A

Minor Project

On

**ONLINE PLATFORM FOR MENTORING STARTUPS**

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K.Naveen Kumar(19E41A05H7)

M.Prakash(19E41A05F8)

K.Sai Kiran(19E41A05G5)

Under the Guidance of

Dr. S Venkata Achuta Rao

(Dean Academics)

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SREE DATTHA INSTITUTE OF ENGINEERING AND SCIENCE**

(Accredited by NAAC, Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) of the UGCAct.1956,

Sheriguda (V), Ibrahimpatnam (M), Ranga Reddy – 501510.

**2019-2023**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

**CERTIFICATE**

This is to certify that the project entitled “**ONLINE PLATFORM FOR MENTORING STARTUPS**” is being submitted by **K.Naveen Kumar(19E41A05H7) M.Prakash(19E41A05F8) K.Sai Kiran(19E41A05G5)** in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2022-23.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Mr. Sundeep kumar Dr.** **Md Sameeruddin Khan**

**Internal Guide Principal**

**Dr. M. Varaprasad Rao External Examiner**

**HoD**

**Submitted for viva voice Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ACKNOWLEDGEMENT**

Apart from our efforts, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard for my guide

Dr. S Venkata Achuta Rao**,** Dean Academics for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We are also thankful to **Dr. M. Varaprasad Rao,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are grateful to **Dr. S. Venkata Achuta Rao,** Dean Academics for his valuable technical support.

We are obliged to **Dr. Md. Sameeruddin Khan,** Principal for being cooperative throughout this project. We would like to express our sincere gratitude to **Dr. GNV Vibhav Reddy**, Vice-Chairman and **Sri.** **G. Panduranga Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout this project.

The guidance and support were received from all the members of **Sree Dattha Institute of Engineering and Science** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**ABSTRACT**

* An entrepreneur treads multiple stages of business lifecycle of a Startup from birth (an idea) through its maturity. Each new phase brings forth new challenges that the entrepreneur must learn to navigate and positives. Stakeholder engagement is also a prime importance for entrepreneur. finding potential partners, resources, professional bodies and entrepreneurs is very difficulty in now a day. to encourage entrepreneur and making their work easy in this project we are developing java-based web application, which can provide the end-to-end solution for all the activities of entrepreneur and Stakeholders. Mentoring offers a host of benefits including gaining industry insights and developing new skills.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 3.1 | Technical architecture | 09 |
| Figure 3.2 | Use case diagram | 10 |
| Figure 3.3 | Class diagram | 11 |
| Figure 3.4 | Sequence diagram | 12 |
| Figure 3.5 | Activity diagram | 13 |

**LIST OF SCREENSHOTS**

|  |  |  |
| --- | --- | --- |
| **SCREENSHOT NO.** | **SCREENSHOT NAME** | **PAGE NO** |
| Screenshot 6.1 | Admin login | 23 |
| Screenshot 6.2 | User registration | 24 |
| Screenshot 6.3 | Mentor registration | 25 |
| Screenshot 6.4 | Mentor activities | 26 |
| Screenshot 6.5 | User activities | 26 |
|  |  |  |

