**ONLINE PSYCHOLOGICAL RESPONDER**

**ABSTRACT**

The project is entitled as “ONLINE PSCHOLOGICAL RESPONDER” which is developed in Java JSP as front end and Mysql as backend to store the information.

This project assists to students who face the psychological problems. In current system people does not have complete knowledge about the psychology. A student faces lots of problem in their education side and personal life. Parents will take their children to the hospital and has to take doctor’s appointment. But using this project, a student can logon to the web site and get the required details from the psychologists. Hence data retrieval can also be very easy and faster. If any person needs more information advice they have to visit doctor or some healthcare institute. But in some cases user cannot get proper information and solution to their problems. This project helps users to get rid of their problem by providing the prevention details. In this psychologists can enter the information to overcome their problems. User has to create register in the web application. There will be an authorized person in admin login. Psychologists have to register in application. Psychologist can use the application once admin will admit them to use it. Admin will admit the psychologist by checking their details. Users of this portal should register before login. There will be two portal like Private and Public. User can choose their own portal whether their problem has to be maintain in private manner or in public manner. If they choose public, 3rd party in the application also can view their case list and advice given by the Psychologists. The Psychologists will post the new information about the student’s problem regularly. So the students will have new updates and that helps to overcome their problems.

Students may have queries about their problem. This web portal helps students to communicate with Psychologists to clarify doubts from anywhere.

**EXISTING SYSTEM**

In existing system people does not have complete knowledge about the disease and problems. Patients have to come to the hospital and have to take Psychologists’ appointment. This may hard to patients to wait, to visit Psychologists. Patients cannot be health conscious because they don’t get any awareness about Psychological problem. They don’t get details about prevention. They have to present physical to get the advice for their problem. So there is a need to record health information in data repositories for immediate access to patients and to psychologicaldiseases which enables better and time-efficient medical decisions.

**DISADVANTAGE**

* This takes more time to get details about disease and treatment.
* Difficult to identify problems
* Patients doubts cannot be clarified
* Slow Process to get advice from the doctor
* Patient should present physically.

**PROPOSED SYSTEM**

In these Psychologists has separate login. They will post the updated new post in the web application. That the patients can have proper update. Student can login and enter their problem and get advice and the details of treatment should be following. Through this patient can get the information about their problem and its prevention. Student can select their portal whether their conversion with psychologists visible to the third person. Or else it has to been maintained secure. If they have any doubt they can post questions and the Psychologists of this website will answer the questions. This project easily identify students problem. Psychologists will post updated information and motivation post in the web application that helps the students to get motivated.

**ADVANTAGES**

* It creates awareness about Psychological problem
* Patients can easily clarify their doubts
* It consumes less time
* Safe and Secure change choose the own portal
* Can have a daily information and motivation.

**MODULES**

* User Registration
* Psychologists Login
* Posting Problems
* Respond to Student
* Update New Post

**USER REGISTRATION**

In this module, new user can register in the web application. Once the registration process is completed user can login to the web application and they can communicate with the Psychologists and get their advice without physical presentation in the hospital.

**PSYCHOLOGISTS LOGIN**

In this module, Psychologists can register in the web application. Once those details are confirmed by an authorized person in the application. Psychologists can log in to a web application to advise students. So that student can have proper advice and information to lead their life.

**POSTING PROBLEMS**

In this module, Student can login to the web application and they can post their problems in that. There will be two portal like Private and Public. User can select their portal and post their problems accordingly. If they choose private the conversion between the user and psychologists will be secure no one can view or read. They select public the advice given to the user from the psychologists can be view by third party also.

**RESPOND TO STUDENT**

In this module, Student can ask any question related to their problems and issues to Psychologists. Psychologists will view the students query and respond to their question in the web application. He/She will provide proper information and give advice to the particular student according to their problem and selected portal.

**UPDATE NEW POST**

In this module, The Psychologists will post the new information and motivated post about the student’s problem regularly to motivate them. So the students will have new updates and that helps to overcome their problems.

**SYSTEM REQUIREMENTS**

**HARDWARE REQUIREMENTS**

* + - Hard disk : 500 GB
    - RAM : 2Gb
    - Processor : Intel i3
    - Monitor : 17” Color monitor
    - Key board, Mouse : Multi media.

**SOFTWARE REQUIREMENTS**

* + - Front End : Java (JSP)
    - Back End : MYSQL
    - Operating System : Windows Family

**INTRODUCTION TO LANGUAGE:-**

**JSP:-**

JavaServer Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

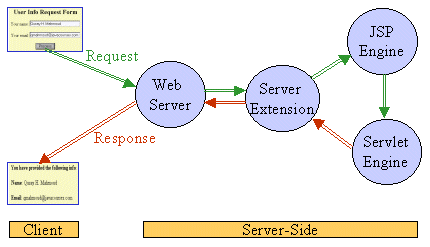
Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc.

**JSP INTRODUCTION:**

JSP stands for Java Server Pages. All your JSP pages are stored on the server.

1. The web server have JSP engine which acts as a container to process JSP pages.
2. All the requests for JSP Pages are intercepted by JSP Container.
3. JSP container along with web server provide the runtime environment to JSP.



**JSP PROCESSING:**

Please find the below steps that are required to process JSP Page –

1. Web browser sends an HTTP request to the web server requesting JSP page.
2. Web server recognizes that the HTTP request by web browser is for JSP page by checking the extension of the file (i.e .jsp)
3. Web server forwards HTTP Request to JSP engine.
4. JSP engine loads the JSP page from disk and converts it into a servlet
5. JSP engine then compiles the servlet into an executable class and forward original request to a servlet engine.
6. Servlet engine loads and executes the Servlet class.
7. Servlet produces an output in HTML format
8. Output produced by servlet engine is then passes to the web server inside an HTTP response.
9. Web server sends the HTTP response to Web browser in the form of static HTML content.
10. Web browser loads the static page into the browser and thus user can view the dynamically generated page.

**WHY USE JSP?**

JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI.

* Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files.
* JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
* JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc.
* JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of Java EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

**ADVANTAGES OF JSP:**

Following is the list of other advantages of using JSP over other technologies:

* **vs. Active Server Pages (ASP):** The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.
* **vs. Pure Servlets:** It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.
* **vs. Server-Side Includes (SSI):** SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.
* **vs. JavaScript:** JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

The web server needs a JSP engine ie. container to process JSP pages. The JSP container is responsible for intercepting requests for JSP pages. This tutorial makes use of Apache which has built-in JSP container to support JSP pages development.

A JSP container works with the Web server to provide the runtime environment and other services a JSP needs. It knows how to understand the special elements that are part of JSPs.

Following diagram shows the position of JSP container and JSP files in a Web Application.

**JSP PROCESSING:**

The following steps explain how the web server creates the web page using JSP:

* As with a normal page, your browser sends an HTTP request to the web server.
* The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine. This is done by using the URL or JSP page which ends with **.jsp**instead of .html.
* The JSP engine loads the JSP page from disk and converts it into a servlet content. This conversion is very simple in which all template text is converted to println( ) statements and all JSP elements are converted to Java code that implements the corresponding dynamic behavior of the page.
* The JSP engine compiles the servlet into an executable class and forwards the original request to a servlet engine.
* A part of the web server called the servlet engine loads the Servlet class and executes it. During execution, the servlet produces an output in HTML format, which the servlet engine passes to the web server inside an HTTP response.
* The web server forwards the HTTP response to your browser in terms of static HTML content.
* Finally web browser handles the dynamically generated HTML page inside the HTTP response exactly as if it were a static page.

All the above mentioned steps can be shown below in the following diagram:

Typically, the JSP engine checks to see whether a servlet for a JSP file already exists and whether the modification date on the JSP is older than the servlet. If the JSP is older than its generated servlet, the JSP container assumes that the JSP hasn't changed and that the generated servlet still matches the JSP's contents. This makes the process more efficient than with other scripting languages (such as PHP) and therefore faster.

So in a way, a JSP page is really just another way to write a servlet without having to be a Java programming wiz. Except for the translation phase, a JSP page is handled exactly like a regular servlet

The key to understanding the low-level functionality of JSP is to understand the simple life cycle they follow.

A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.

The following are the paths followed by a JSP

* Compilation
* Initialization
* Execution
* Cleanup

The four major phases of JSP life cycle are very similar to Servlet Life Cycle and they are as follows:

**JSP Compilation:**

When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

* Parsing the JSP.
* Turning the JSP into a servlet.
* Compiling the servlet.

**JSP Initialization:**

When a container loads a JSP it invokes the jspInit() method before servicing any requests. If you need to perform JSP-specific initialization, override the jspInit() method:

public void jspInit(){

// Initialization code...

}

Typically initialization is performed only once and as with the servlet init method, you generally initialize database connections, open files, and create lookup tables in the jspInit method.

**JSP Execution:**

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed.

Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the **\_jspService()** method in the JSP.

The \_jspService() method takes an **HttpServletRequest** and an **HttpServletResponse**as its parameters as follows:

void \_jspService(HttpServletRequest request,

HttpServletResponse response)

{

// Service handling code...

}

The \_jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods ie. GET, POST, DELETE etc.

**JSP Cleanup:**

The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The **jspDestroy()** method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.

