploy and manage highly available and scalable Web applications. Internet Security and Acceleration Server 2008 for establishing secure,fast Internet connectivity.

Major Features of ASP.NET Server Controls

When you develop an ASP.NET Web Form, you can use the following type of controls: HTML Server Controls You can manipulate these controls at the server-side. Before dispatching a form to the client, the ASP Engine converts them to the equivalent HTML elements. These controls are included in the System.Web.UI.HtmlControls namespace.

Web Server Controls (also known as Web Controls or ASP.NET Web Form Controls) These are the new generation's controls developed by Microsoft. They have many useful built-in features, and a standard set of properties. In the HTML or aspx file, these are typically referenced with an asp: prefix such as asp:Label, asp:Button, or asp:TextBox. Besides the form-type server controls such as labels, button, and dropdown. there are a number of special-purpose controls like the Calendar and AdRotator controls. The ASP Engine also maps these controls to standard HTML. equivalent controls before dispatching the page to the client. These Web server controls are available in the System.Web.UI.WebControls namespace.

Validation Controls this set of controls provides Rapid Application Development (RAD) features for automatically checking the specified validity of user inputs. These controls are available in the System.Web.UI.WebControls namespace.

Custom Controls You can develop your own server controls by extending an existing control or group of controls to provide additional functionalities. There are two versions of custom controls: Web User Controls and Web Custom Controls. The Web User Controls are easy to develop, and are typically stored as asex files. The Web Custom Controls require in-depth knowledge of Object Oriented Programming and the Common Language Runtime (CLR). These are stored in compiled form as assemblies.

AutoPostBack Attributes of Server Controls

In this section, we will illustrate an important behavior of certain server-side controls. Some server-side controls can generate automatic postbacks on selected events. That means, to submit a form, we may not have to wait until the user clicks the submit button. For example, the SelectedIndexChange event of an asp:ListBox is an event that is capable of triggering a postback. If we want this mechanism to work, we will have to set the AutoPostBack property of the List box to "True."

Structure of an ASP.NET Web Form

- A Web Form is an ASP.NET technology that we use to create a programmable Web page. It can present information, using any markup language, to the user in any browser, and can use code on the server to implement application logic. In NET documentation, Microsoft has outlined the following characteristics of a Web form:
- A Web form of your design can run on a specific browser of your choice. Or it can run on any browser and automatically render the browser-compliant HTML.
- It is built on the Common Language Runtime, thereby providing a managed execution environment, type safety, inheritance, and dynamic compilation. It can be programmed in any CLR-supported language.

The ASP Object Model

The code above uses the CreateObject method of the ASP Server object to instantiate the ADO objects used for data access. In addition to providing a script execution engine, ASP also provides a set of six objects, including Server, to facilitate the development of Web applications. Here is a brief summary of these objects:

- The Request object is used to read data that was packaged inside the HTTP request for the page.
- The Response object allows you to inject data, including HTML, cookies, or redirection headers into the response stream that is sent back to the client's browser.
- A Session object is created when the first request from a particular client is processed, and it stays in scope until a timeout period expires following the last request from that user, allowing you to store data and objects that span multiple requests from one user.

- The Application object is similar to the Session object, but its data is shared across all client requests over the lifetime of the application, and it also allows you to write code that runs automatically when the applications start or ends.
- The Object Context object is used to commit or abort transactions managed by MTS or COM+.
- The Server object provides a set of generic utility methods for creating COM objects, encoding data as HTML or URL strings that can be embedded within the HTML sent back to a browser, and finding the actual file locations that correspond to virtual paths.

Validation Controls

A validation control enables us to validate an input and display an error message if necessary. It is very much like other server-side controls with certain additional methods and properties. First, the server treats it as an invisible control. After the user has entered erroneous data, it becomes visible. It is a powerful, rapid application development feature; however, a developer needs to understand its behavior and the methods thoroughly before he or she can appreciate it. There are certain rough edges in the Beta 2 version, which hopefully will be polished in the final product. The best strategy to learn the family of controls is to learn them one at a time, and finally to apply the summary validation.

Various types of validation controls are as follows:

- RequiredFieldValidator Checks if the input control has any value.
- RegularExpression Validator Checks the value against a regular expression (pattern). Compare Validator Checks if the value is acceptable compared to a given value or compared to the content of another control. RangeValidator Checks if the input control's value is within a specified range.
- CustomValidator Allows you to develop custom validation. ValidationSummary Reports a summary of all errors.

The Benefits of ASP.NET

Microsoft, realizing that ASP does possess some significant shortcomings, developed ASP.NET. ASP.NET is a set of components that provide developers with a framework with which to implement complex functionality. Two of the major improvements of ASP.NET over traditional ASP are scalability and availability. ASP.NET is scalable in that it provides state services that can be utilized to manage session variables across multiple Web servers in a server farm. Additionally, ASP.NET possesses a high performance process model that can detect application failures and recover from them. Along with improved availability and scalability, ASP.NET provides the following additional benefits:

- Simplified development: ASP.NET offers a very rich object model that developers can use to reduce the amount of code they need to write.
- Language independence: ASP pages must be written with scripting. In other words, ASP pages must be written in a language that is interpreted rather than compiled. ASP.NET allows compiled languages to be used, providing better performance and crosslanguage compatibility.
- Simplified deployment: With .NET components, deployment is as easy as copying a component assembly to its desired location.
- Crossclient capability: One of the foremost problems facing developers today is writing code that can be rendered correctly on multiple client types. For example, writing one script that will render correctly in Internet Explorer 5.5 and Netscape Navigator 4.7 and on a PDA and a mobile phone is very difficult, if not impossible, and time consuming. ASP.NET provides rich serverside components that can automatically produce output specifically targeted at each type of client.
- Web services: ASP.NET provides features that allow ASP.NET developers to effortlessly create Web services that can be consumed by any client that understands HTTP and XML, the de facto language for inter device communication.
- Performance: ASP.NET pages are compiled whereas ASP pages are interpreted. When an ASP.NET page is first requested, it is compiled and cached, or saved in memory, by the .NET Common Language Runtime (CLR). This cached copy can then be reused for each subsequent request for the page. Performance is thereby improved because after the first request, the code can run from a much faster compiled version.

ASP.NET application: authentication, authorization, and impersonation.

Authentication is the process of discovering and verifying the identity of a user or service by examining the user's credentials and validating those credentials against some authority such as an LDAP server, a database, an XML file or even a Web service such as Microsoft Passport. Several authentication mechanisms are available for use with the .NET Framework Role based security. ASP.NET natively supports Windows, Cookie, and Passport modes of authentication.

The purpose of authorization is to determine whether a user with a specific identity should be provided with a requested type of access to a given resource. This is typically handled by assigning an authenticated user to a predefined role. A role such as end user, super user, power user, administrator or anonymous is defined by the application and given access to execute certain files, run certain functions or add/update/delete certain data.

Impersonation is when an application assumes the user's identity as the request is passed to the application from IIS. Then, access is granted or denied based on the impersonated identity. So, we could establish two accounts in the application called generic User and superUser, we could then selectively have incoming Web clients run as one of these accounts depending upon the rules established during authorization for each specific user.

ADO.NET

ADO.NET should be your strategic choice for n-tier application development in a COM environment. While it is fairly easy for you to run your ADO code in an *.aspx page as .NET provides backward compatibility, you will not be able to take advantage of some key benefits of the .NET framework such as databinding. ADO uses RecordSet objects that follow guidelines that are no longer suitable in .NET. Therefore, you cannot use a RecordSet object to populate an ASP.NET or Windows Forms datagrid by attaching it as a datasource.

ADO.NET clearly provides some key advantages in applications that require a readily transportable disconnected data store or need to support a highly scalable middle tier. However, there is one core feature that ADO.NET doesn't support, the use of serverside cursors. A serverside cursor requires that an open connection be maintained between the application and the database. In client/server applications, this was a fairly common approach and there are a number of applications and situations where you still should use serverside cursors.

Understanding the Changes in ADO.NET

As mentioned above, ADO.NET has a relatively long history. As far as software development goes, if you are going to make dramatic enhancements, it is sometimes necessary to start from scratch, taking what you learned from the last implementation and looking forward with wisdom and clairvoyance. More than likely, it will result in a product that is not backward compatible and that requires significant change to bring older applications up to par.

The DataSet is really an in-memory relational database. A programmer will create one or more DataTable objects in a DataSet and "fill" them with data. A DataTable contains a collection of DataRows, each of which contains a collection of DataColumns. We can optionally create DataViews based on these DataTables, and even define relations to enforce data integrity. Again with all this functionality we really don't have the need for a Recordset object.

The DataSet requires a DataAdapter to actually interact with a data source. The DataAdapter represents the connection to a data source and the commands used to communicate with the data source to "fill" a DataSet or update a data source. After we are finished adding or updating data in the DataSet, the application would then call the Update method of the DataAdapter to INSERT, UPDATE, and DELETE records as appropriate at the data source.

The other thing to keep in mind, especially since we are developing for ASP.NET, is that since a DataSet is a disconnected copy of our data, it is most suitable for small mounts of data. For ASP.NET, one would expect to find most of the work of retrieving data to be done using a DataReader, with DataSets being used for relatively static data that must be retrieved often. A DataSet in this scenario could be used at the session level to save some processing at the data source. For example, a Web site might have a drop- down list that contains the 50 states in the United States. If this drop-down list is used more than once on a page, and the number of states is static, we could fill a DataSet and bind every instance of the drop-down list to this DataSet. This way we hit the database once for all 50 states and for all instances of the drop-down list, thus saving many database hits.

The DataReader can be thought of as a firehose Recordset. A firehose Recordset was a nickname given to a read-only, forward-only Recordset in classic ADO. So, a DataReader is a forward-only, non-updateable stream of data from the data provider. Consider this as proof of a DataReader's speed; a DataAdapter creates a DataReader behind the scenes to populate a DataSet. Because of this simple fact, the DataReader is very useful for ASP.NET work. In a stateless environment such as the Internet, fast access to the data is very important. It may be wasteful to retrieve this data into a DataSet, read through it once to render HTML, and then discard it. The point here is to be aware of the overhead that the DataSet has and use it when it makes sense.

In the case of the System.Data.OleDb namespace, we select the OLEDB provider in much the same way that we selected them in classic ADO. We specify the Provider attribute in the connection string. In the case of the System.Data.SqlClient namespace, Microsoft has written this namespace to bypass the OLEDB protocol and instead use the Tabular Data Stream (TDS) protocol. The TDS protocol is much more efficient than the OLEDB protocol and allows for much greater speed when working with data. The downside is that the System.Data, SqlClient namespace can only be used to interact with SQL Server versions 7.0 and up; therefore, we do not need to specify the Provider attribute when using the System Data, SqlClient namespace.

3. SYSTEM ANALYSIS

3.1 Existing system

- Huang et al. proposed iSELF, a SSL method based on local Fisher discrimination analysis for disease gene classification.
- Nguyen et al constructed a protein-protein interaction network, which defines interacted genes as candidate genes and the rest as negative genes for SSL based on Gaussian fields and harmonic functions.
- Garla et al. applied Laplacian SVM as a SSL approach for cancer case management.
- Wang et al. proposed a graph-based SSL method that is able to learn patient risk groups for patient risk stratification.
- Kim et al. proposed a co-training graph-based SSL method for breast cancer survivability prediction. It iteratively assigns pseudo-labels to unlabeled data when there is a consensus amongst the learners and includes the pseudo-labeled instances in the labeled set until the unlabeled set stops decreasing.

Disadvantages of existing system

- Most existing classification methods on healthcare data do not consider the issue of unlabeled data. They either have expert-defined low-risk or control classes or simply treat non-positive cases as negative.
- Methods that consider unlabeled data are generally based on Semi-Supervised Learning (SSL) that learns from both labeled and unlabeled data

3.2 Proposed system

- This paper proposes a semi-supervised heterogeneous graph-based algorithm called **SHG-Health** (Semisupervised Heterogeneous Graph on Health) as an evidence-based risk prediction approach to mining longitudinal health examination records.

- To handle heterogeneity, it explores a Heterogeneous graph based on Health Examination Records called **HeteroHER** graph, where examination items in different categories are modelled as different types of nodes and their temporal relationships as links.

- To tackle large unlabeled data, SHG-Health features a semi-supervised learning method that utilizes both labeled and unlabeled instances. In addition, it is able to learn an additional K +1 “unknown” class for the participants who do not belong to the K known high-risk disease classes.

Advantages of proposed system

- We present the SHG-Health algorithm to handle a challenging multi-class classification problem with substantial unlabeled cases which may or may not belong to the known classes. This work pioneers in risk prediction based on health examination records in the presence of large unlabeled data.