**TABLE OF CONTENTS**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ABSTRACT** | | |  |  | | | | I |
| **LIST OF FIGURES** | | | |  | | | | II |
| **LIST OF SCREENSHOTS** | | | | | | | | III |
| **1.** | **INTRODUCTION** | | | | | | | 1 |
|  | 1.1 | | PROJECT SCOPE | | | | | 2 |
|  | 1.2 | | EXISTING SYSTEM | | | | | 2 |
|  | 1.3 | | DISADVANTAGES OF EXISTING SYSTEM | | | | | 2 |
| **2.** | **SYSTEM ANALYSIS** | | | | | | | 3 |
|  | 2.1 | | PROPOSED SYSTEM  2.1.1 SYSTEM ANALYSIS  2.1.2 PROPOSED SYSTEM | | | | | 4  5  5 |
|  |  | | 2.1.3 | ADVANTAGES OF PROPOSED SYSTEM | | | |  |
|  | 2.2 | | FEASIBILITY STUDY | | | | | 5 |
|  |  | | 2.2.1 | TECHNICAL FESIBILITY | | | | 5 |
|  |  | | 2.2.2 | OPERATIONAL FEASIBILITY | | | | 5 |
|  |  | | 2.2.3 | ECONOMIC FEASIBILITY | | | | 6 |
|  | 2.3 | | HARDWARE & SOFTWARE REQUIREMENTS | | | | | 6 |
|  |  | | 2.3.1 | HARDWARE REQUIREMENTS | | | | 6 |
|  |  | | 2.3.2 | SOFTWARE REQUIREMENTS | | | | 7 |
| **3.** | **ARCHITECTURE** | | | | | | | 8 |
|  | 3.1 | | TECHNICAL ARCHITECTURE | | | | | 9 |
|  | 3.2 | | DESCRIPTION | | | | | 9 |
|  | 3.3 | | USECASE DIAGRAM | | | | | 10 |
|  | 3.4 | | CLASS DIAGRAM | | | | | 11 |
|  | 3.5 | | SEQUENCE DIAGRAM | | | | | 12 |
|  | 3.6 | | ACTIVITY DIAGRAM | | | | | 13 |
| **4.** | **IMPLEMENTATION** | | | | | | | 14 |
|  | 4.1 | | SAMPLE CODE | | | | | 14 |
| **5.** | **TESTING** | | |  | | | | 19 |
|  | | 5.1 | INTRODUCTION TO TESTING | | | | | 20 | |
|  | | 5.2 | TYPES OF TESTING | | | | | 20 | |
|  | |  | 5.2.1 | | | UNIT TESTING | | 20 | |
|  | |  | 5.2.2 | | | INTEGRATION TESTING | | 20 | |
|  | |  | 5.2.3 | | | VALIDATION TESTING | | 21 | |
| **6.** | **SCREENSHOTS** | | | | | | | 22 |
| **7.** | | **CONCLUSION & FUTURE SCOPE** | | | | | | 27 | |
|  | | 7.1 | PROJECT CONCLUSION | | | | | 28 | |
|  | | 7.2 | FUTURE SCOPE | | | | | 28 | |
| **8.** | | **BIBLIOGRAPHY** | | | | | | 29 | |
|  | | 8.1  8.2 | REFERENCES  GITHUB LINK | | | | | 30  30 | |
|  | |  | | | | |
|  | |  |  | | | | |  | |

**1. INTRODUCTION**

**1. INTRODUCTION**

**1.1 PROJECT SCOPE**

Outcome Intelligence Capture essential outcomes in real-time from all the mentorships in your mentoring groups. Skills development outcomes show how mentees developed from pre-mentorship to post-mentorship. Mentor Pitch compiles Mentoring Outcome Intelligence™ which includes metrics on each individual mentoring relationships and total group outcomes. This can be extremely valuable for group reporting.

**1.2 EXISTING SYSTEM**

The entrepreneur has to be discovers a problem or identifies an opportunity that has a business potential. Mentorship support for entrepreneur is still challenging for entrepreneur, because includes lot of stages to analyze the Concept or idea. Currently There is not any single platform which will helps to connect multiple stages of business lifecycle for Startup.

**1.3 PROBLEMS OF EXISTING SYSTEM**

* There is no single platform for providing mentoring for entrepreneurs.
* Here entrepreneurs didn’t get clarity what and how to do the business.
* Mentoring is possible by through physical classes only.
* For mentees money is required for the guidance from the mentors.

**2. SYSTEM ANALYSIS**

**2. SYSTEM ANALYSIS**

**2.1.1 SYSTEM ANALYSIS**

**SOFTWARE REQUIREMENT SPECIFICATION**

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

1. **Problem/Requirement Analysis:**

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

1. **Requirement Specification:**

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

**2.2.2 PROPOSED SYSTEM**

Entrepreneur builds a probable solution in the form of “a proof of concept” or “prototype” with relevant assumptions. Mentors, advisors and support systems: Several mentor organizations and groups working together to take the spirit of entrepreneurship to each level. Entrepreneur also get benefit from this proposed application. It is expanding the convenience of online mentorship platforms by allowing users to filter for their specific needs and receive answers on-demand from anywhere in the world.