The jspDestroy() method has the following form:

public void jspDestroy()

{

// Your cleanup code goes here.

}

**MySQL**

**MySQL** is (as of July 2013) the world's second most widely used [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS) and most widely used open-source RDBMS. It is named after co-founder [Michael Widenius](http://en.wikipedia.org/wiki/Michael_Widenius)'s daughter, my. The [SQL](http://en.wikipedia.org/wiki/SQL) acronym stands for [Structured Query Language](http://en.wikipedia.org/wiki/Structured_Query_Language).

The MySQL development project has made its [source code](http://en.wikipedia.org/wiki/Source_code) available under the terms of the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License), as well as under a variety of [proprietary](http://en.wikipedia.org/wiki/Proprietary_software) agreements. MySQL was owned and sponsored by a single [for-profit](http://en.wikipedia.org/wiki/Business) firm, the [Swedish](http://en.wikipedia.org/wiki/Sweden) company [**MySQL AB**](http://en.wikipedia.org/wiki/MySQL_AB)**,** now owned by [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29) open source web application software stack (and other ['AMP'](http://en.wikipedia.org/wiki/List_of_AMP_packages) stacks). LAMP is an acronym for "[Linux](http://en.wikipedia.org/wiki/Linux), [Apache](http://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](http://en.wikipedia.org/wiki/Perl)/[PHP](http://en.wikipedia.org/wiki/PHP)/[Python](http://en.wikipedia.org/wiki/Python_%28programming_language%29)." [Free-software](http://en.wikipedia.org/wiki/Free_software)-open source projects that require a full-featured database management system often use MySQL.

For proprietary use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: [TYPO3](http://en.wikipedia.org/wiki/TYPO3), [MODx](http://en.wikipedia.org/wiki/MODx), [Joomla](http://en.wikipedia.org/wiki/Joomla), [WordPress](http://en.wikipedia.org/wiki/WordPress), [phpBB](http://en.wikipedia.org/wiki/PhpBB), [MyBB](http://en.wikipedia.org/wiki/MyBB), [Drupal](http://en.wikipedia.org/wiki/Drupal) and other software. MySQL is also used in many high-profile, large-scale [websites](http://en.wikipedia.org/wiki/Website), including [Google](http://en.wikipedia.org/wiki/Google)[[13]](http://en.wikipedia.org/wiki/MySQL#cite_note-mysqlatgoogle-14)[[14]](http://en.wikipedia.org/wiki/MySQL#cite_note-15) (though not for searches), [Face book](http://en.wikipedia.org/wiki/Facebook), [Twitter](http://en.wikipedia.org/wiki/Twitter), [Flickr](http://en.wikipedia.org/wiki/Flickr), and [YouTube](http://en.wikipedia.org/wiki/YouTube).

## Interfaces

MySQL is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS), and ships with no [GUI](http://en.wikipedia.org/wiki/Graphical_user_interface) tools to administer MySQL databases or manage data contained within the databases. Users may use the included [command line](http://en.wikipedia.org/wiki/Command_line) tools, or use MySQL "front-ends", desktop software and web applications that create and manage MySQL databases, build database structures, back up data, inspect status, and work with data records. The official set of MySQL front-end tools, [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is actively developed by Oracle, and is freely available for use.

### Graphical

The official [MySQL Workbench](http://en.wikipedia.org/wiki/MySQL_Workbench) is a free integrated environment developed by MySQL AB, that enables users to graphically administer MySQL databases and visually design database structures. MySQL Workbench replaces the previous package of software, [MySQL GUI Tools](http://en.wikipedia.org/wiki/MySQL_GUI_Tools). Similar to other third-party packages, but still considered the authoritative MySQL front end, MySQL Workbench lets users manage database design & modeling, SQL development (replacing MySQL Query Browser) and Database administration (replacing MySQL Administrator).

MySQL Workbench is available in two editions, the regular [free and open source](http://en.wikipedia.org/wiki/Free_and_open_source_software) *Community Edition* which may be downloaded from the MySQL website, and the proprietary *Standard Edition* which extends and improves the feature set of the Community Edition.

Third-party proprietary and free graphical administration applications (or "front ends") are available that integrate with MySQL and enable users to work with database structure and data visually. Some well-known front ends, in alphabetical order, are:

* [Adminer](http://en.wikipedia.org/wiki/Adminer) – a free MySQL front end written in one [PHP](http://en.wikipedia.org/wiki/PHP) script, capable of managing multiple databases, with many [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets) skins available
* [Chive](http://en.wikipedia.org/wiki/Chive) a free, open source, web-based database management tool designed as an alternative to phpMyAdmin
* [Database Workbench](http://en.wikipedia.org/wiki/Database_Workbench) – a software application for development and administration of multiple relational databases including MySQL, with interoperationality between different database systems
* [DBEdit](http://en.wikipedia.org/wiki/DBEdit) – a free front end for MySQL and other databases
* [HeidiSQL](http://en.wikipedia.org/wiki/HeidiSQL) – a full featured free front end that runs on [Windows](http://en.wikipedia.org/wiki/Windows), and can connect to local or remote MySQL servers to manage databases, tables, column structure, and individual data records. Also supports specialised GUI features for date/time fields and enumerated multiple-value fields.
* [LibreOffice Base](http://en.wikipedia.org/wiki/LibreOffice) – LibreOffice Base allows the creation and management of databases, preparation of forms and reports that provide end users easy access to data. Like [Microsoft Access](http://en.wikipedia.org/wiki/Microsoft_Access), it can be used as a front-end for various database systems, including Access databases (JET), ODBC data sources, and MySQL or [PostgreSQL](http://en.wikipedia.org/wiki/PostgreSQL).
* [Navicat](http://en.wikipedia.org/wiki/Navicat) – a series of proprietary graphical database management applications, developed for Windows, Macintosh and Linux
* [OpenOffice.org](http://en.wikipedia.org/wiki/OpenOffice.org) – freely available [OpenOffice.org Base](http://en.wikipedia.org/wiki/OpenOffice.org_Base) can manage MySQL databases if the entire suite is installed
* [phpMyAdmin](http://en.wikipedia.org/wiki/PhpMyAdmin) – a free Web-based front-end, widely installed by [web hosting services](http://en.wikipedia.org/wiki/Web_hosting_service) since it is developed in PHP and included in the LAMP stack, and [MAMP](http://en.wikipedia.org/wiki/MAMP), [XAMPP](http://en.wikipedia.org/wiki/XAMPP) and [WAMP](http://en.wikipedia.org/wiki/WAMP_%28software_bundle%29) software bundle installers
* [SQLBuddy](http://en.wikipedia.org/wiki/SQLBuddy) – a free Web-based front end, developed in PHP
* [SQLyog](http://en.wikipedia.org/wiki/SQLyog) – proprietary, but there is also a free 'community' edition available
* [Toad for MySQL](http://en.wikipedia.org/wiki/TOAD_%28software%29) – a free development and administration front end for MySQL from [Dell Software](http://en.wikipedia.org/wiki/Dell_Software)
* [Webmin](http://en.wikipedia.org/wiki/Webmin) – a free Web-based management utility and a MySQL front end, developed in Perl with some parts written in Java

Other available proprietary MySQL front ends include [dbForge Studio for MySQL](http://en.wikipedia.org/w/index.php?title=DbForge_Studio_for_MySQL&action=edit&redlink=1), DBStudio, [Epictetus](http://en.wikipedia.org/wiki/Epictetus_Database_Client), [Microsoft Access](http://en.wikipedia.org/wiki/Microsoft_Access), [Oracle SQL Developer](http://en.wikipedia.org/wiki/Oracle_SQL_Developer), SchemaBank, [SQLPro SQL Client](http://en.wikipedia.org/wiki/SQLPro_SQL_Client), [Toad Data Modeler](http://en.wikipedia.org/wiki/Toad_Data_Modeler) and [DaDaBIK](http://en.wikipedia.org/wiki/DaDaBIK).

### Command line

MySQL ships with many [command line](http://en.wikipedia.org/wiki/Command_line) tools, from which the main interface is 'mysql' client. Third parties have also developed tools to manage MySQL servers.

* MySQL Utilities – a set of utilities designed to perform common maintenance and administrative tasks. Originally included as part of the MySQL Workbench, the utilities are now a stand-alone download available from Oracle.
* Percona Toolkit – a cross-platform toolkit for MySQL, developed in [Perl](http://en.wikipedia.org/wiki/Perl).[[31]](http://en.wikipedia.org/wiki/MySQL#cite_note-32) Percona Toolkit can be used to prove replication is working correctly, fix corrupted data, automate repetitive tasks, and speed up servers. Percona Toolkit is included with several [Linux](http://en.wikipedia.org/wiki/Linux) distributions such as [CentOS](http://en.wikipedia.org/wiki/CentOS) and [Debian](http://en.wikipedia.org/wiki/Debian), and packages are available for [Fedora](http://en.wikipedia.org/wiki/Fedora_%28operating_system%29) and [Ubuntu](http://en.wikipedia.org/wiki/Ubuntu_%28operating_system%29) as well. Percona Toolkit was originally developed as Maatkit, but as of late 2011, Maatkit is no longer developed.

**SYSTEM DESIGN**

INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system. In the project, the input design is made in various windows forms with various methods.