- A novel graph extraction mechanism is introduced for handling heterogeneity found in longitudinal health examination records.

- The proposed graph-based semi-supervised learning algorithm SHG-Health that combines the advantages from heterogeneous graph learning and class discovery shows significant performance gain on a large and comprehensive real health examination dataset.

4. SYSTEM DESIGN

4.1 Table design

Database Name: DataBaseHealthMining.mdf

Table Name:

1. UserRegister
2. AddDoctor
3. AddDataSet

Table Name: 1.UserRegister

COLUMN NAME	**DATA TYPE**	**LENGTH**
Id	int	
Name	varchar	50
Email	varchar	100
Password	varchar	50
Country	varchar	50
Phone	varchar	50

Table Name: 2.AddDoctor

Column Name	**Data Type**	**Length**
Id	Int	
Name	Varchar	50
MobileNo	Varchar	50
EmailId	Varchar	100
Gender	Varchar	50
Dob	Varchar	50
Address	Varchar	200
Specialised	Varchar	100

Table Name: 3.AddDataSet

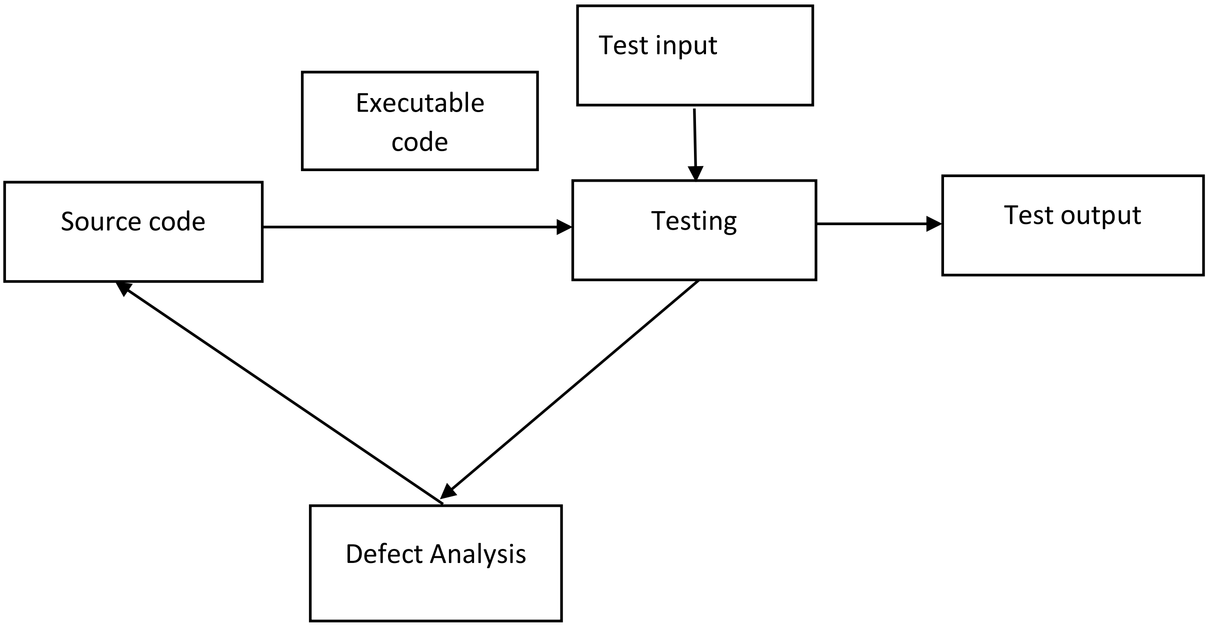
Column Name	**Data Type**	**Length**
Id	Int	
DiseaseName	Varchar	50
Age	Varchar	3
Gender	Varchar	50
ChestPaintype	Varchar	50
FastingBloodSugar	Varchar	50
RestingElectrographite	Varchar	50
ExerciseInducedAngina	Varchar	50
Slope	Varchar	50
NoOfMajorvessels	Varchar	50
Thal	Varchar	50
TrestBloodPressure	Varchar	50
SerumCholestrol	Varchar	50
MaxHeartRateAchieved	Varchar	50

STDepressionInducedByExercise	Varchar	50
Description	Varchar	2000

4.2 Input & output design

In the testing process the source code is converted into executable code and various inputs are given. For every test input the output is analyzed to check whether the program is functioning correctly or not

If there is default or bug in the testing, the defect is analyzed and the source code is modified and again the testing process is carried out with the test inputs. This process is repeated till the source code is defect free.



5. SYSTEM TESTING

SYSTEM TESTING

INTRODUCTION

Testing is a schedule process carried out by the software development team to capture all the possible errors, missing operations and also a complete verification to verify objective are met and user requirement are satisfied. The design of tests for software and other engineering products can be as challenging as the initial design to the product itself.

TESTING TYPES

A software engineering product can be tested in one of two ways:

- Black box testing
- White box testing

Black box testing

Knowing the specified function that a product has been designed to perform, determine whether each function is fully operational.

White box testing

Knowing the internal workings of a software product determine whether the internal operation implementing the functions perform according to the specification, and all the internal components have been adequately exercised.

2.TESTING STRATEGIES

Four Testing Strategies that are often adopted by the software development team Include:

- Unit Testing
- Integration Testing Validation Testing
- System Testing

This system was tested using Unit Testing and Integration Testing Strategies to to the project because there were the most relevant approaches for this project."

UNIT TESTING

We adopt white box testing when using this testing technique. This testing was carried out on individual components of the software that were designed. Each individual module was tested using this technique during the coding phase. Every component was checked to make sure that they adhere strictly to the specifications spelt out in the data flow diagram and ensure that they perform the purpose intended for them.

All the names of the variables are scrutinized to make sure that they are truly reflected of the element they represent. All the looping mechanisms were verified to ensure that they were as decided. Beside these, we trace through the code manually to capture syntax errors and logical errors.

INTEGRATION TESTING

After finishing the Unit Testing process, next is the integration testing process. In this testing process we put our focus on identifying the interfaces between components and their functionality as dictated by the DFD diagram. The Bottom up incremental approach was adopted during this testing. Low level modules are integrated and combined as a cluster before testing.

VALIDATION TESTING

Software testing and validation is achieved through a series of black box tests that demonstrate conformity with requirements. A test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both, the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met. After each validation test case has been conducted, one of the two possible conditions exists. They are,

- The function or performance characteristics conform to specification and are accepted.

- A deviation from specification is uncovered and a deficiency list is created. The deviation or error discovered at this stage in project can rarely be corrected prior to scheduled completion. It is necessary to negotiate with the customer to establish a method for resolving deficiencies.

SYSTEM TESTING

System testing is a series of different tests whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all the work should verify that all system elements have been properly integrated and perform allocated functions.

Recovery testing is done in such as way that failure is forced to a software system and checked whether the recovery is proper and accurate. The performance of the system is highly effective.

Software testing is critical element of software quality assurance and represents ultimate review of specification, design and coding. Test case design focuses on a set of technique for the creation of test cases that meet overall testing objectives. Planning and testing of a programming system involve formulating a set of test cases, which are similar to the real data that the system is intended to manipulate. In principle, testing of a program must be extensive Every statement in the program should be exercised and every possible path combination through the program should be executed at least once.

TESTINGS FOR OUR PROJECT

Manual testing is vital one to every project, before the installation the project at client place.

FEEDBACKS OF TESTING

Login-screen testing

If user enters the wrong username or password, error information is displaying on the screen. Hereby we can come to a conclusion; this area is working as promptly.

Auto Generation Test

If identification oriented fields are spawning and generate as automatically in project, volunteer entry is no need to user. Hereby user can avoid the following problems.

- In sequence entry
- Non related entry
- Repeated entry.

Triggering Test

Triggering means feting process. This is a core test to every project.

Following have tested by this section.

- Referential integration.
- Automatically updating.
- Auto deletion operation.

Violent-entry test

If user enters the violent values in any corner of this project, they are rejected and display the appropriated error information on screen. All kind of validation and restriction are checking in this section. Now this project is eligible to client place installation.

6. CONCLUSION

Through this project we have covered the all over composed activities of "Mining of Health Examination Records" This project is designed and developed as user friendly. Future enhancement facilities are attached with this project. In future we can modify our project, without existing system any damage.

This project helps us to improve our knowledge as well as the practical knowledge. All the manual works are computerized in our project. This project helps us to know about how to develop a project to the given title. We gained a lot of knowledge about the purpose of computer program.

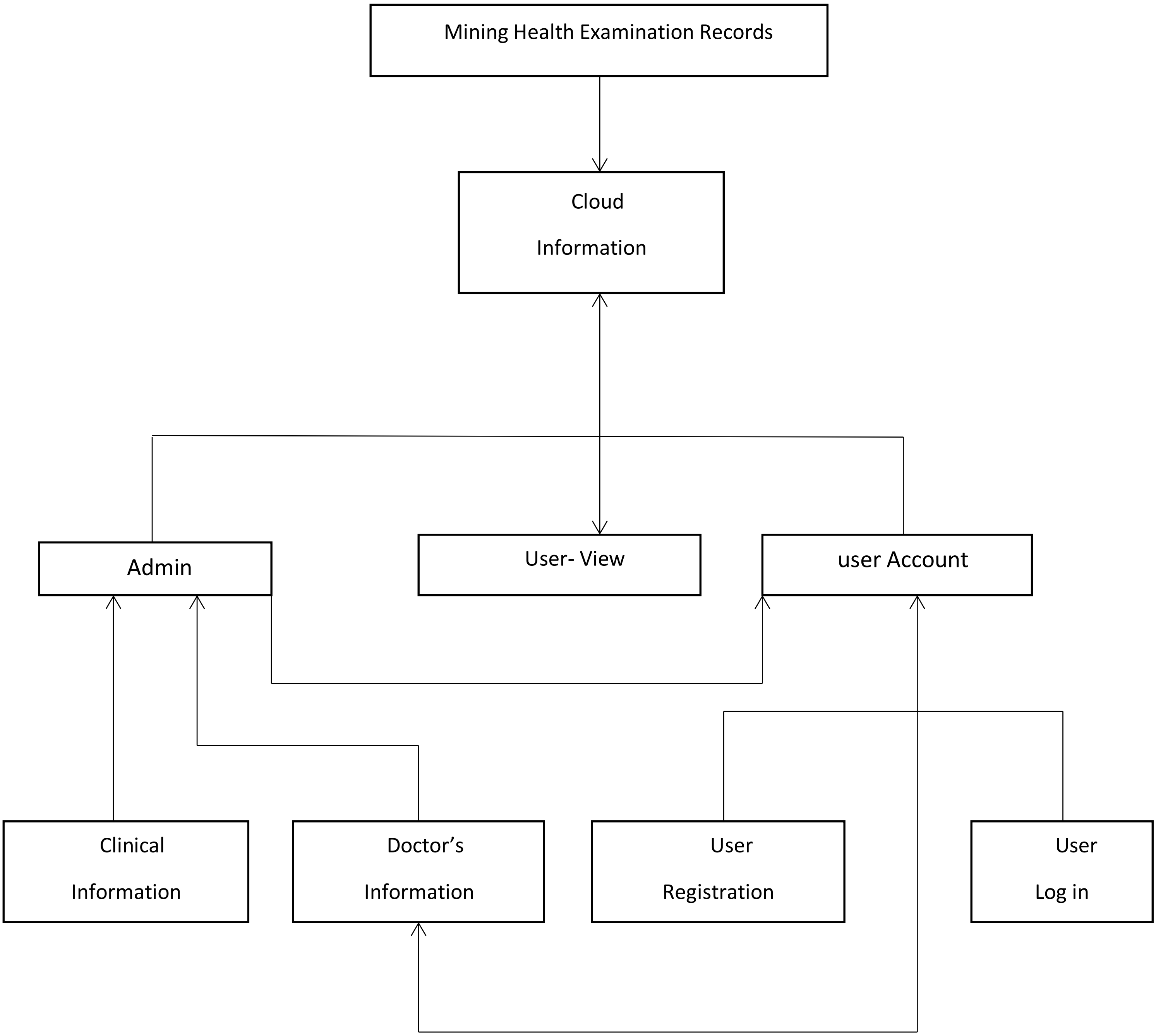
In our project all the manual works are computerized and also be modulated. Through this project we gained knowledge about the project overview and it will be useful for our future reference.

This project is fully satisfied the requirements of concern. It is also useful for improving our programming skills.