**2.2.3 ADVANTAGES OF THE PROPOSED SYSTEM**

* It is a platform providing proper guidance to the entrepreneurs.
* The platform aims to help anyone anywhere in the world to succeed with their ideas by connecting them with experienced mentors.
* It offers users a casual one-to-one conversation via any communication platform, including Skype, Zoom or Google Hangouts.
* The platform saves users time and money as it provides answers to short-term problems on-demand.

**2.2 FEASIBILITY STUDY**

The next step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time“. But in reality both resources and time are scarce. Project should confirm to time bounce and should be optimal in there consumption of resources. These places a constant are approval of any project.

Feasibility has applied to Maintenance of Elementary School Data pertains to the following areas:

* Technical feasibility
* Operational feasibility
* Economical feasibility

**2.2.1 TECHNICAL FEASIBILITY**

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system.

Maintenance of Elementary School Data uses the web technologies, which is rampantly employed these days worldwide. The world without the web is incomprehensible today. That goes to proposed system is technically feasible.

**2.2.2 OPERATIONAL FEASIBILITY**

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly and to use.

**2.2.3 ECONOMICAL FEASIBILITY**

To decide whether a project is economically feasible, we have to consider various factors as:

* + - * Cost benefit analysis
      * Long-term returns
      * Maintenance costs

The proposed Maintenance of Elementary School Data is computer based. It requires average computing capabilities and access to internet, which are very basic requirements and can be afforded by any organization hence it doesn’t incur additional economic overheads, which renders the system economically feasible.

**2.3 HARDWARE & SOFTWARE REQUIREMENTS**

**2.3.1 HARDWARE REQUIREMENTS**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

* Processor : Intel core i5 and above
* Hard disk : 256GB SSD or 1TB HDD
* RAM : 8GB or Above.
* Monitor : 15 inches or above.

**2.3.2 SOFTWARE REQUIREMENTS:**

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

* Operating system : Windows 10
* Technologies : Java version 8
* Web based Technologies : HTML, CSS, Java Script, Servlets, JSP
* IDE : Eclipse
* Web Server : Apache Tomcat 8.5
* Database : MySQL 5.5

**3. ARCHITECTURE**

**3. ARCHITECTURE**

**3.1 TECHNICAL ARCHITECTURE**



Figure 3.1: Technical architecture

**3.2 DESCRIPTION**

Input Data: Enter the input for admin, mentor and user

Wed server: The input data is sent to the web server to validate the user, admin and mentor

Database: The valid user, mentor and admin details is stored in the database tables

**3.3 USE CASE DIAGRAM**

A use case diagram is a graph of actors set of use cases enclosed by a system boundary, communication associations between the actors and users and generalization among use cases. The use case model defines the outside(actors) and inside(use case) of the system’s behavior.

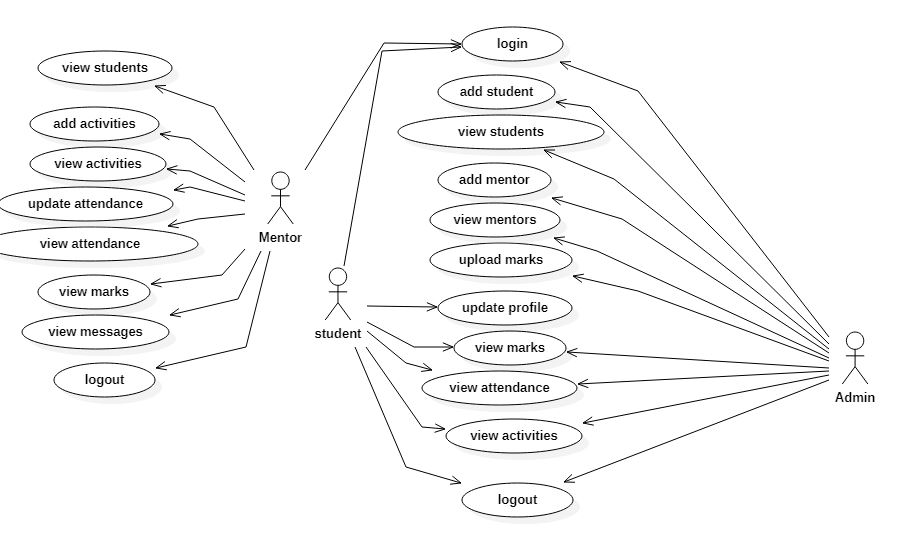
 use case diagram is quite simple in nature and depicts two types of elements: one representing the business roles and the other representing the business processes.