For example, in the login form, the empty username and password is not allowed. The username if exists in the database, the input is considered to be invalid and is not accepted.

Input design contains several forms there are,

Login

Register

Add problem

OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

In the project, the output is to

View user details

View problem

status

**DATABASE DESIGN:**

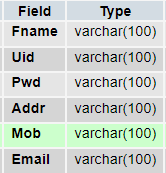
The database design is a must for any application developed especially more for the data store projects. Since the chatting method involves storing the message in the table and produced to the administrator, proper handling of the table is a must. In the project, login table is designed to be unique in accepting the username and the length of the username and password should be greater than zero. In this database design the admin details and the user details are stored in the database. The different users view the data in different format according to the privileges given.

The complete listing of the tables and their fields are provided in the annexure under the title ‘Table Structure’.

Tables

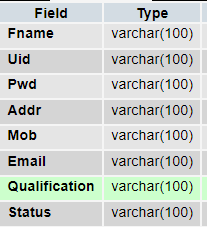
User tables

Primary key:Uid



Psychologiststbl

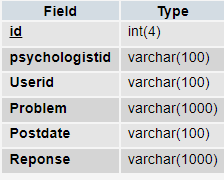
Primary key:Uid



Post problem

Primary key:iid

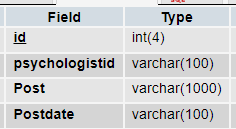
Foreign key:userid, Psychologistid



New post

Primary key:iid

Foreign key: Psychologistid



Testing

**INTRODUCTION:**

The most important phase in system development life cycle is system testing. The number and nature of errors in a newly designed system depends on the system specifications and the time frame given for the design.

A newly designed system should have all the subsystems working together, but in reality each subsystems work independently. During this phase, all the subsystems are gathered into one pool and tested to determine whether it meets the user requirements.

Testing is done at two level -Testing of individual modules and testing the entire system. During the system testing, the system is used experimentally to ensure that the software will run according to the specifications and in the way the user expects. Each test case is designed with the intent of finding errors in the way the system will process it.

Testing plays a very critical role in determining the reliability and efficiency of software and hence is a very important stage in software development. Software testing is done at different levels. They are the unit testing and system testing which comprises of integration testing and acceptance testing.

**TYPES OF TESTING**

**Unit Testing**

This is the first level of testing. The different modules are tested against the specifications produced during the integration. This is done to test the internal logic of each module. Those resulting from the interaction between modules are initially avoided. The input received and output generated is also tested to see whether it falls in the expected range of values. Unit testing is performed from the bottom up, starting with the smallest and lowest modules and proceeding one at a time.

The units in a system are the modules and routines that are assembled and integrated to perform a specific function. The programs are tested for correctness of logic applied and detection of errors in coding. Each of the modules was tested and errors are rectified. They were then found to function properly.

**Integration Testing**

In integration testing, the tested modules are combined into sub-systems, which are then tested. The goal of integration testing to check whether the modules can be integrated properly emphasizing on the interfaces between modules. The different modules were linked together and integration testing done on them.

**Validation Testing**

The objective of the validation test is to tell the user about the validity and reliability of the system. It verifies whether the system operates as specified and the integrity of important data is maintained. User motivation is very important for the successful performance of the system.

All the modules were tested individually using both test data and live data. After each module was ascertained that it was working correctly and it had been "integrated" with the system. Again the system was tested as a whole. We hold the system tested with different types of users. The System Design, Data Flow Diagrams, procedures etc. were well documented so that the system can be easily maintained and upgraded by any computer professional at a later

**System Testing**

The integration of each module in the system is checked during this level of testing. The objective of system testing is to check if the software meets its requirements. System testing is done to uncover errors that were not found in earlier tests. This includes forced system failures and validation of total system as the user in the operational environment implements it. Under this testing, low volumes of transactions are generally based on live data. This volume is increased until the maximum level for each transactions type is reached. The total system is also tested for recovery after various major failures to ensure that no data are lost during the breakdown.

**SYSTEM IMPLEMENTATION**

Implementation is the most crucial stage in achieving a successful system and giving the user’s confidence that the new system is effective and workable. Implementation of this project refers to the installation of the package in its real environment to the full satisfaction of the users and operations of the system.

Testing is done individually at the time of development using the data and verification is done the way specified in the program specification. In short, implementation constitutes all activities that are required to put an already tested and completed package into operation. The success of any information system lies in its successful implementation.

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

The project execution was checked with live environment and the user requirements are satisfied. Proper implementation is essential to provide a reliable system to meet the organization requirements.

**CONCLUSION**

It is concluded that the application works well and satisfy the end users. The application is tested very well and errors are properly debugged. The application is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested.

This system is user friendly so everyone can use easily. Proper documentation is provided. The end user can easily understand how the whole system is implemented by going through the documentation. The system is tested, implemented and the performance is found to be satisfactory. All necessary output is generated. Thus, the project is completed successfully.

Further enhancements can be made to the application, so that the application functions very attractive and useful manner than the present one. The speed of the transactions become more enough now.

**Scope of Future Development**

There is scope for future development of this project. The world of computer fields is not static; it is always subject to be dynamic. The technology which is famous today becomes outdated the very next day. To keep abstract of technical improvements, the system may be further refined. So, it is not concluded. Yet it will improve with further enhancements.

Enhancements can be done in an efficient manner. We can even update the same with further modification establishment and can be integrated with minimal modification. Thus the project is flexible and can be enhanced at anytime with more advanced features.

**BIBLIOGRAPHY**

1. “Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam “, by Bryan Basham (Author), Kathy Sierra (Author), Bert Bates (Author), O'Reilly Media; Second Edition edition (April 1, 2008).2. “Core Java™, Volume I--Fundamentals (8th Edition) “ , by Cay S. Horstmann, Prentice Hall; 8 edition (April 18, 2008).3. .JavaServer Pages   **By: SAMS, Techmedia.**