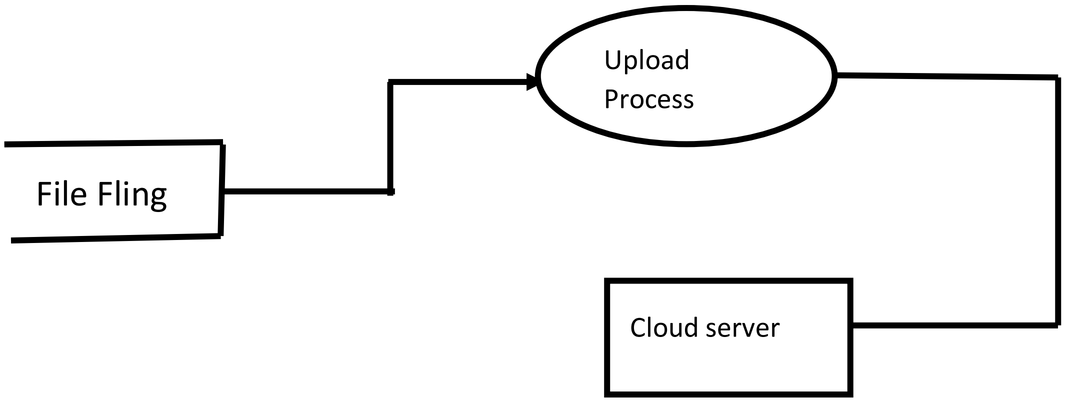
7. APPENDIX

7.1 Data flow diagram

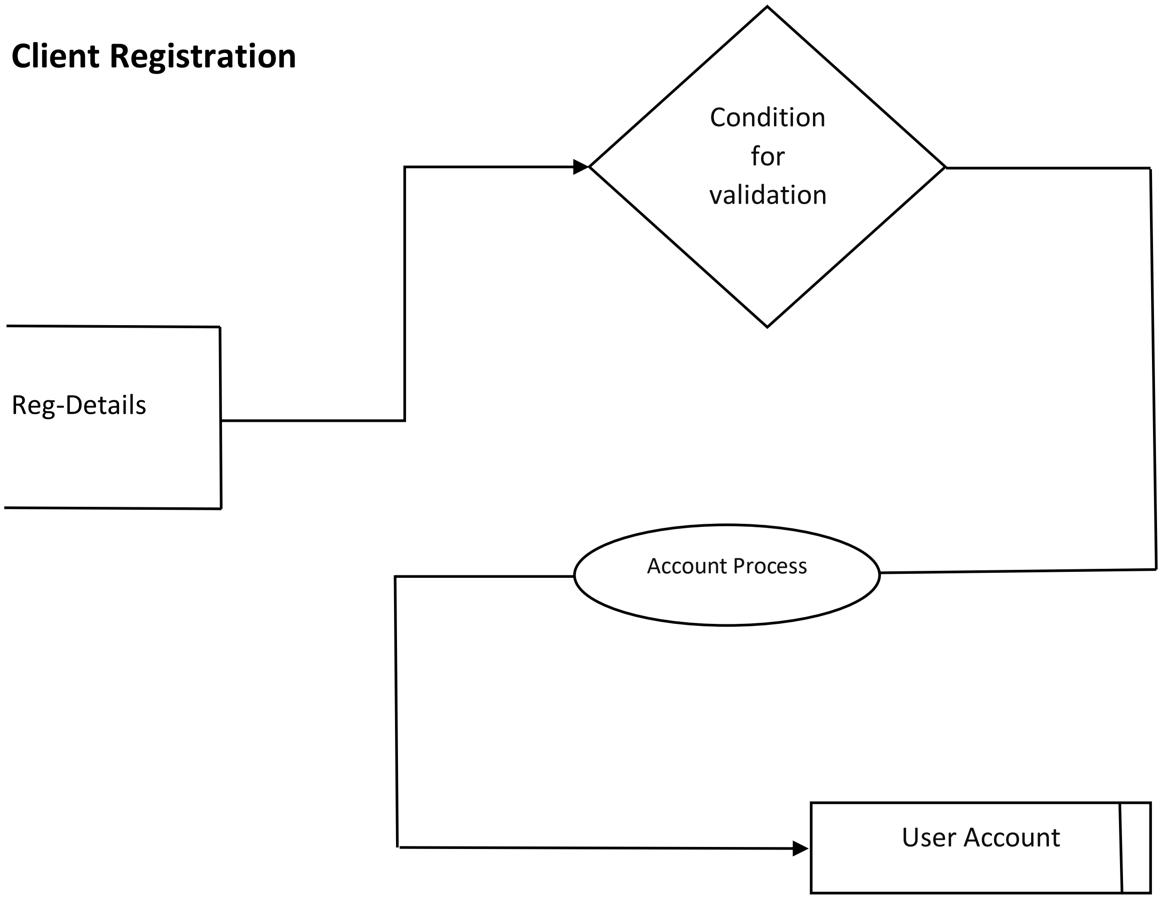
Architectural Diagram



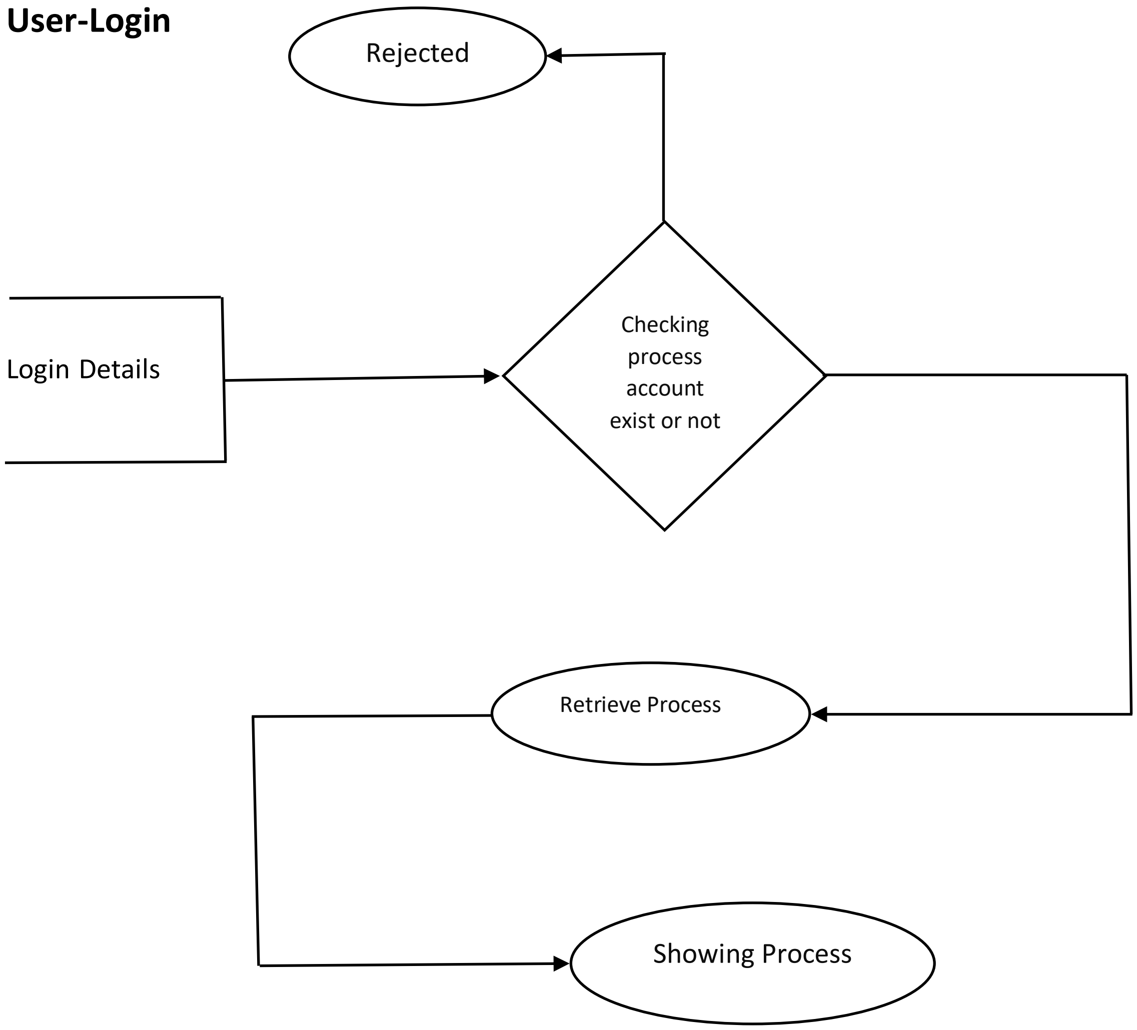
Admin upload



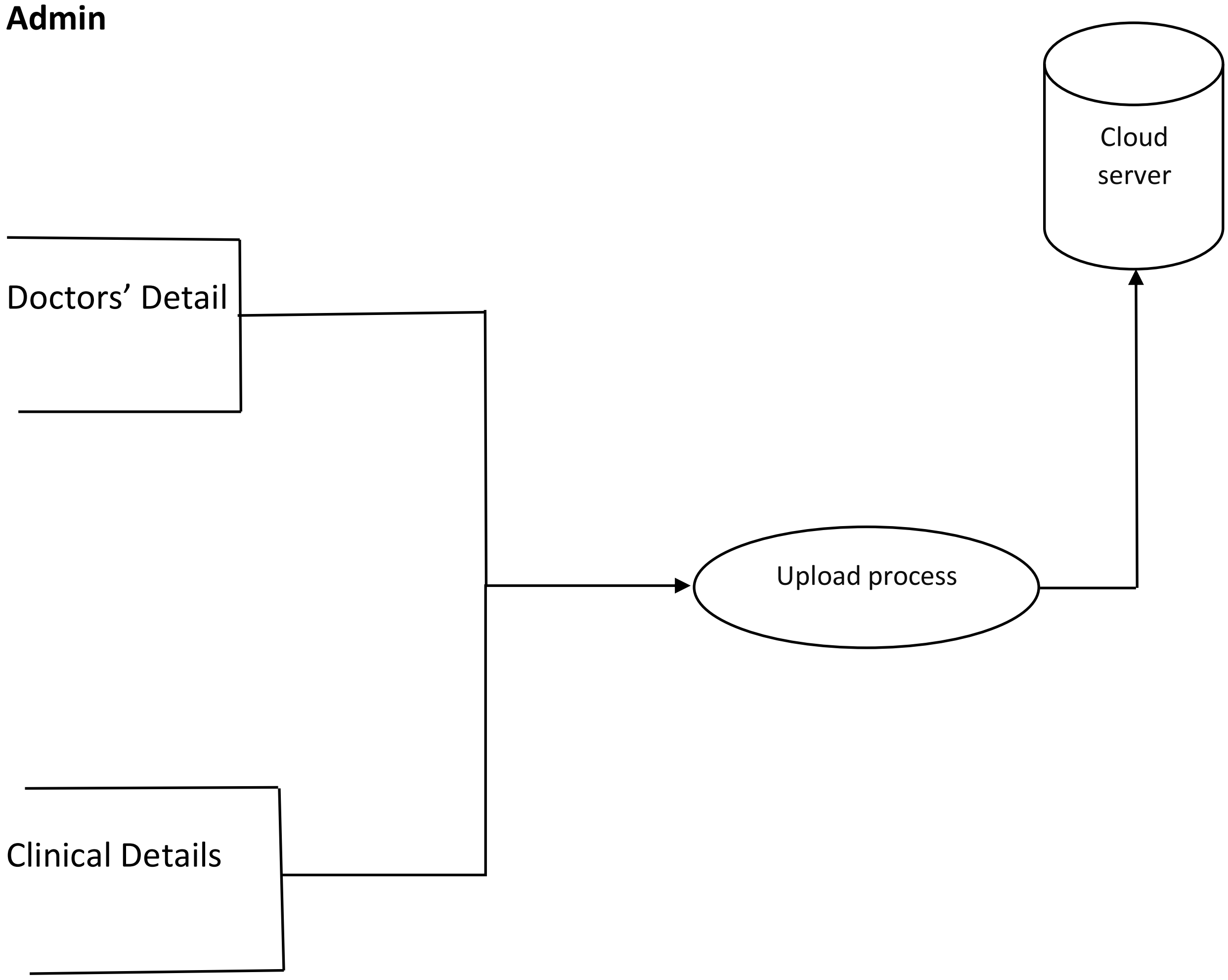
Client Registration



User-Login


Rejected
Login Details
Checking process account exist or not
Retrieve Process
Showing Process