Figure 3.2: Use case for online platform for mentoring startups

**3.4 CLASS DIAGRAM**

A class is a representation of an object and, in many ways; it is simply a template from which objects are created. Classes form the main building blocks of an object-oriented application.

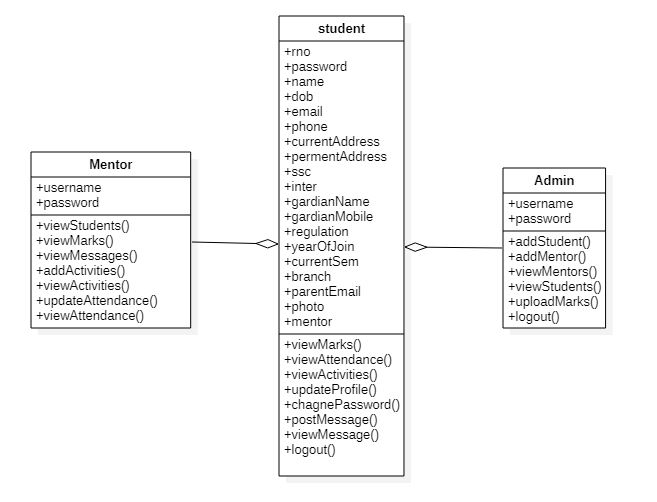


Figure 3.3: Class Diagram for Online platform for mentoring startups

**3.5 SEQUENCE DIAGRAM**

The Figure 3.4 UML sequence diagrams are used to represent the flow of messages, events and actions between the objects or components of a system. Time is represented in the vertical direction showing the sequence of interactions of the header elements, which are displayed horizontally at the top of the diagram.

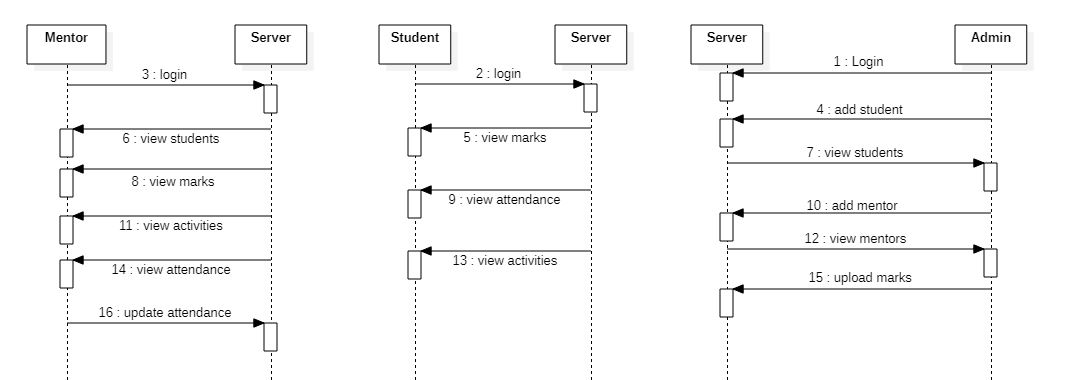
****

Figure 3.4: Sequence Diagram for Online platform for mentoring startups

3.6 ACTIVITY DIAGRAM

Activity diagrams represent the business and operational workflows of a system. An Activity diagram is a dynamic diagram that shows the activity and the event that causes the object to be in the particular state.