4. System Analysis & Design     By: Ilias M Awad

**Web references**

<http://www.jsp.net/>

<http://www.tutorialspoint.com/mysql/>

<http://www.javatpoint.com/java-tutorial>

http://www.w3schools.in/java-tutorial/

Data Flow Diagram

Level 0

User

Psychologist

Database

Level 1

Psychologist

Problem

New post

Psychologisttbl

User

Level 2

User

User table

newpost

Problem

**ER Diagram**

Admin

Login

Enter

Students Details

Psychologiststbl

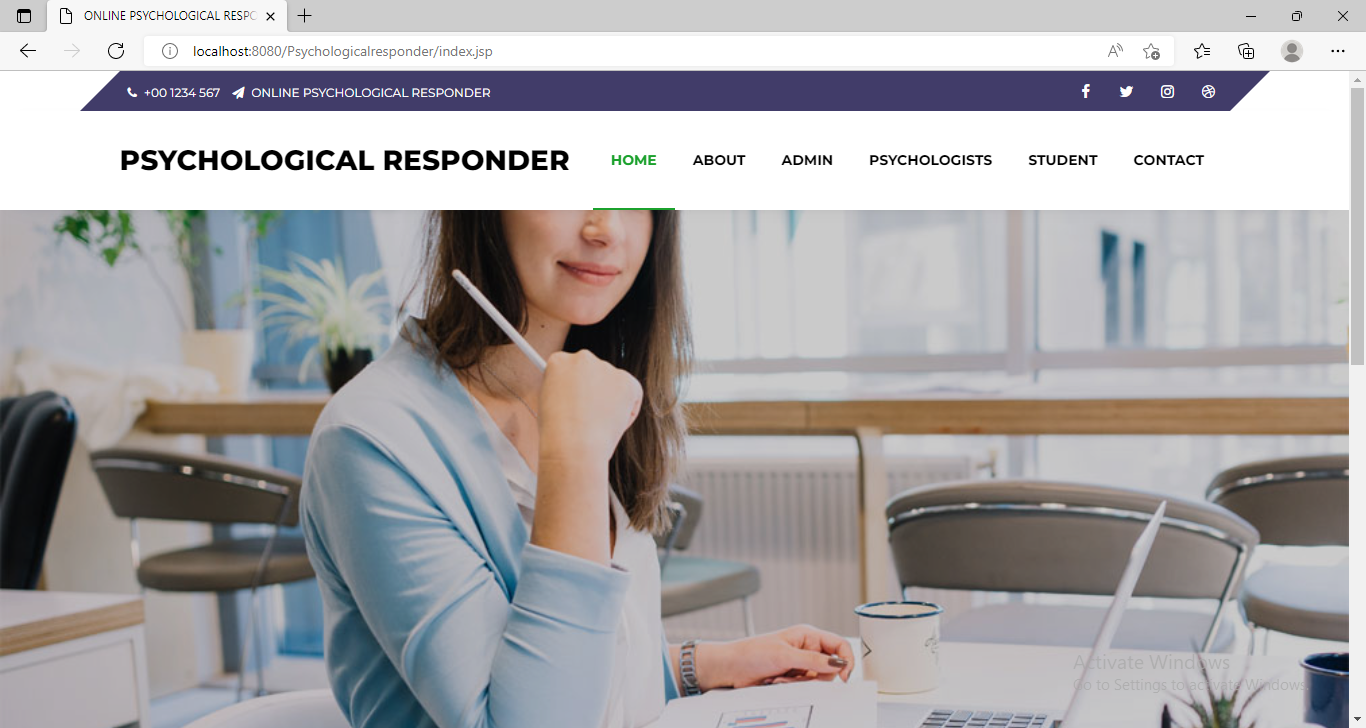
View

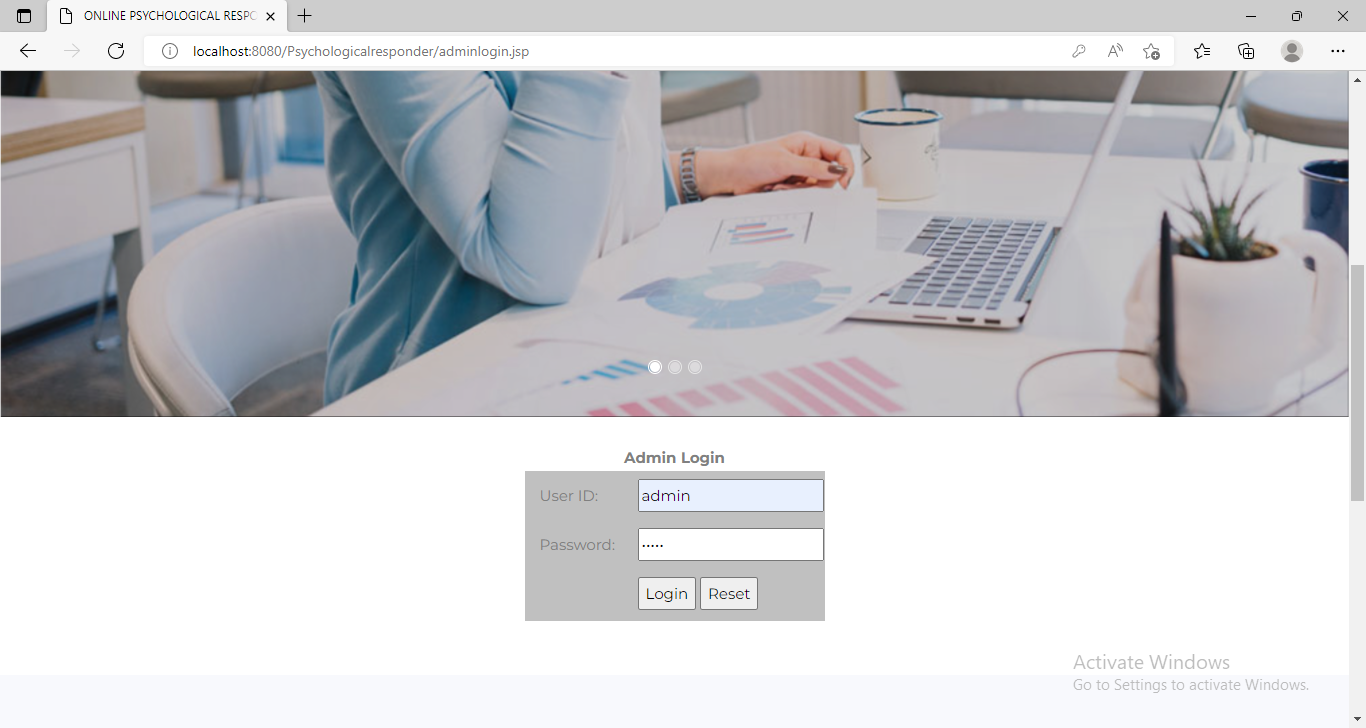
Problem

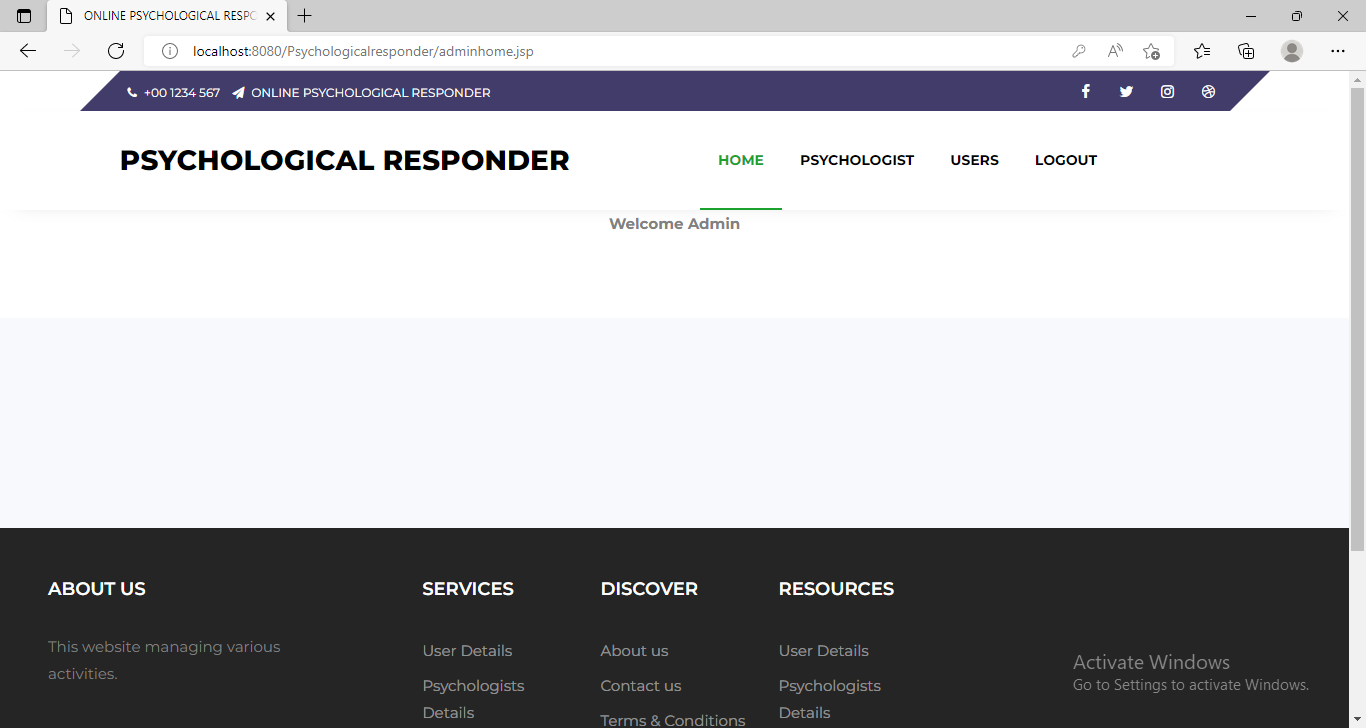
Post problem

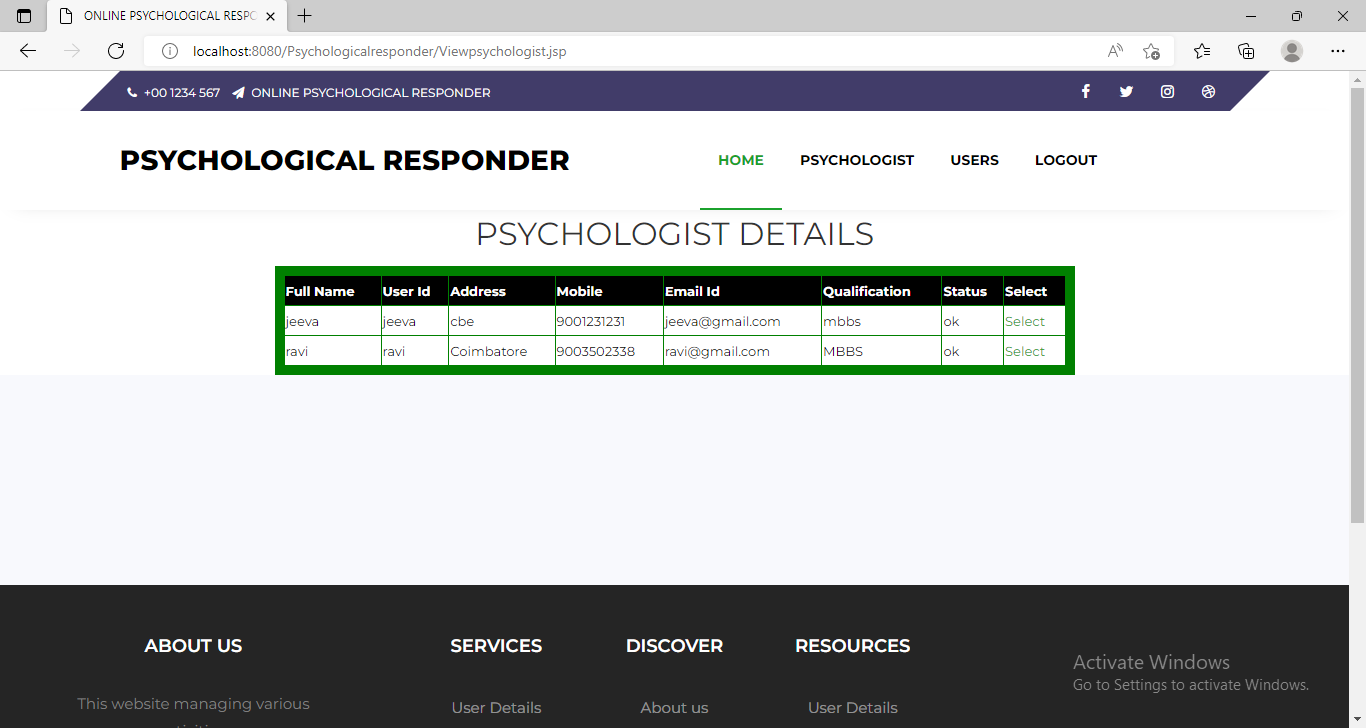
New post

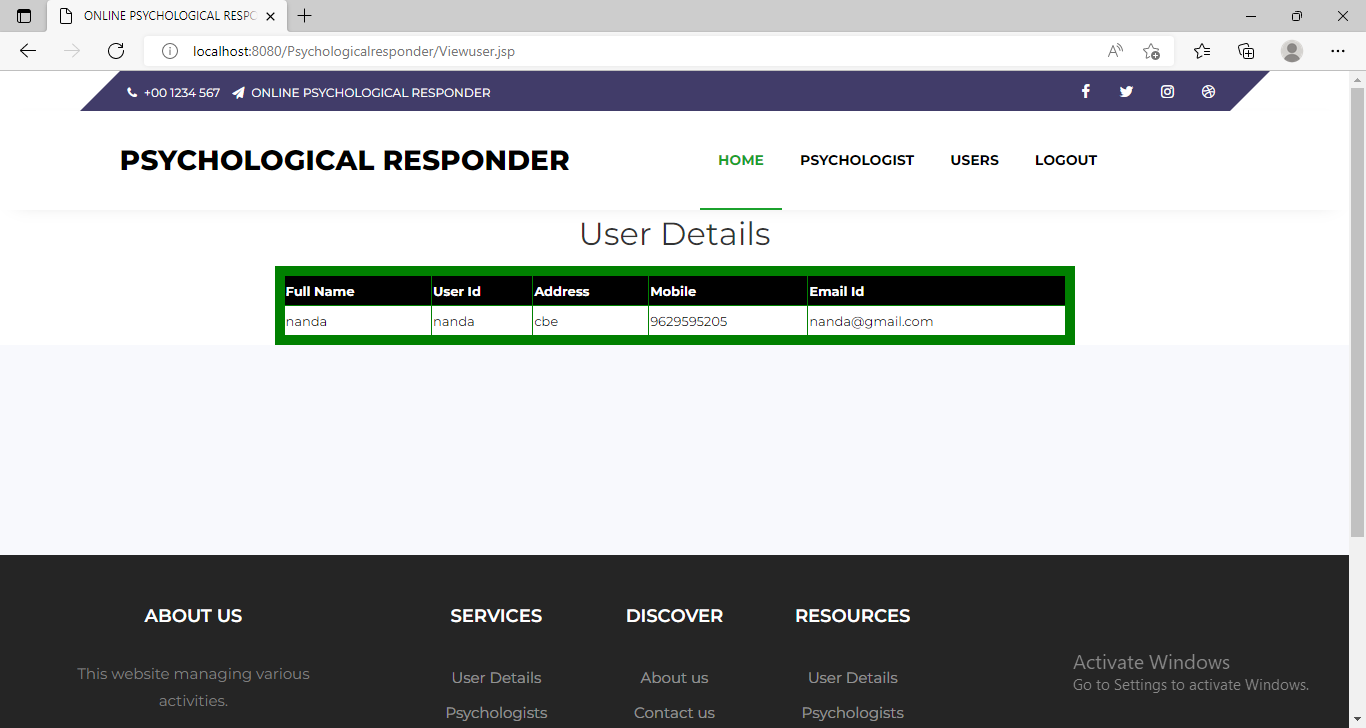
Screens

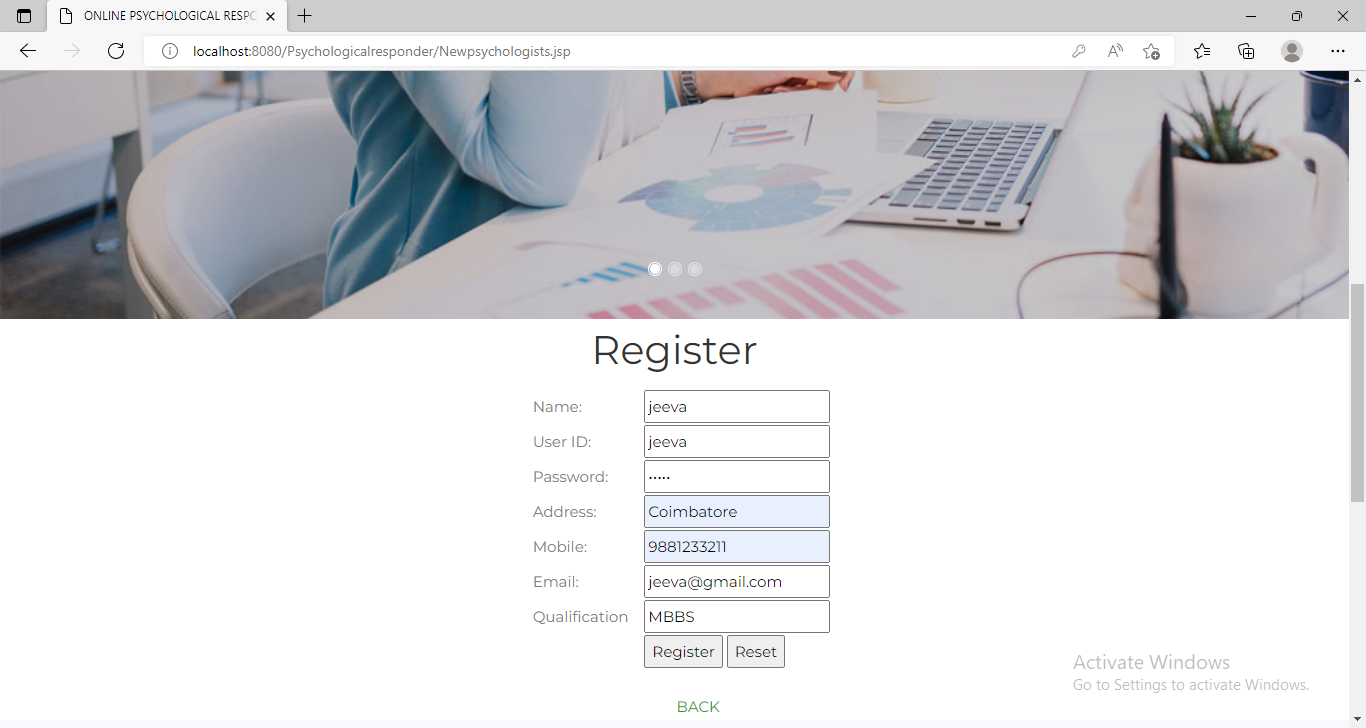


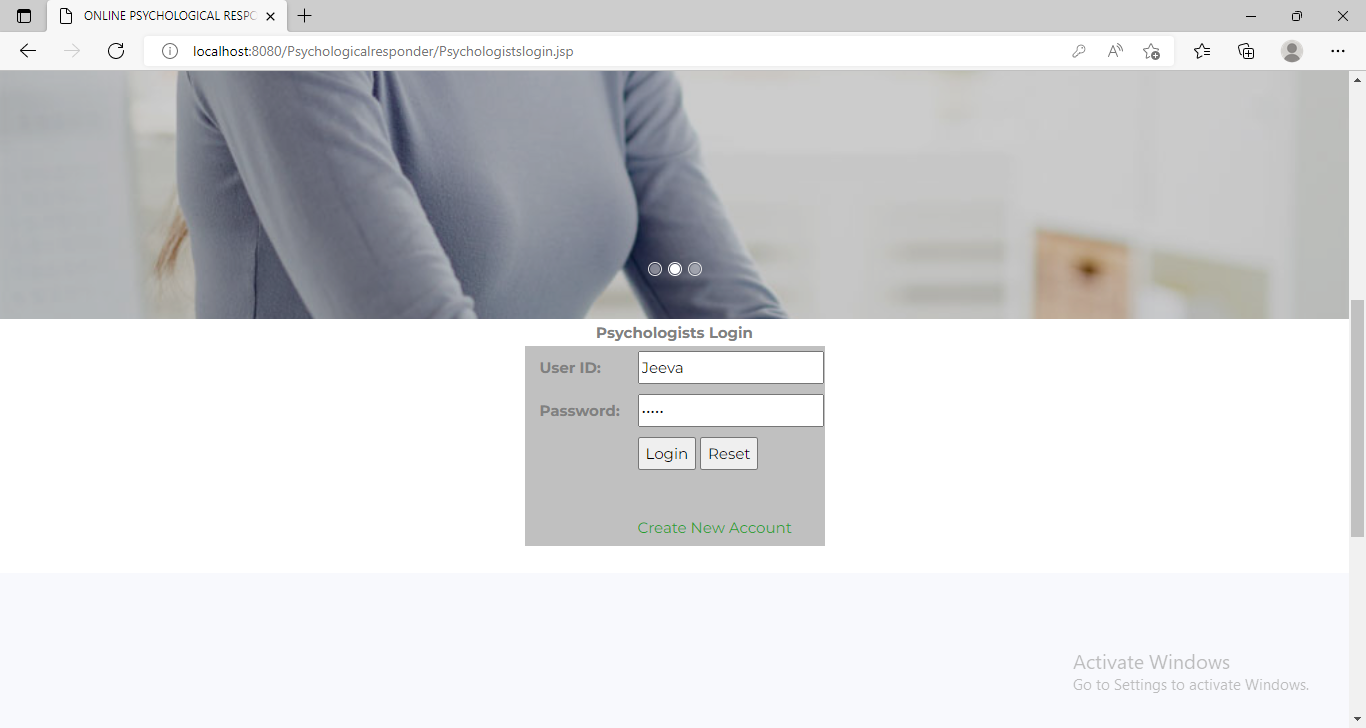


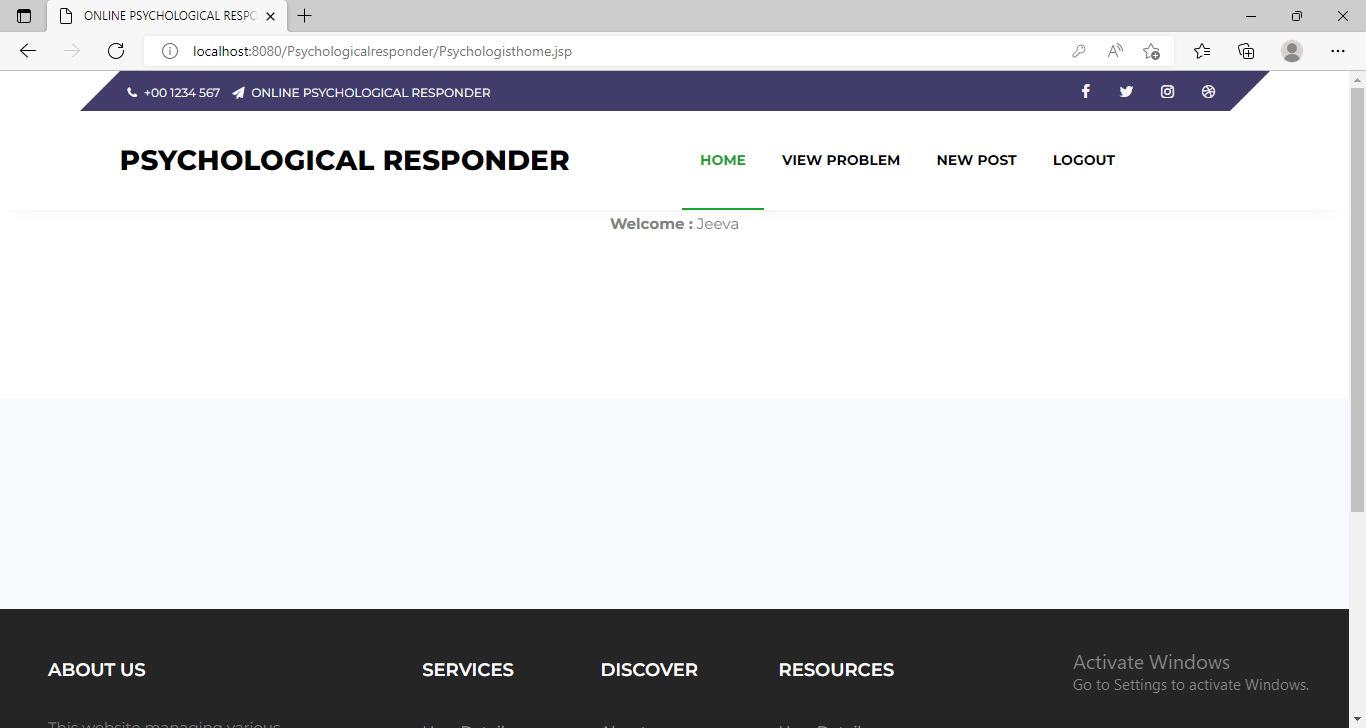


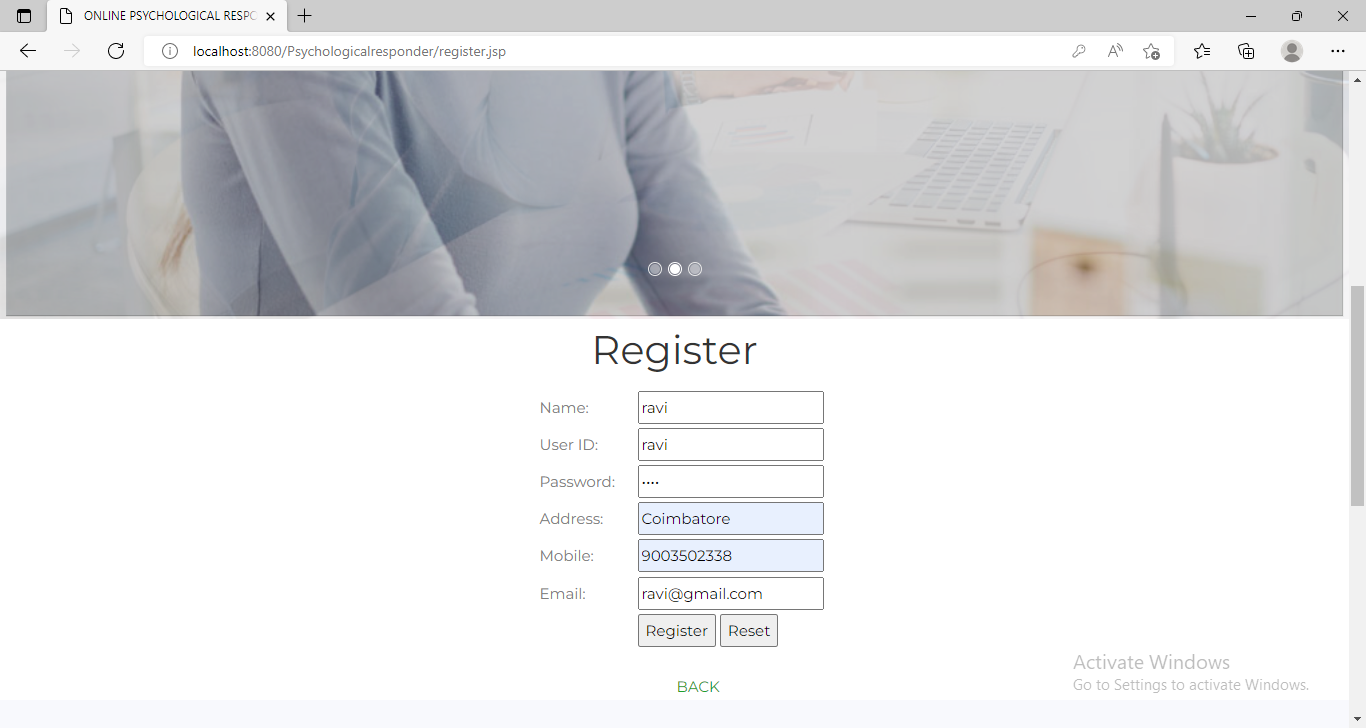


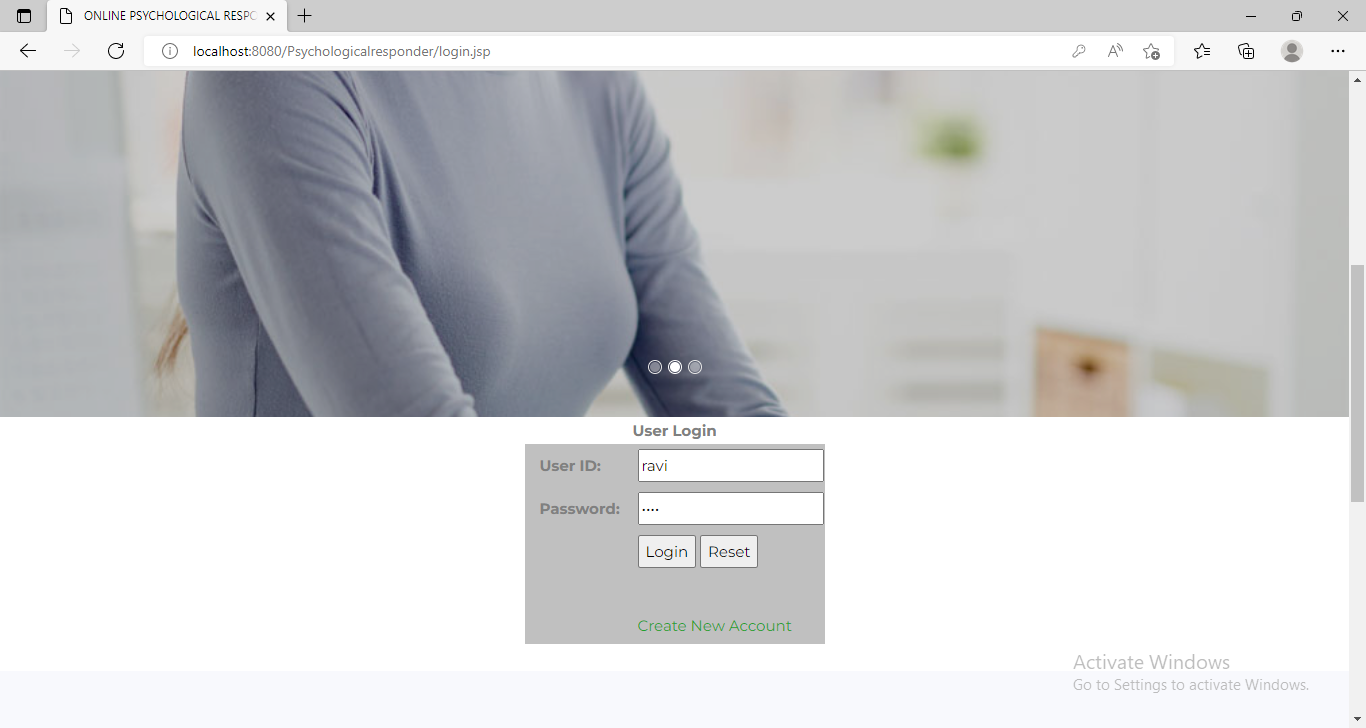


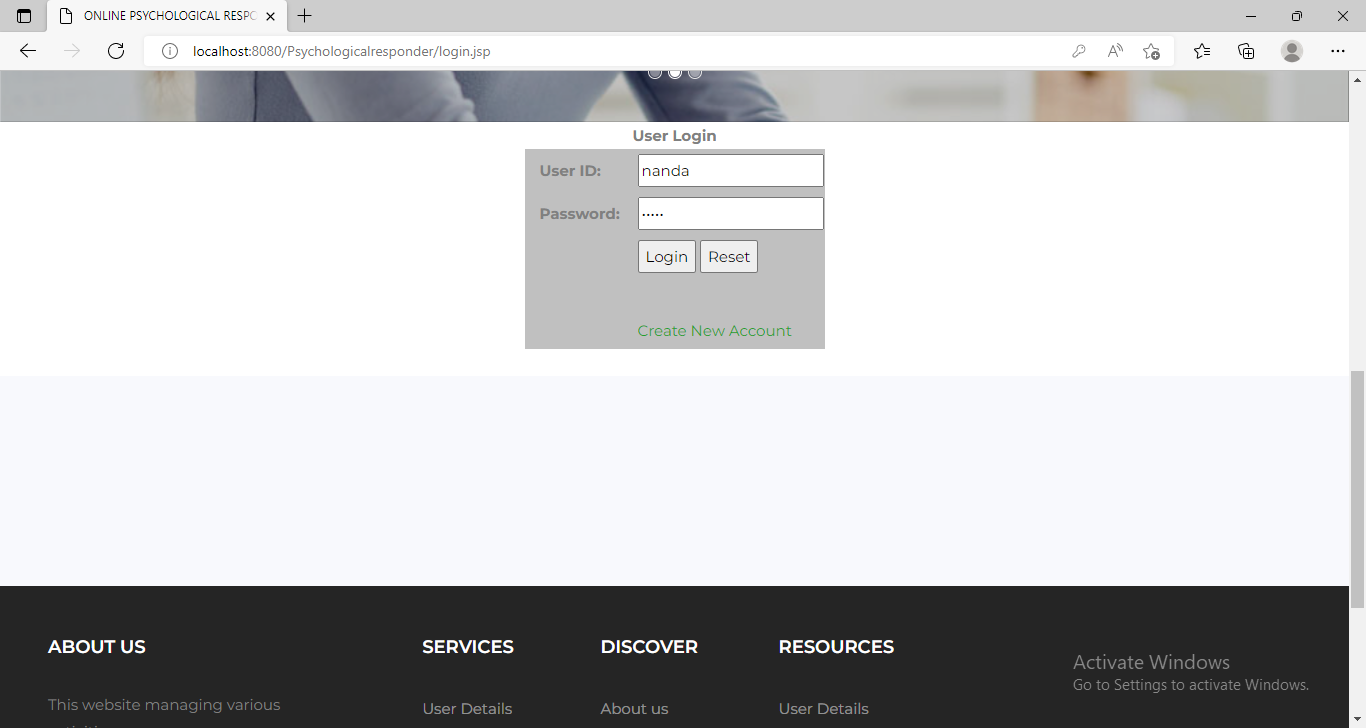


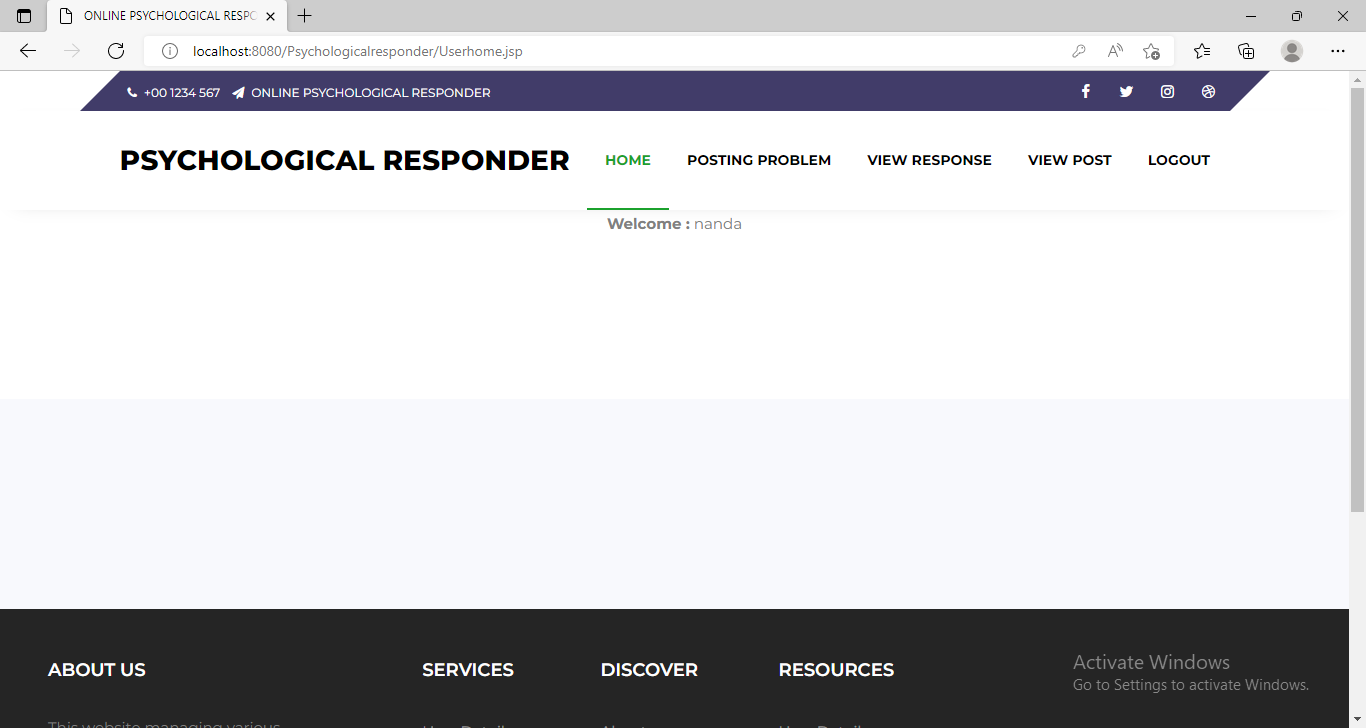


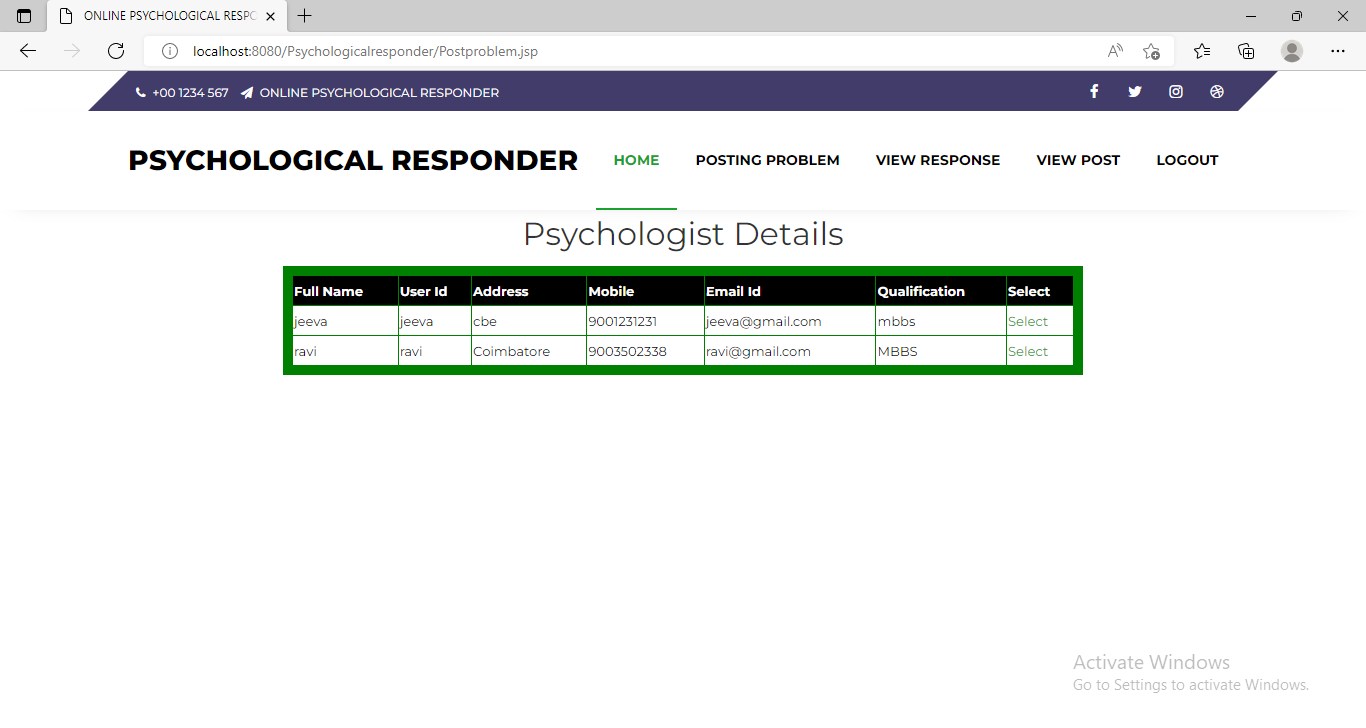












Code

<%--

Document : adminlogin

Created on : Jul 16, 2019, 12:18:42 PM

Author : Dell-pc

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<%@include file="Header.jsp" %>

<%@ page import="java.io.\*,java.sql.\*" %>

<%

if (request.getParameter("login")!=null)

{

String uid=request.getParameter("uid");

String pwd=request.getParameter("pwd");

if (uid.equals("Admin") && pwd.equals("Admin") )

{

response.sendRedirect("adminhome.jsp");

}

else

{

out.println("<script>alert('invalid Login!');</script>");

}

}

%>

<div class="a-grid">

<div align="center">

<form action="" name="login\_form" id="login\_form" method="post" >

<br>

<b>Admin Login</b>

<br>