Admin


Cloud server
Doctors' Detail
Upload process
Clinical Details

7.2 Source code & 7.3 Screen layout

Page:1

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="Admin.aspx.cs"
Inherits="Mining_Health.Admin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <div class="container">
        <br />
        <br />
            <div class="col-md-6">
                <div class="box-for overflow">
                    <div class="col-md-12 col-xs-12
register-blocks">
                        <h2 style="color: Red;">
                            <b>Admin Login : </b>
                        </h2>
                        <form action="" method="post">
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Email</label>
                            <input type="text"
class="form-control" id="TxtEmail" runat="server" />
                        </div>
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Password</label>
                            <input type="password"
class="form-control" id="TxtPassword" runat="server" />
                        </div>
                        <br />
                        <div class="text-center">
                            <asp:Button ID="Button1"
Text="Login" CssClass="btn btn-info" runat="server"
                                onclick="Button1_Click" />
                        </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </center>
</asp:Content>
```

Page:2

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="Admin.aspx.cs"
Inherits="Mining_Health.Admin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
```

```
        <div class="container">
        <br />
        <br />
            <div class="col-md-6">
                <div class="box-for overflow">
                    <div class="col-md-12 col-xs-12
register-blocks">
                        <h2 style="color: Red;">
                            <b>Admin Login : </b>
                        </h2>
                        <form action="" method="post">
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Email</label>
                            <input type="text"
class="form-control" id="TxtEmail" runat="server" />
                        </div>
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Password</label>
                            <input type="password"
class="form-control" id="TxtPassword" runat="server" />
                        </div>
                        <br />
                        <div class="text-center">
                            <asp:Button ID="Button1"
Text="Login" CssClass="btn btn-info" runat="server"
                                onclick="Button1_Click" />
                        </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </center>
</asp:Content>
```

Page: 3

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="Admin.aspx.cs"
Inherits="Mining_Health.Admin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <div class="container">
        <br />
        <br />
            <div class="col-md-6">
                <div class="box-for overflow">
                    <div class="col-md-12 col-xs-12
register-blocks">
                        <h2 style="color: Red;">
                            <b>Admin Login : </b>
                        </h2>
                        <form action="" method="post">
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Email</label>
                            <input type="text"
class="form-control" id="TxtEmail" runat="server" />
                        </div>
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Password</label>
                            <input type="password"
class="form-control" id="TxtPassword" runat="server" />
```

```
                        </div>
                        <br />
                        <div class="text-center">
                            <asp:Button ID="Button1"
Text="Login" CssClass="btn btn-info" runat="server"
                                onclick="Button1_Click" />
                        </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </center>
</asp:Content>
```

Page: 4

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="Admin.aspx.cs"
Inherits="Mining_Health.Admin" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <div class="container">
        <br />
        <br />
            <div class="col-md-6">
                <div class="box-for overflow">
```

```
                    <div class="col-md-12 col-xs-12
register-blocks">
                        <h2 style="color: Red;">
                            <b>Admin Login : </b>
                        </h2>
                        <form action="" method="post">
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Email</label>
                            <input type="text"
class="form-control" id="TxtEmail" runat="server" />
                        </div>
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Password</label>
                            <input type="password"
class="form-control" id="TxtPassword" runat="server" />
                        </div>
                        <br />
                        <div class="text-center">
                            <asp:Button ID="Button1"
Text="Login" CssClass="btn btn-info" runat="server"
                                onclick="Button1_Click" />
                        </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </center>
</asp:Content>
```

Page: 5

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Admin.Master" AutoEventWireup="true"
    CodeBehind="AdminViewDataSet.aspx.cs"
Inherits="Mining_Health.AdminViewDataSet" %>
```

```
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <br />
        <br />
        <asp:DataGrid ID="GridView1" runat="server"
CellPadding="4" ForeColor="#333333" GridLines="None">
            <AlternatingItemStyle BackColor="White"
ForeColor="#284775" />
            <EditItemStyle BackColor="#999999" />
            <FooterStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
            <HeaderStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
            <ItemStyle BackColor="#F7F6F3"
ForeColor="#333333" />
            <PagerStyle BackColor="#284775"
ForeColor="White" HorizontalAlign="Center" />
            <SelectedItemStyle BackColor="#E2DED6" Font-
Bold="True" ForeColor="#333333" />
        </asp:DataGrid>
    </center>
</asp:Content>
```

Page: 6

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Admin.Master" AutoEventWireup="true"
CodeBehind="AdminViewDoctor.aspx.cs"
Inherits="Mining_Health.AdminViewDoctor" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<center>
        <br />
        <br />
        <asp:DataGrid ID="GridView2" runat="server"
CellPadding="4" ForeColor="#333333" GridLines="None">
            <AlternatingItemStyle BackColor="White"
ForeColor="#284775" />
            <EditItemStyle BackColor="#999999" />
            <FooterStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
            <HeaderStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
            <ItemStyle BackColor="#F7F6F3"
ForeColor="#333333" />
            <PagerStyle BackColor="#284775"
ForeColor="White" HorizontalAlign="Center" />
            <SelectedItemStyle BackColor="#E2DED6" Font-
Bold="True" ForeColor="#333333" />
        </asp:DataGrid>
    </center>
</asp:Content>
```

Page: 7

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Admin.Master" AutoEventWireup="true"
CodeBehind="AdminViewUsers.aspx.cs"
Inherits="Mining_Health.AdminViewUsers" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<center>
        <br />
        <br />
        <asp:DataGrid ID="GridView3" runat="server"
CellPadding="4" ForeColor="#333333" GridLines="None">
            <AlternatingItemStyle BackColor="White"
ForeColor="#284775" />
            <EditItemStyle BackColor="#999999" />
            <FooterStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
            <HeaderStyle BackColor="#5D7B9D" Font-
Bold="True" ForeColor="White" />
```

```
            <ItemStyle BackColor="#F7F6F3"
ForeColor="#333333" />
            <PagerStyle BackColor="#284775"
ForeColor="White" HorizontalAlign="Center" />
            <SelectedItemStyle BackColor="#E2DED6" Font-
Bold="True" ForeColor="#333333" />
        </asp:DataGrid>
    </center>
</asp:Content>
```

Page: 8

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="Contactus.aspx.cs"
Inherits="Mining_Health.Contactus" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
        #rrr
        {
            background-color: RGB(20, 90, 50, .8);
            width: 700px;
            color: White;
            font-size: large;
            text-align: center;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <br />
        <br />
        <h2 style="color: Red;">
            <b>Contact Us:</b></h2>
        <br />
        <div id="rrr">
            <p style="color: Orange;">
                <b>Submitted By</b></p>
            <br />
            <p>
                <b>Student Name</b></p>
            <br />
            <p>
                <b>Department Name</b></p>
            <br />
            <p style="color: Orange;">
                <b>Guided By</b></p>
            <br />
            <p>
                <b>Guide Name</b></p>
            <br />
            <p>
                <b>College Name</b></p>
        </div>
    </center>
</asp:Content>
```

Page: 9

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeBehind="Home.master.cs" Inherits="Mining_Health.Home"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
    <link href="css/bootstrap.min.css" rel="stylesheet"
type="text/css" />
```

```
    <script src="js/bootstrap.min.js" 
type="text/javascript"></script>

    <nav class="navbar navbar-expand-lg navbar navbar-dark 
bg-dark">
  <a class="navbar-brand" href="HomePage.aspx">Clinical 
Automata For Problem Diagnose</a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarText" aria-
controls="navbarText" aria-expanded="false" aria-
label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarText">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="HomePage.aspx">Home 
<span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="Admin.aspx">Admin</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="User.aspx">User</a>
      </li>
       <li class="nav-item">
        <a class="nav-link" 
href="Register.aspx">Registration</a>
      </li>
       <li class="nav-item">
        <a class="nav-link" 
href="Contactus.aspx">ContactUs</a>
      </li>
    </ul>



  </div>
</nav>

</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1"
runat="server">

        </asp:ContentPlaceHolder>
    </div>
    </form>
</body>
</html>
```

Page: 10

```
 <%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="HomePage.aspx.cs"
Inherits="Mining_Health.HomePage" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("Hospital.jpg");
            background-size: cover;
        }
        #ss
        {
            text-align: center;
            padding-top: 100px;
        }
        #rrr
        {
            background-color: RGB(20, 90, 50, .8);
            width: 900px;
            color: White;
            font-size: large;
            text-align: center;
        }
    </style>
</asp:Content>
```

```
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <div id="ss">
        <h1 style="color: Red;">
            <b>Mining Health Examination Record</b></h1>
        <br />
        <center>
            <div id="rrr">
                <h2 style="color: orange;">
                    <b><u>Abstract</u></b></h2>
                <br />
                <p>
                    <b>General health examination is an
integral part of healthcare in many countries. Identifying
                        the participants at risk is
important for early warning and preventive intervention.
                        The fundamental challenge of
learning a classification model for risk prediction
                        lies in the unlabeled data that
constitutes the majority of the collected dataset.
                        Particularly, the unlabeled data
describes the participants in health examinations
                        whose health conditions can vary
greatly from healthy to very-ill. There is no ground
                        truth for differentiating their
states of health. In this paper, we propose a graph-based,
                        semi-supervised learning algorithm
called SHG-Health (Semi-supervised Heterogeneous
                        Graph on Health) for risk
predictions to classify a progressively developing
situation
                        with the majority of the data
unlabeled. An efficient iterative algorithm is designed
                        and the proof of convergence is
given. Extensive experiments based on both real
                        health examination datasets and
synthetic datasets are performed to show the effectiveness
                        and efficiency of our
method.</b></p>
            </div>
        </center>
    </div>
```

```
</asp:Content>
```

Page: 11

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="Register.aspx.cs"
Inherits="Mining_Health.Register" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <div class="sss">
            <br />
            <br />
            <table class="sssss" border='0' width='480px'
cellpadding='0' cellspacing='0' align='center'>
                <tr>
                    <td>
                        <h1 style="color: Red;"><b>
                            Signup form!</b></h1>
                    </td>
                </tr>
                <table class="ssss" border='0'
width='480px' cellpadding='0' cellspacing='0'
align='center'>
                    <tr>
                        <td align='center' style="color:
Blue; font-size:larger;">
                            Name:
                        </td>
                        <td>
                            <input type='text' name='name'
id="txtname" runat="server" required class="form-control"
/>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center' style="color:
Blue; font-size:larger;">
                            Email:
                        </td>
                        <td>
                            <input type='text' name='name'
id="txtemail" runat="server" required class="form-control"
/>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center' style="color:
Blue; font-size:larger;">
                            Password:
                        </td>
                        <td>
                            <input type="password"
name='name' id="txtpassword" runat="server" required
class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center' style="color:
Blue; font-size:larger;">
                            Country:
                        </td>
                        <td>
                            <input type='text' name='name'
id="txtcountry" runat="server" required class="form-
control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center' style="color:
Blue; font-size:larger;">
                            Phone:
                        </td>
                        <td>
                            <input type='text' name='name'
id="txtphone" runat="server" required class="form-control"
/>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <table border='0' cellpadding='0'
cellspacing='0' width='480px' align='center'>
                        <tr>
                            <td align='center'>
                                <asp:Button ID="Button1"
Text="Signup!" CssClass="btn btn-info"
                                    runat="server"
onclick="Button1_Click" />
                            </td>
                        </tr>
                    </table>
                </table>
            </table>
        </div>
    </center>
</asp:Content>
```

Page: 12

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Home.Master" AutoEventWireup="true"
    CodeBehind="User.aspx.cs"
Inherits="Mining_Health.User" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
    <style>
        body
        {
            background-image: url("stethoscope.jpg");
            background-size: cover;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <div class="container">
        <br />
        <br />
            <div class="col-md-6">
                <div class="box-for overflow">
                    <div class="col-md-12 col-xs-12
register-blocks">
                        <h2 style="color: Red;">
                            <b>User Login : </b>
                        </h2>
                        <form action="" method="post">
                        <div class="form-group">
```

```
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Email</label>
                            <input type="text"
class="form-control" id="TxtEmail" runat="server" />
                        </div>
                        <div class="form-group">
                            <label for="name"
style="color: Blue; font-size: larger;">
                                Password</label>
                            <input type="password"
class="form-control" id="TxtPassword" runat="server" />
                        </div>
                        <br />
                        <div class="text-center">
                            <asp:Button ID="Button1"
Text="Login" CssClass="btn btn-info" runat="server"
                                onclick="Button1_Click" />
                        </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </center>
</asp:Content>
```

Page: 13

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeBehind="User.master.cs" Inherits="Mining_Health.User1"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>

    <link href="css/bootstrap.min.css" rel="stylesheet" 
type="text/css" />
    <script src="js/bootstrap.min.js" 
type="text/javascript"></script>
     <nav class="navbar navbar-expand-lg navbar navbar-
dark bg-dark">
  <a class="navbar-brand" href="HomePage.aspx">Clinical 
Automata For Problem Diagnose</a>
  <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarText" aria-
controls="navbarText" aria-expanded="false" aria-
label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarText">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="HomePage.aspx">Home 
<span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" 
href="UserHome.aspx">Predictions</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" 
href="UserViewDoctor.aspx">UserViewDoctor</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" 
href="HomePage.aspx">Logout</a>
      </li>
    </ul>


  </div>
</nav>

</head>
<body>
```

```
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1"
runat="server">

        </asp:ContentPlaceHolder>
    </div>
    </form>
</body>
</html>
```