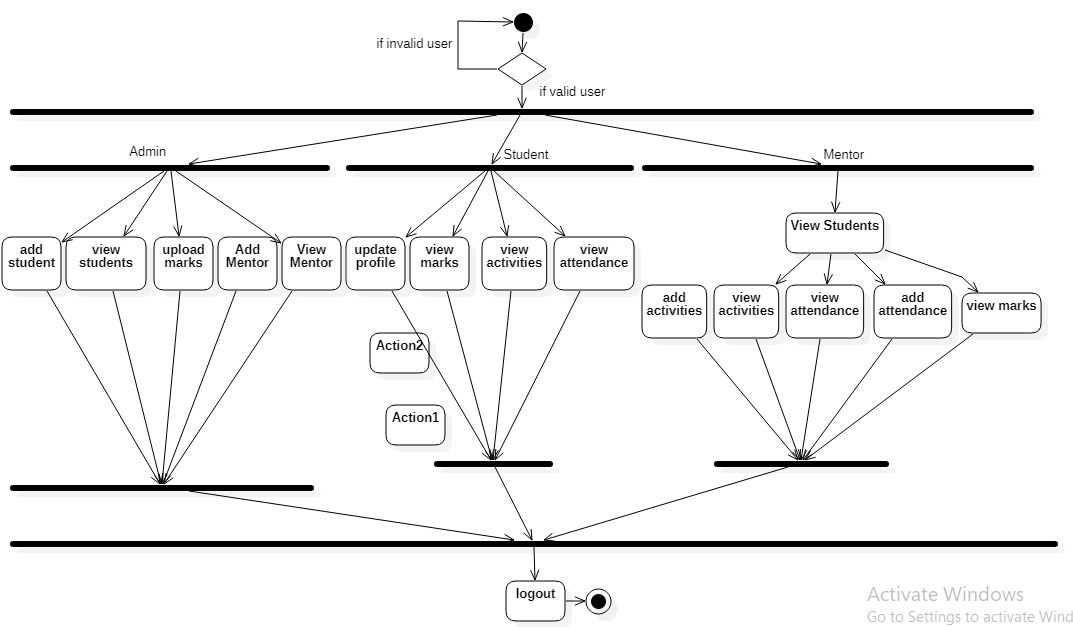


Figure 3.5: Activity Diagram for User for Online platform for mentoring startups

`**4. IMPLEMENTATION**

**4. IMPLEMENTATION**

**4.1 SAMPLE CODE**

**Httprequest:**

**package** com.voidmain.servlets;

**import** java.io.File;

**import** java.lang.reflect.Method;

**import** java.util.ArrayList;

**import** java.util.Enumeration;

**import** java.util.HashMap;

**import** java.util.List;

**import** java.util.Map;

**import** javax.servlet.http.HttpServletRequest;

**import** org.apache.tomcat.util.http.fileupload.FileItem;

**import** org.apache.tomcat.util.http.fileupload.disk.DiskFileItemFactory;

**import** org.apache.tomcat.util.http.fileupload.servlet.ServletFileUpload;

**import** org.apache.tomcat.util.http.fileupload.servlet.ServletRequestContext;

**public** **class** HttpRequestParser {

**public** **static** Map<Object,List<String>> parseMultiPartRequest(HttpServletRequest request,Object obj)

{

Map<Object,List<String>> map=**new** HashMap<Object, List<String>>();

**if**(ServletFileUpload.*isMultipartContent*(request))

{

**try** {

List<String> files=**new** ArrayList<String>();

**for** (FileItem item : **new** ServletFileUpload(**new** DiskFileItemFactory()).parseRequest(**new** ServletRequestContext(request))) {

// Root Directory.

String uploadRootPath = request.getServletContext().getRealPath("")+"/documents/";

System.***out***.println("uploadRootPath=" + uploadRootPath);

File uploadRootDir = **new** File(uploadRootPath);

// Create directory if it not exists.

**if** (!uploadRootDir.exists()) {

uploadRootDir.mkdirs();

}

**if** (!item.isFormField()) {

item.write(**new** File(uploadRootPath+item.getName()));

files.add(item.getName());

}

**else**

{

String fieldName = item.getFieldName();

String fieldValue = item.getString();

Method[] methods=obj.getClass().getDeclaredMethods();

**for**(Method method : methods)

{

String methodName=method.getName();

**if**(methodName.equalsIgnoreCase("set"+fieldName))

{

**try** {

method.setAccessible(**true**);

String type=method.getParameterTypes()[0].getName();

**if**(type.equals("java.lang.String"))

{

method.invoke(obj,fieldValue);

}**else** **if**(type.equals("int"))

{

method.invoke(obj,Integer.*parseInt*(fieldValue));

}**else** **if**(type.equals("float"))

{

method.invoke(obj,Float.*parseFloat*(fieldValue));

}**else** **if**(type.equals("long"))

{

method.invoke(obj,Long.*parseLong*(fieldValue));

}**else** **if**(type.equals("dobule"))

{

method.invoke(obj,Double.*parseDouble*(fieldValue));