<table border="0" cellspacing="4" cellspadding="4" class="displaycontent" bgcolor="Silver" width="300" Height="150">

<tr>

<td class="col">User ID:</td>

<td><input type="text" name="uid" value="" /></td>

</tr>

<tr>

<td class="col">Password:</td>

<td><input type="password" name="pwd" value="" class="required"/></td>

</tr>

<tr>

<td>&nbsp;</td>

<td >

<input type="submit" name="login" value="Login" />

<input type="Reset" value="Reset" />

</td>

</tr>

<tr>

<td>

</td>

</tr>

</table>

<br>

<br>

</form>

</div>

</div>

[%@include file="Footer.jsp" %](mailto:%25@include%20file=%22Footer.jsp%22%20%25)

<%--

Document : Viewpackage

Created on : Jul 17, 2019, 12:42:03 PM

Author : Dell-pc

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<%@include file="Psychologistsheader.jsp" %>

<%@ page import="java.io.\*,java.sql.\*" %>

<% Class.forName("com.mysql.jdbc.Driver");

Connection con1 =DriverManager.getConnection("jdbc:mysql://localhost:3306/PSYCHOLOGICALDB","root","");

Statement st=con1.createStatement();

ResultSet resultset = st.executeQuery("select \* from postproblem where psychologistid='"+ session.getAttribute("uid") +"'") ;

%>

<div align="center">

<form action="" name="Viewpackage" id="Viewpackage" method="post">

<div class="a-grid">

<h2> Problems From Users </h2>

<table border="2" cellspacing="6" class="displaycontent" width="800" style="border:10px solid Green;" align="center">

<tr>

<th bgcolor=Black><font color=white size=2>Id </font></th>

<th bgcolor=Black><font color=white size=2>psychologist id </font></th>

<th bgcolor=Black><font color=white size=2>User id </font></th>

<th bgcolor=Black><font color=white size=2>Problem </font></th>

<th bgcolor=Black><font color=white size=2>Postdate </font></th>