Page: 14

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/User.Master" AutoEventWireup="true"
    CodeBehind="UserHome.aspx.cs"
Inherits="Mining_Health.UserHome" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <center>
        <div class="sss">
            <br />
            <br />
            <table class="sssss" border='0' width='480px'
cellpadding='0' cellspacing='0' align='center'>
                <tr>
                    <td>
                        <h1>
                            Predict Results!</h1>
                    </td>
                </tr>
                <table class="ssss" border='0'
width='480px' cellpadding='0' cellspacing='0'
align='center'>
                    <tr>
                        <td align='center'>
                            Age:
                        </td>
                        <td>
                            <input type="text" name='name' 
id="T12" required runat="server" class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            ChestPaintype
                        </td>
                        <td>
                            <input type="text" name='name' 
id="T144" required runat="server" class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            FastingBloodSugar:
                        </td>
                        <td>
                            <input type='text' name='name' 
id="T15" required runat="server" class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            RestingElectrographite:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T16" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            ExerciseInducedAngina:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T17" runat="server" class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            Slope:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T18" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            No of Major vessels :
                        </td>
```

```
                        <td>
                            <input type='text' name='name'
id="T19" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            Thal:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T20" runat="server" class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            Trest Blood Pressure :
                        </td>
                        <td>
                            <input type='text' name='name'
id="T21" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            Serum cholestrol:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T22" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            MaxHeartRateAchieved:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T23" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
                        <td align='center'>
                            ST Depression Induced by
Exercise(old Peak:
                        </td>
                        <td>
                            <input type='text' name='name'
id="T24" runat="server" required class="form-control" />
                        </td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <tr>
```

```
                    <td><asp:Label ID="LblInfo" Font-
Bold="true" ForeColor="Blue" Font-Size="X-Large"
runat="server"></asp:Label></td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;
                        </td>
                    </tr>
                    <table border='0' cellpadding='0'
cellspacing='0' width='480px' align='center'>
                        <tr>
                            <br />
                            <td align='center'>
                                <asp:Button ID="Button1"
OnClick="PredictResult11" Text="PredictResult"
CssClass="btn btn-success"
                                    runat="server" />
                            </td>
                        </tr>
                    </table>
                </table>
            </table>
        </div>
    </center>
</asp:Content>
```

Page: 15

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/User.Master" AutoEventWireup="true"
CodeBehind="UserViewDoctor.aspx.cs"
Inherits="Mining_Health.UserViewDoctor" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2"
ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
<center>
        <br />
        <br />
        <asp:DataGrid ID="GridView2" runat="server"
BackColor="#DEBA84"
            BorderColor="#DEBA84" BorderStyle="None"
BorderWidth="1px" CellPadding="3"
            CellSpacing="2">
            <FooterStyle BackColor="#F7DFB5"
ForeColor="#8C4510" />
            <HeaderStyle BackColor="#A55129" Font-
Bold="True" ForeColor="White" />
            <ItemStyle BackColor="#FFF7E7"
ForeColor="#8C4510" />
            <PagerStyle ForeColor="#8C4510"
HorizontalAlign="Center" Mode="NumericPages" />
            <SelectedItemStyle BackColor="#738A9C" Font-
Bold="True" ForeColor="White" />
        </asp:DataGrid>
    </center>
</asp:Content>
```

Page: 16

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Mining_Health
{
    public partial class Admin : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {

        }

        protected void Button1_Click(object sender,
EventArgs e)
        {
            if ((TxtEmail.Value == "Admin") &&
(TxtPassword.Value == "Admin"))
            {
                Response.Redirect("AdminAddDoctor.aspx");
            }
            else
            {

ClientScript.RegisterStartupScript(GetType(), "alert",
"alert('Email or Password is invalid')", true);
            }
        }
    }
}
```

Page: 17

```
;
using System.Web.UI.WebControls;
using BusinessBA;

namespace Mining_Health
{
    public partial class AdminAddDataSet :
System.Web.UI.Page
    {
        AddDataSetBA objAddDataSetBA = new AddDataSetBA();
        protected void Page_Load(object sender, EventArgs
e)
        {

        }

        protected void Button1_Click(object sender,
EventArgs e)
        {
            int result =
objAddDataSetBA.AddDataSet(T11.Value, T12.Value, T13.Text,
T14.Value, T15.Value, T16.Value, T17.Value, T18.Value,
T19.Value, T20.Value, T21.Value, T22.Value, T23.Value,
T24.Value, txtDescription.Text);
            if (result > 0)
            {
```

```
ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "alertMessage",
"alert('DetailsAddedSuccessfully')", true);
            }
        }
    }
}
```

Page: 18

```
using System.Web.UI.WebControls;
using BusinessBA;

namespace Mining_Health
{
    public partial class AdminAddDoctor :
System.Web.UI.Page
    {
        AddDoctorBA objAddDoctorBA = new AddDoctorBA();
        protected void Page_Load(object sender, EventArgs
e)
        {

        }

        protected void Button1_Click(object sender,
EventArgs e)
        {

            int result =
objAddDoctorBA.AddDoctorDetails(TxtName.Value,
TxtMobile.Value, TxtEmail.Value, TxtGender.Text,
TxtDob.Value, TxtAddress.Text, TxtSpecialised.Text);
            if (result > 0)
            {


ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "alertMessage",
"alert('DoctorAddedSuccessfully')", true);

            }
        }
    }
}
```

Page: 19

```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace Mining_Health
{
    public partial class AdminViewDataSet :
System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {
          using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["Sqlc
onn"].ConnectionString))
            {
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = "select * from
AddDataSet";

                    SqlDataReader dr =
cmd.ExecuteReader();
                    dr.Read();
                        GridView1.DataSource = dr;
                        GridView1.DataBind();
                }
            }
        }



    }
}
```

Page: 20

```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace Mining_Health
{
    public partial class AdminViewDataSet :
System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {
          using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["Sqlc
onn"].ConnectionString))
            {
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = "select * from
AddDataSet";

                    SqlDataReader dr =
cmd.ExecuteReader();
                    dr.Read();
                        GridView1.DataSource = dr;
                        GridView1.DataBind();
                }
            }
        }



    }
}
```

Page: 21

```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace Mining_Health
{
    public partial class AdminViewDoctor :
System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {
            using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["Sqlc
onn"].ConnectionString))
            {
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = "select * from
AddDoctor";

                    SqlDataReader dr =
cmd.ExecuteReader();
                    if (dr.Read())
                    {
                        GridView2.DataSource = dr;
                        GridView2.DataBind();
                    }
                    else
                    {

                    }
                }
            }
        }

    }
}
```

Page: 22

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace Mining_Health
{
    public partial class AdminViewDoctor :
System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {
            using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["Sqlc
onn"].ConnectionString))
            {
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = "select * from
AddDoctor";

                    SqlDataReader dr =
cmd.ExecuteReader();
                    if (dr.Read())
                    {
                        GridView2.DataSource = dr;
                        GridView2.DataBind();
                    }
                    else
                    {

                    }
                }
            }
        }

    }
}
```

Page: 23

```
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace Mining_Health
{
    public partial class AdminViewUsers :
System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {
            using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["Sqlc
onn"].ConnectionString))
            {
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = "select * from
UserRegister";

                    SqlDataReader dr =
cmd.ExecuteReader();
```

```
                    if (dr.Read())
                    {
                        GridView3.DataSource = dr;
                        GridView3.DataBind();
                    }
                    else
                    {

                    }
                }
            }
        }
    }
}
```

Page: 24

```
using System.Web.UI.WebControls;
using System.Data;
using BusinessBA;

namespace Mining_Health
{
    public partial class User : System.Web.UI.Page
    {
        LoginBA objLoginBA = new LoginBA();
        protected void Page_Load(object sender, EventArgs
e)
        {

        }

        protected void Button1_Click(object sender,
EventArgs e)
        {
            DataTable dt = new DataTable();
            dt =
objLoginBA.GetLogin11(TxtEmail.Value,TxtPassword.Value);
            if (dt.Rows.Count > 0)
            {
                Response.Redirect("UserHome.aspx");
            }
            else
            {

ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "alertMessage", "alert('Your Username or
Password is incorrect')", true);
            }
        }
    }
}
```

Page: 25

```
using BusinessBA;
using System.Data;

namespace Mining_Health
{
    public partial class UserHome : System.Web.UI.Page
    {
        PredictResultBA objPredictResultBA = new
PredictResultBA();
        protected void Page_Load(object sender, EventArgs
e)
        {

        }
        public void PredictResult11(object o, EventArgs e)
        {
            string
result=objPredictResultBA.PredictResult(T12.Value,T144.Val
ue,T15.Value,T16.Value,T17.Value,T18.Value,T19.Value,T20.V
alue,T21.Value,T22.Value,T23.Value,T24.Value);
            if(result!="")
            {
                Session["ShowResult"]=result;

LblInfo.Text=Session["ShowResult"].ToString();
```

```
                T12.Value = "";
                T144.Value = "";
                T15.Value = "";
                T16.Value = "";
                T17.Value = "";
                T18.Value = "";
                T19.Value = "";
                T20.Value = "";
                T21.Value = "";
                T22.Value = "";
                T23.Value = "";
                T24.Value = "";

            }
            else
            {

ScriptManager.RegisterClientScriptBlock(this,
this.GetType(), "alertMessage", "alert('ResultNotFound')",
true);
            }

        }
    }
}
```

Page: 26

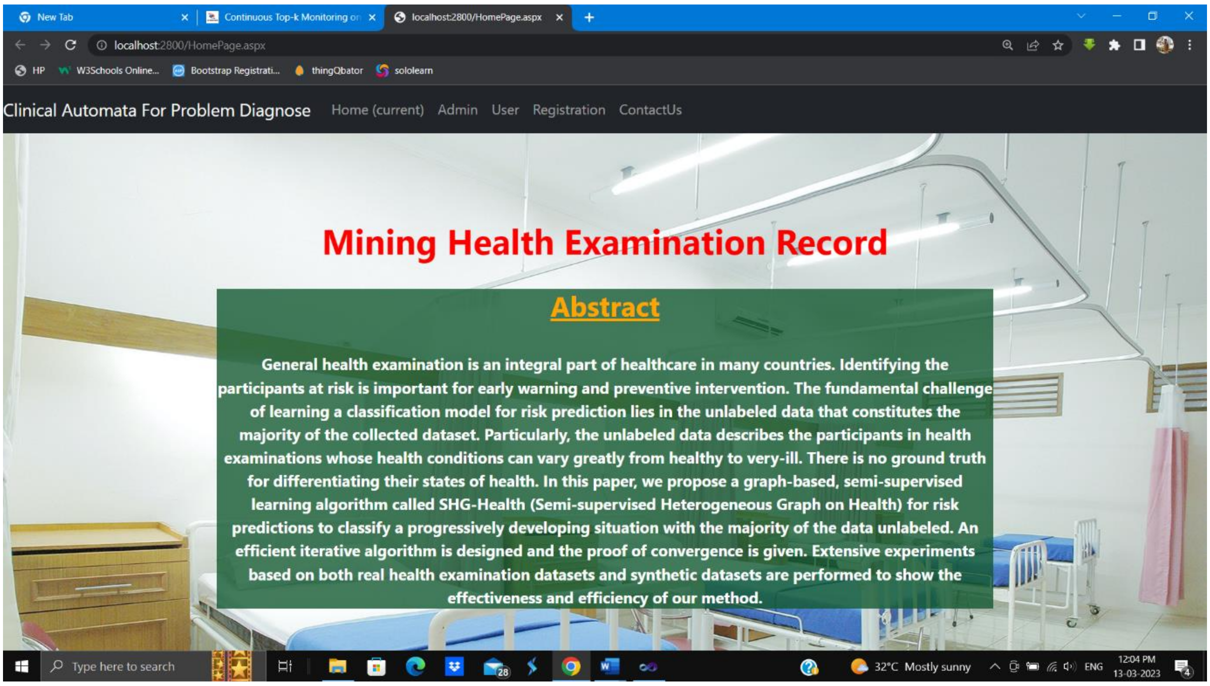
```
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

namespace Mining_Health
{
    public partial class UserViewDoctor :
System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs
e)
        {
            using (SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["Sqlc
onn"].ConnectionString))
            {
                using (SqlCommand cmd = new SqlCommand())
                {
                    cmd.Connection = con;
                    con.Open();
                    cmd.CommandText = "select * from
AddDoctor";

                    SqlDataReader dr =
cmd.ExecuteReader();
                    if (dr.Read())
                    {
                        GridView2.DataSource = dr;
                        GridView2.DataBind();
                    }
                    else
                    {

                    }
                }
            }
        }
    }
}
```