}**else** **if**(type.equals("boolean"))

{

method.invoke(obj,Boolean.*parseBoolean*(fieldValue));

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

}

}

map.put(obj,files);

}

**catch** (Exception e) {

e.printStackTrace();

}

}

**return** map;

}

**public** **static** Object parseRequest(HttpServletRequest request,Object obj)

{

Enumeration<String> enumeration=request.getParameterNames();

**while**(enumeration.hasMoreElements())

{

String parameterName=enumeration.nextElement();

Method[] methods=obj.getClass().getDeclaredMethods();

**for**(Method method : methods)

{

String methodName=method.getName();

**if**(methodName.equalsIgnoreCase("set"+parameterName))

{

**try** {

method.setAccessible(**true**);

String type=method.getParameterTypes()[0].getName();

**if**(type.equals("java.lang.String"))

{

method.invoke(obj,request.getParameter(parameterName));

}**else** **if**(type.equals("int"))

{

method.invoke(obj,Integer.*parseInt*(request.getParameter(parameterName)));

}**else** **if**(type.equals("float"))

{

method.invoke(obj,Float.*parseFloat*(request.getParameter(parameterName)));

}**else** **if**(type.equals("long"))

{

method.invoke(obj,Long.*parseLong*(request.getParameter(parameterName)));

}**else** **if**(type.equals("dobule"))

{

method.invoke(obj,Double.*parseDouble*(request.getParameter(parameterName)));

}**else** **if**(type.equals("boolean"))

{

method.invoke(obj,Boolean.*parseBoolean*(request.getParameter(parameterName)));

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

}

**return** obj;

}

}

**5. TESTING**

**5. TESTING**

**5.1 INTRODUCTION TO TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

**5.2 TYPES OF TESTING**

**5.2.1 UNIT TESTING**

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

**5.2.2 INTEGRATION TESTING**

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

**5.2.3 VALIDATION TESTING**

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

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

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

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

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

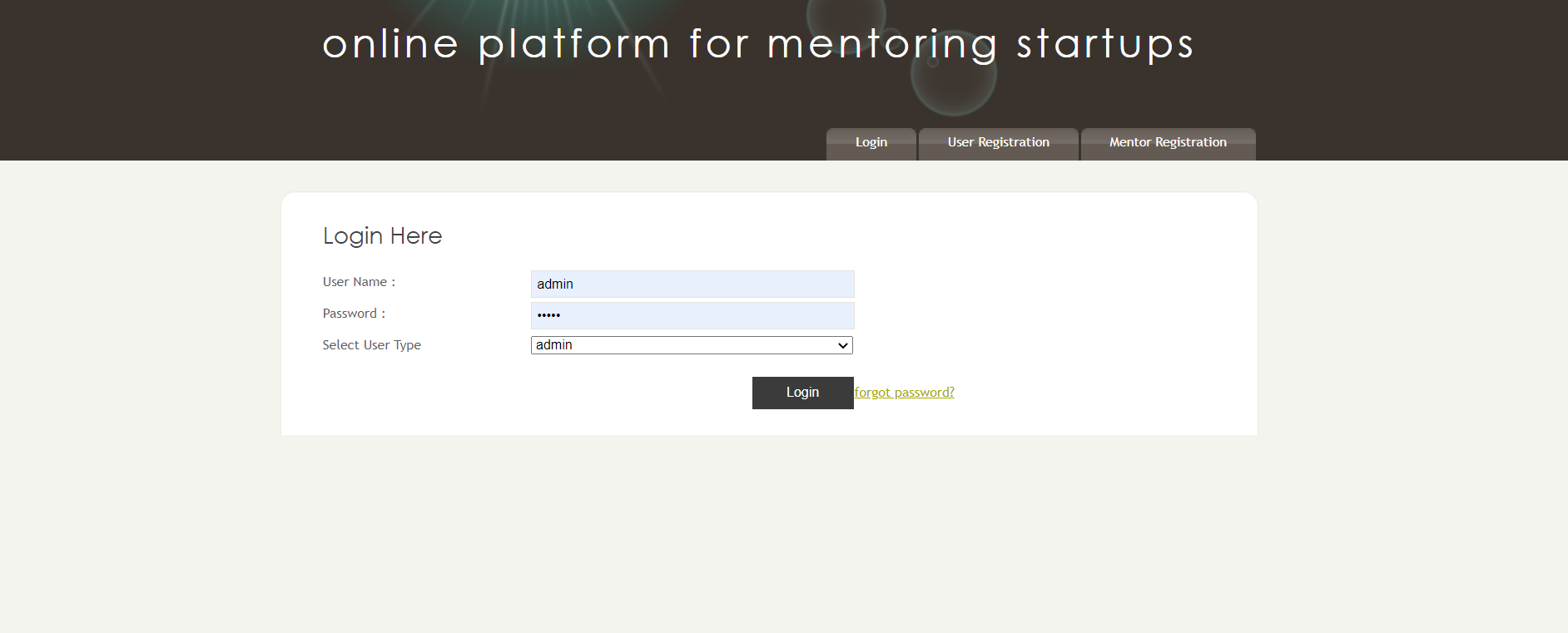
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