<th bgcolor=Black><font color=white size=2>Reponse</font></th>

<th bgcolor=Black><font color=white size=2>Select </font></th>

</tr>

<% while(resultset.next()){ %>

<tr>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(1) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(2) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(3) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(4) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(5) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(6) %></font></td>

<td bgcolor=white><font color=#000000 size=2><a href="Updateresponse.jsp?select=<%= resultset.getString(1) %>">Select</a></font></td>

</tr>

<% } %>

</table>

<br>

<!-- news -->

<%--

Document : Viewpackage

Created on : Jul 17, 2019, 12:42:03 PM

Author : Dell-pc

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<%@include file="Userheader.jsp" %>

<%@ page import="java.io.\*,java.sql.\*" %>

<% Class.forName("com.mysql.jdbc.Driver");

Connection con1 =DriverManager.getConnection("jdbc:mysql://localhost:3306/PSYCHOLOGICALDB","root","");

Statement st=con1.createStatement();

ResultSet resultset = st.executeQuery("select \* from postproblem where Userid='"+ session.getAttribute("uid") +"'") ;

%>

<div align="center">

<form action="" name="Viewpackage" id="Viewpackage" method="post">

<div class="a-grid">

<h2> Problems Status </h2>

<table border="2" cellspacing="6" class="displaycontent" width="800" style="border:10px solid Green;" align="center">

<tr>

<th bgcolor=Black><font color=white size=2>Id </font></th>

<th bgcolor=Black><font color=white size=2>psychologist id </font></th>

<th bgcolor=Black><font color=white size=2>User id </font></th>