Home Page

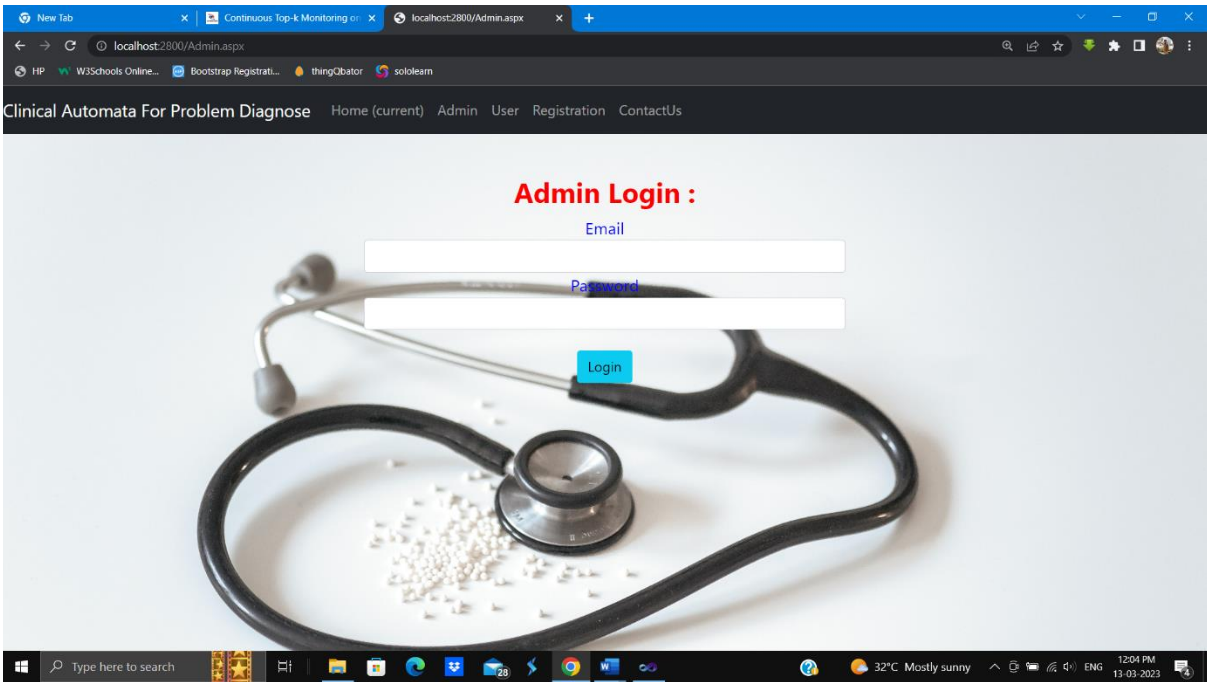


Mining Health Examination Record

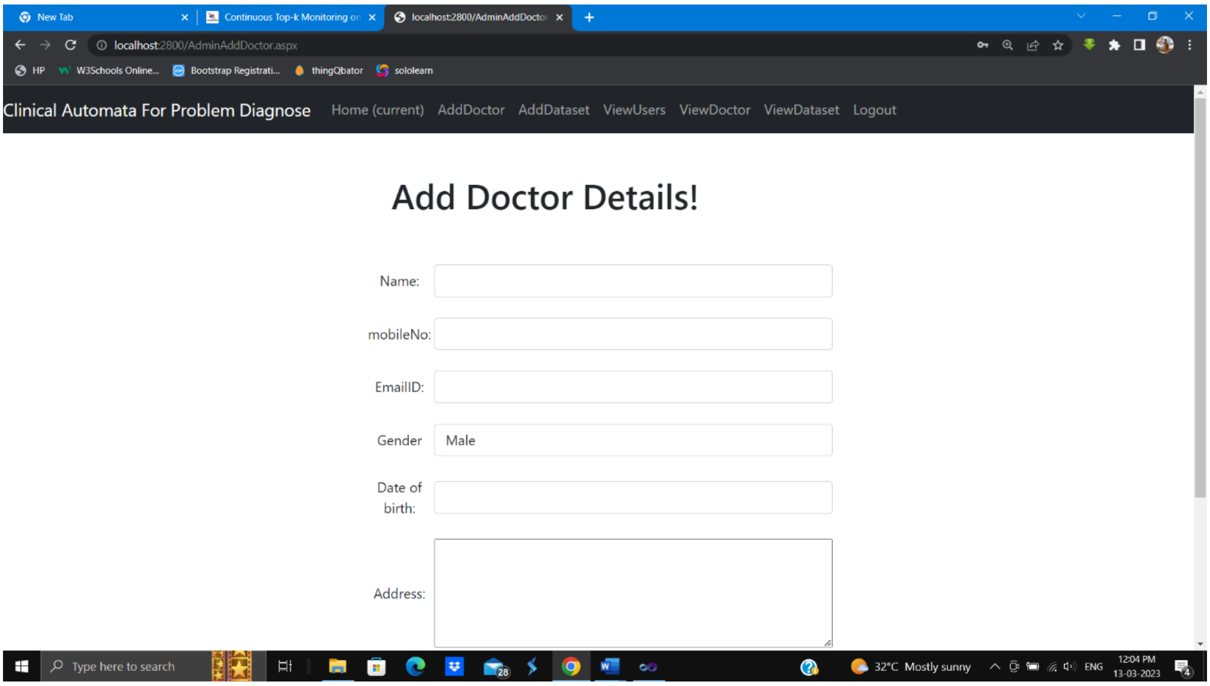
Abstract

General health examination is an integral part of healthcare in many countries. Identifying the participants at risk is important for early warning and preventive intervention. The fundamental challenge of learning a classification model for risk prediction lies in the unlabeled data that constitutes the majority of the collected dataset. Particularly, the unlabeled data describes the participants in health examinations whose health conditions can vary greatly from healthy to very-ill. There is no ground truth for differentiating their states of health. In this paper, we propose a graph-based, semi-supervised learning algorithm called SHG-Health (Semi-supervised Heterogeneous Graph on Health) for risk predictions to classify a progressively developing situation with the majority of the data unlabeled. An efficient iterative algorithm is designed and the proof of convergence is given. Extensive experiments based on both real health examination datasets and synthetic datasets are performed to show the effectiveness and efficiency of our method.

Admin page

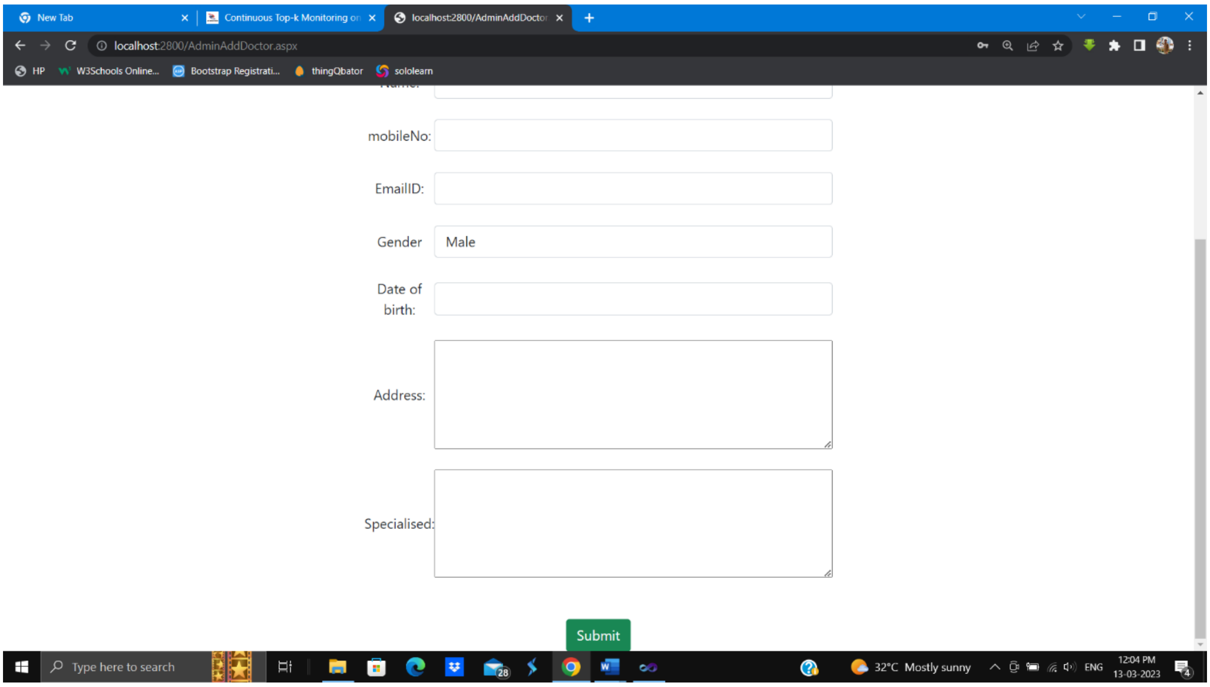


Add Doctor page



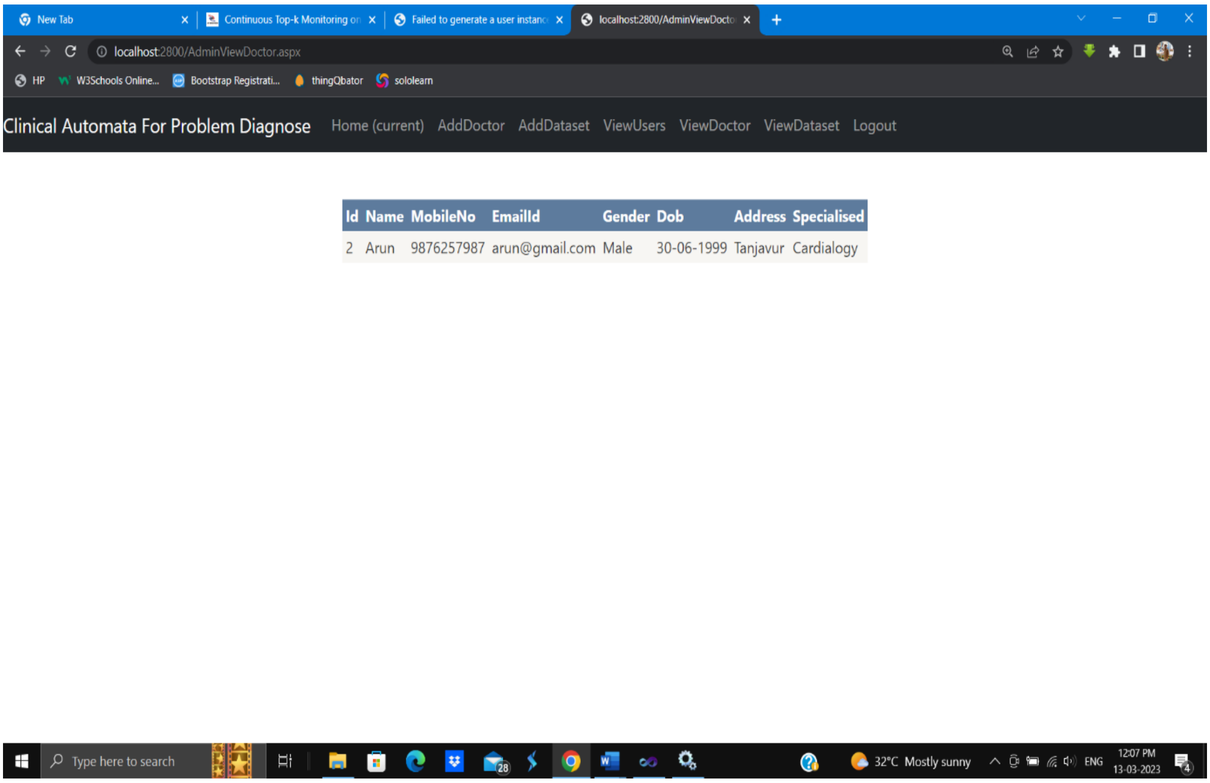
Add Doctor Details!