**6. SCREENSHOTS**

**6. SCREENSHOTS**

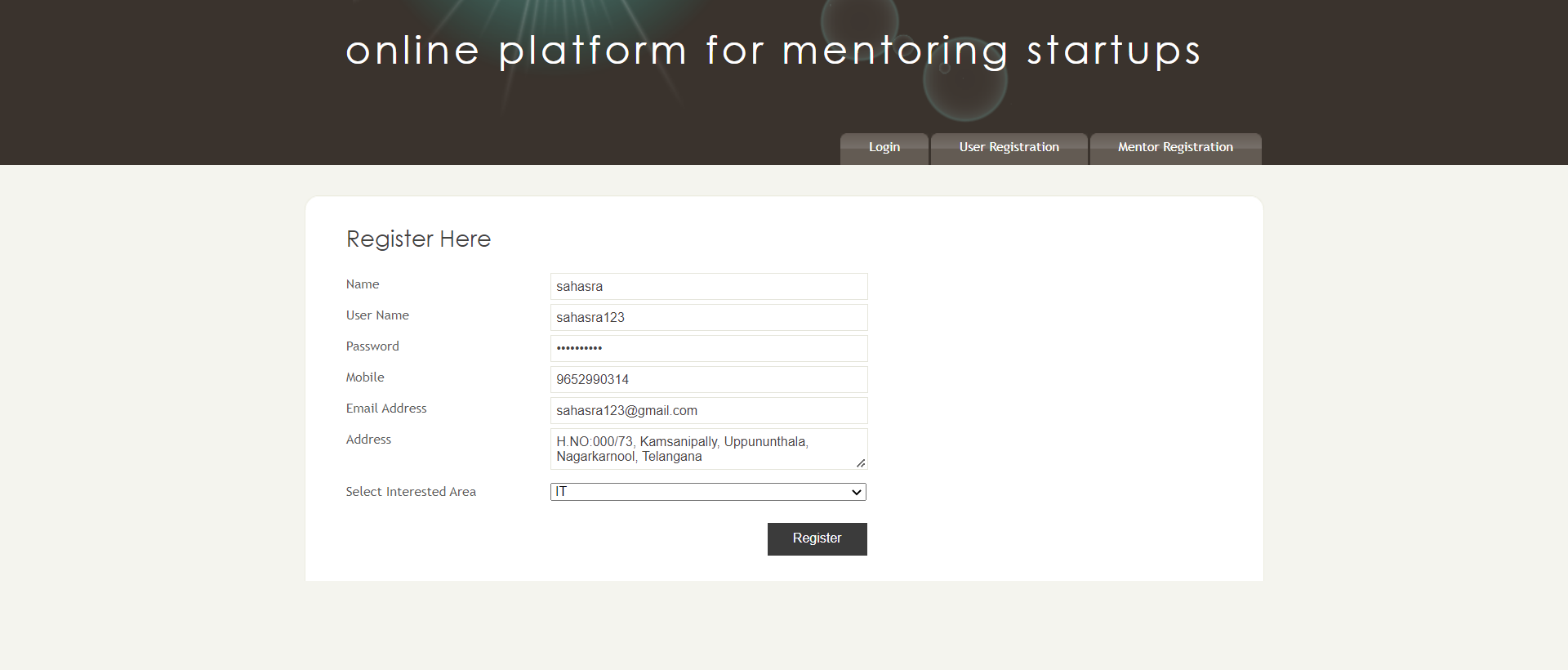
**6.1 RESULTS**

In the below screenshots we can see the results of Admin login page. The following screenshot 6.1 shows us the page of admin.