<th bgcolor=Black><font color=white size=2>Problem </font></th>

<th bgcolor=Black><font color=white size=2>Postdate </font></th>

<th bgcolor=Black><font color=white size=2>Reponse</font></th>

</tr>

<% while(resultset.next()){ %>

<tr>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(1) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(2) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(3) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(4) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(5) %></font></td>

<td bgcolor=white><font color=#000000 size=2><%= resultset.getString(6) %></font></td>

</tr>

<% } %>

</table>

<br>

<!-- news -->

<%--

Document : Orderupdate

Created on : Jul 17, 2019, 12:49:27 PM

Author : Dell-pc

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<link href="jsDatePick\_ltr.min.css" type="text/css" rel="stylesheet" />

<script type="text/javascript" src="jsDatePick.min.1.3.js"></script>

<script type="text/javascript">

window.onload = function(){

new JsDatePick({

useMode:2,

target:"Adate",

dateFormat:"%d-%M-%Y",

selectedDate:{

day:31,

month:1,

year:2017

},

yearsRange:[1984,1996],

limitToToday:false,

//cellColorScheme:"beige",

dateFormat:"%d-%m-%Y",

//imgPath:"img/",

weekStartDay:1

});

};

function sum() {

var txtFirstNumberValue = document.getElementById('cost').value;

var txtSecondNumberValue = document.getElementById('tmem').value;

var result = parseInt(txtFirstNumberValue) \* parseInt(txtSecondNumberValue);

document.getElementById('tamt').value = result;

}

</script>

<%@ page import="java.io.\*,java.sql.\*" %>

<%

if (request.getParameter("select")!=null)

{

Class.forName("com.mysql.jdbc.Driver");