Name:	
mobileNo:	
EmailID:	
Gender	Male
Date of birth:	
Address:	



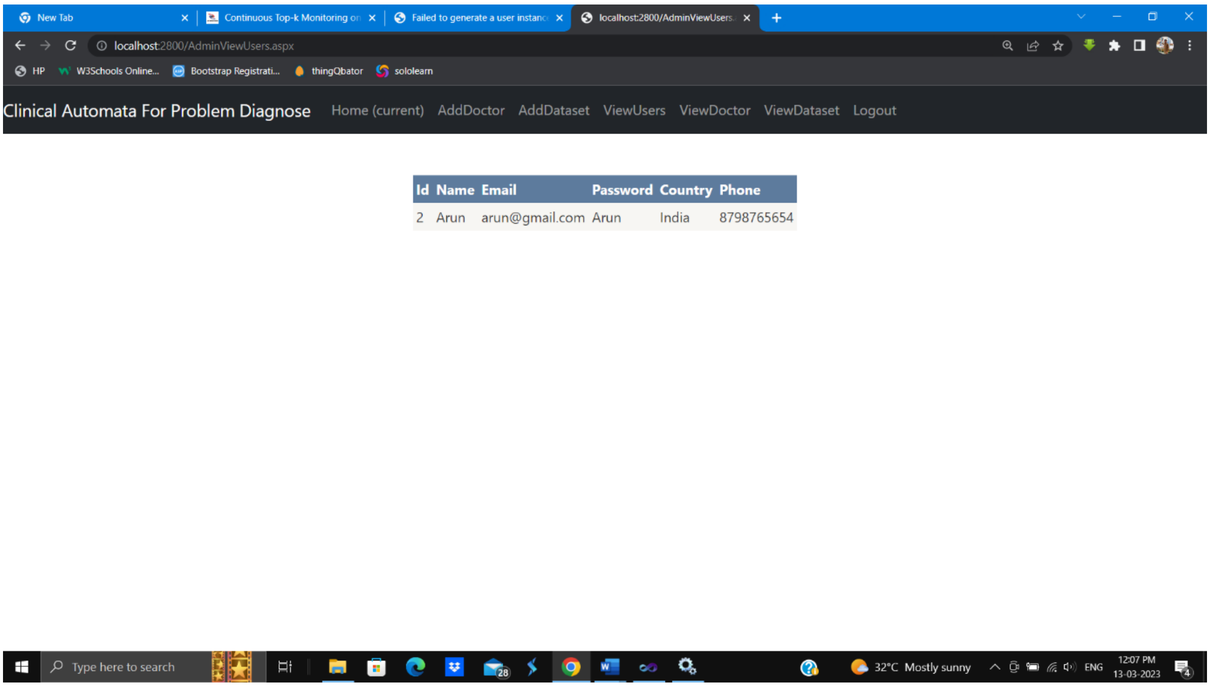
mobileNo:	
EmailID:	
Gender	Male
Date of birth:	
Address:	
Specialised:	

View Doctor


localhost:2800/AdminViewDoctor.aspx
Clinical Automata For Problem Diagnose
Home (current)
AddDoctor
AddDataset
ViewUsers
ViewDoctor
ViewDataset
Logout

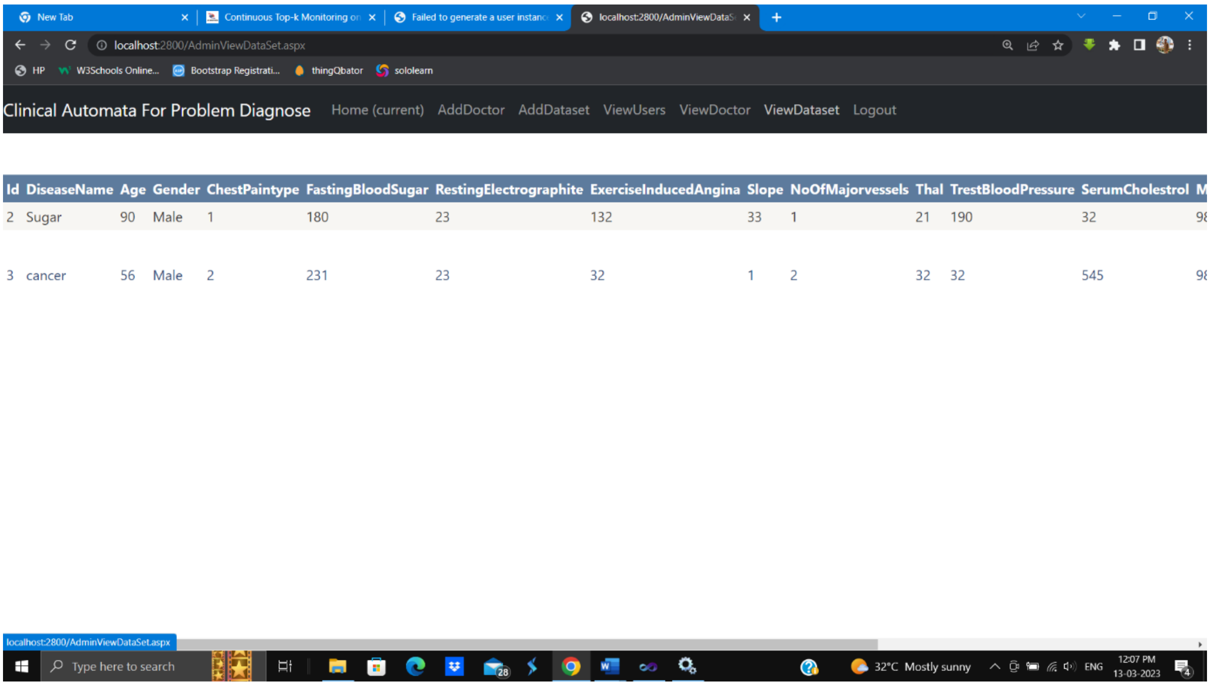
Id	Name	MobileNo	EmailId	Gender	Dob	Address	Specialised
2	Arun	9876257987	arun@gmail.com	Male	30-06-1999	Tanjavur	Cardialogy

View user


localhost:2800/AdminViewUsers.aspx
Clinical Automata For Problem Diagnose
Home (current)
AddDoctor
AddDataset
ViewUsers
ViewDoctor
ViewDataset
Logout

Id	Name	Email	Password	Country	Phone
2	Arun	arun@gmail.com	Arun	India	8798765654

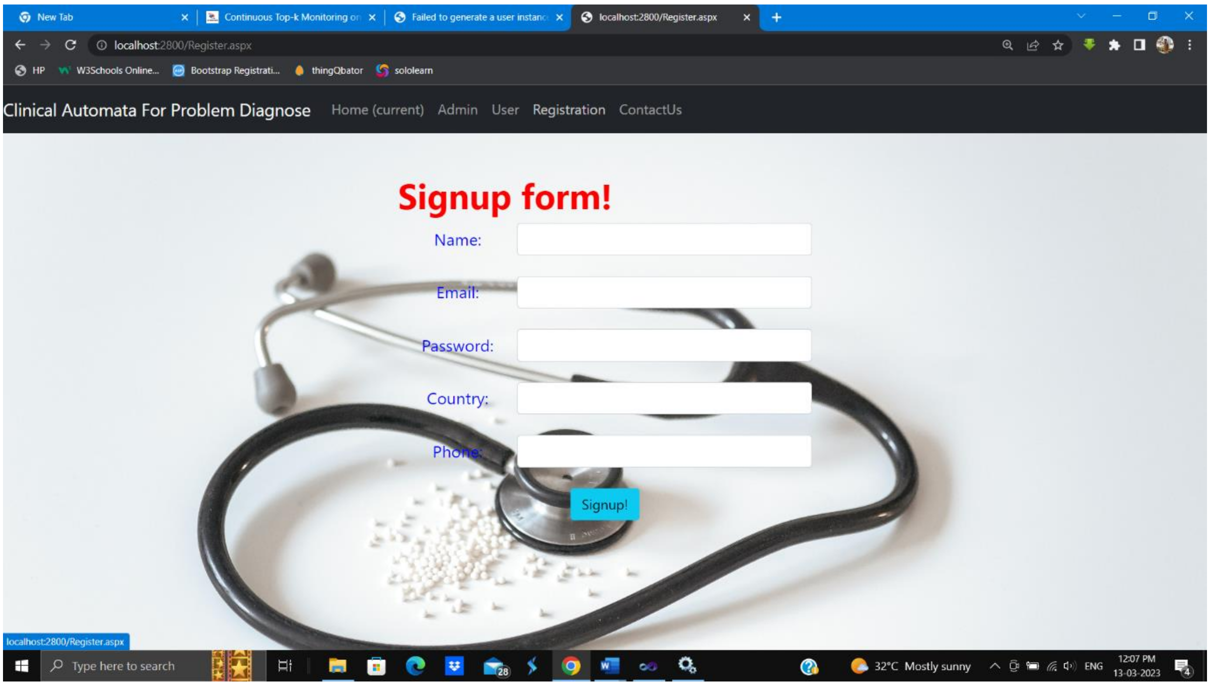
View Dataset


New Tab
Continuous Top-k Monitoring on
Failed to generate a user instanc
localhost:2800/AdminViewDataS
localhost:2800/AdminViewDataSet.aspx
HP
W3Schools Online...
Bootstrap Registrati...
thingQbator
sololearn
Type here to search
32°C Mostly sunny
ENG
12:07 PM
13-03-2023

Clinical Automata For Problem Diagnose Home (current) AddDoctor AddDataset ViewUsers ViewDoctor ViewDataset Logout

Id	DiseaseName	Age	Gender	ChestPaintype	FastingBloodSugar	RestingElectrographite	ExerciseInducedAngina	Slope	NoOfMajorvessels	Thal	TrestBloodPressure	SerumCholestrol	M
2	Sugar	90	Male	1	180	23	132	33	1	21	190	32	9
3	cancer	56	Male	2	231	23	32	1	2	32	32	545	9

Signup page



Clinical Automata For Problem Diagnose Home (current) Admin User Registration ContactUs

Signup form!

Name:

Email:

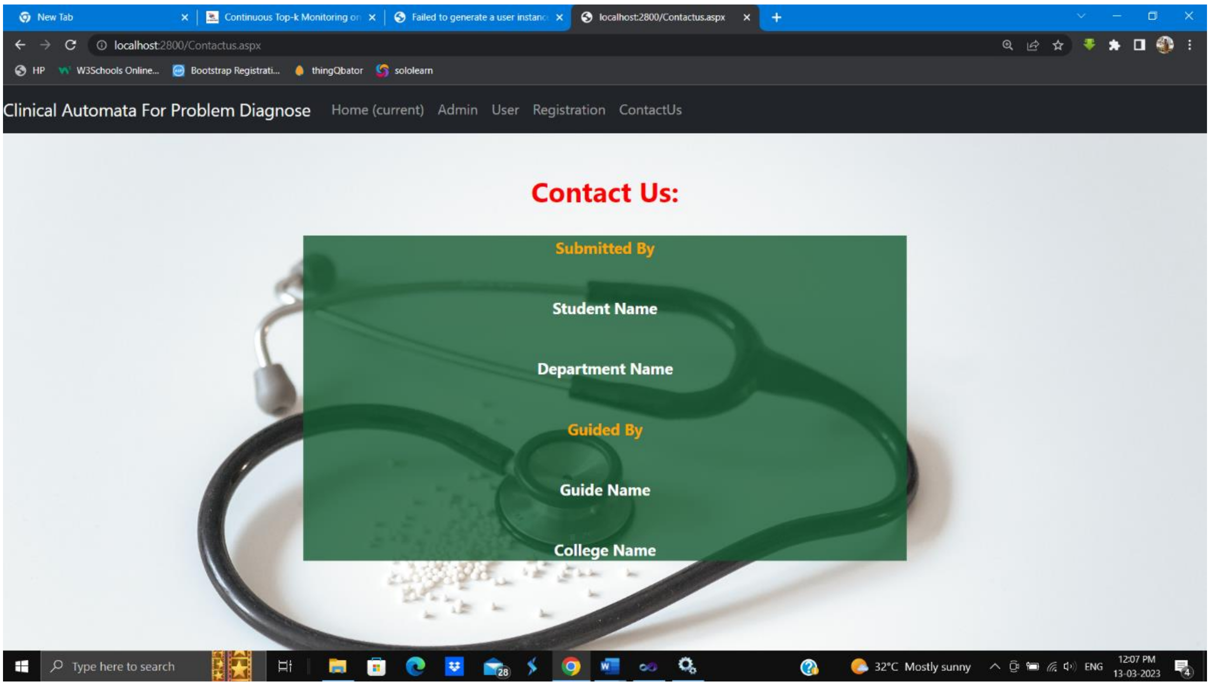
Password:

Country:

Phone:

Signup!

Contactus page


localhost:2800/Contactus.aspx
Clinical Automata For Problem Diagnose
Home (current)
Admin
User
Registration
ContactUs

Contact Us:

Submitted By

Student Name

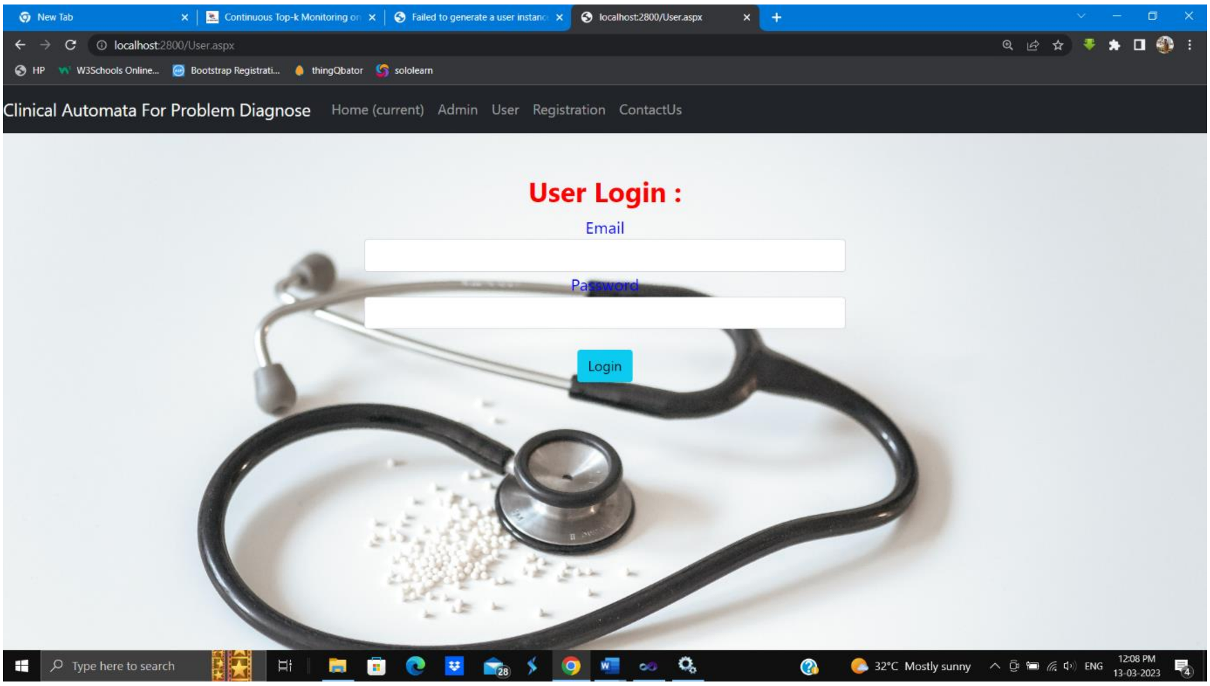
Department Name

Guided By

Guide Name

College Name

UserLogin page

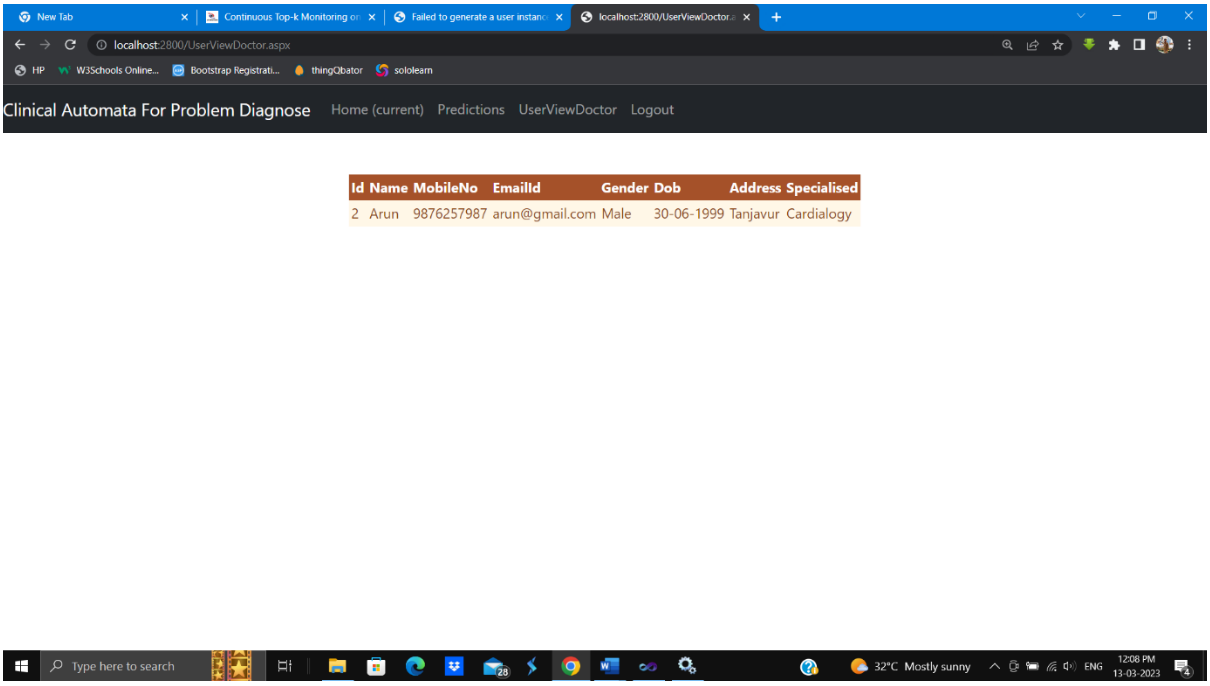


User Login :

Email

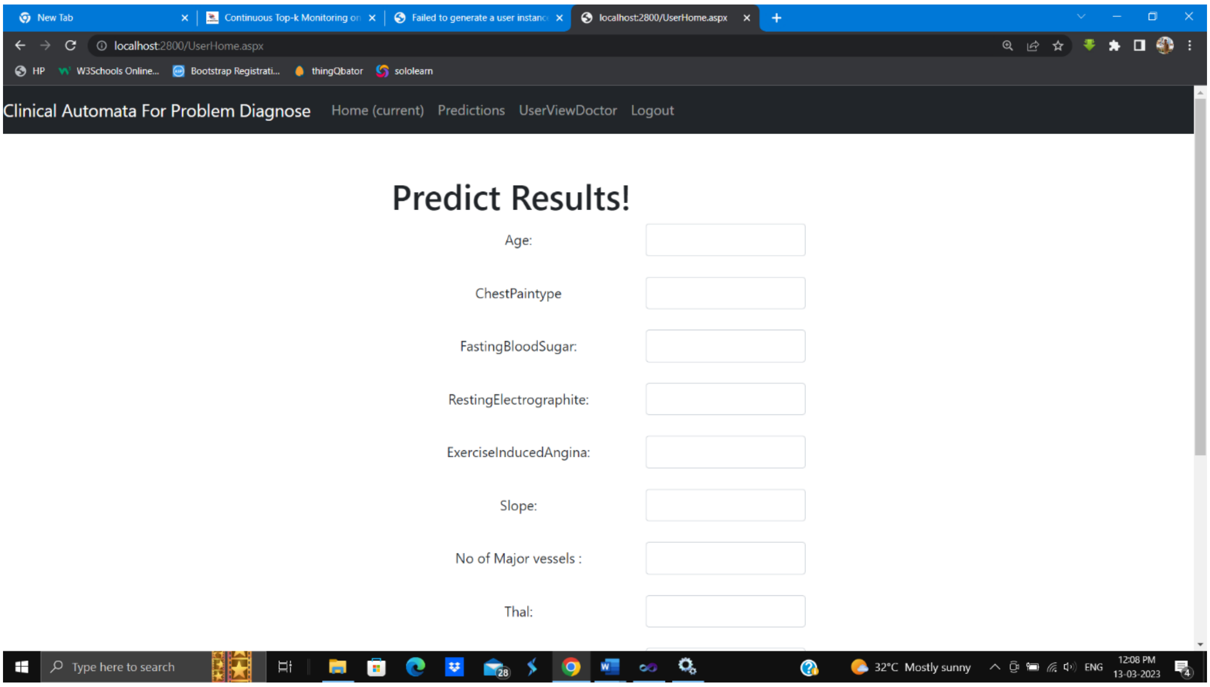
Password

View Doctor


localhost:2800/UserViewDoctor.aspx
Clinical Automata For Problem Diagnose
Home (current)
Predictions
UserViewDoctor
Logout

Id	Name	MobileNo	EmailId	Gender	Dob	Address	Specialised
2	Arun	9876257987	arun@gmail.com	Male	30-06-1999	Tanjavur	Cardiology

PredictResult Page


localhost:2800/UserHome.aspx
Clinical Automata For Problem Diagnose
Home (current)
Predictions
UserViewDoctor
Logout

Predict Results!

Age:

ChestPaintype

FastingBloodSugar:

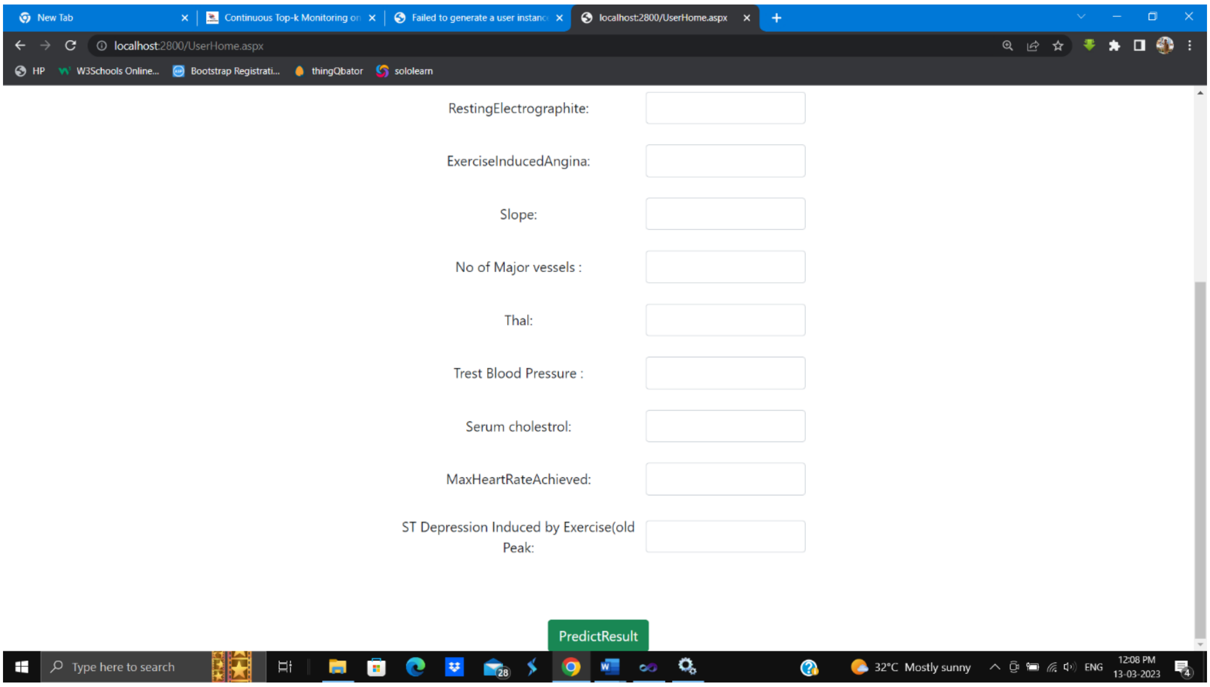
RestingElectrographite:

ExerciseInducedAngina:

Slope:

No of Major vessels :

Thal:



RestingElectrographite:

ExerciseInducedAngina:

Slope:

No of Major vessels :

Thal:

Trest Blood Pressure :

Serum cholestrol:

MaxHeartRateAchieved:

ST Depression Induced by Exercise(old Peak:

PredictResult

8. BIBLIOGRAPHY

Primary Sources

1 Database Management System - James Martin

2 Software Engineering - Pressman

3 The complete reference ASP.NET - Steven A Smith

4 ASP.NET Developers Cookbook - Rob Howard

5 Professional .NET Framework - Joe Duffy

6 SQL-Server 2008 - Chris Ryner