Connection con =DriverManager.getConnection("jdbc:mysql://localhost:3306/PSYCHOLOGICALDB","root","");

Statement st=con.createStatement();

ResultSet resultset = st.executeQuery("select \* from postproblem where id='"+ request.getParameter("select") +"'");

if(resultset.next()){

%>

<%@include file="Psychologistsheader.jsp" %>

<form action="" name="Orderupdate\_form" id="Orderupdate\_form" method="post">

<br>

<div align="center">

<div class="a-grid">

<h2> Order</h2>

<center >

<table border="0" cellspacing="4" cellspadding="4" class="displaycontent" width="400" height="200">

<tr>

<td class="col"><b>id:</b></td>

<td><input type="text" name="t1" value="<%= resultset.getString(1) %>" class="required email" readonly/></td>

</tr>

<tr>

<td class="col"><b>Response Here:</b></td>

<td><input type="text" name="t2" value="" class="required" cols="40" rows="5" style="width:200px; height:50px;"/></td>

</tr>

<tr>

<td>&nbsp;</td>

<td >

<input type="submit" name="register" value="Register" />

</td>

</tr>

<%} }%>

</table>

<br>

<!-- news -->

</div>

</div>

</form>

<% try

{

if (request.getParameter("register")!=null)

{

Class.forName("com.mysql.jdbc.Driver");

Connection con =DriverManager.getConnection("jdbc:mysql://localhost:3306/PSYCHOLOGICALDB","root","");

Statement st=con.createStatement();

String a1=request.getParameter("t1");

String a2=request.getParameter("t2");

// out.println(uid);

int i=0;

i= st.executeUpdate("update postproblem set Reponse='"+ a2 +"' where id='"+ a1 +"'");

if (i==1)

{

//out.println("<script>alert('Add Successfully');</script>");

response.sendRedirect("Viewproblems.jsp");

}

else

{

out.println("<script>alert('Not Successful');</script>");

}

}

}

catch(Exception ex)

{

out.println(ex);

}

%>