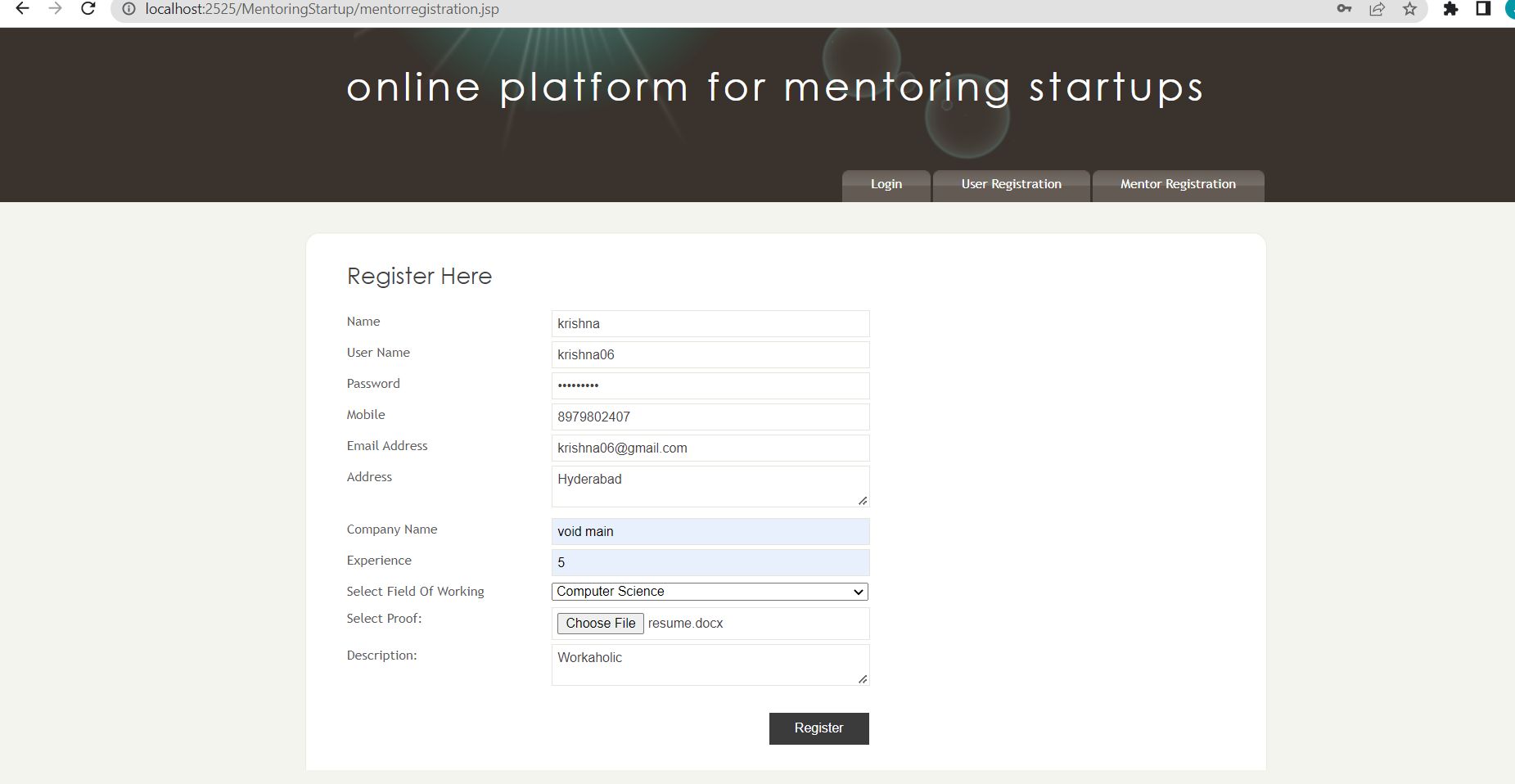
Screenshot 6.1: Admin page

The following screenshot 6.2 shows us the user registration page, where it gives the details of user data



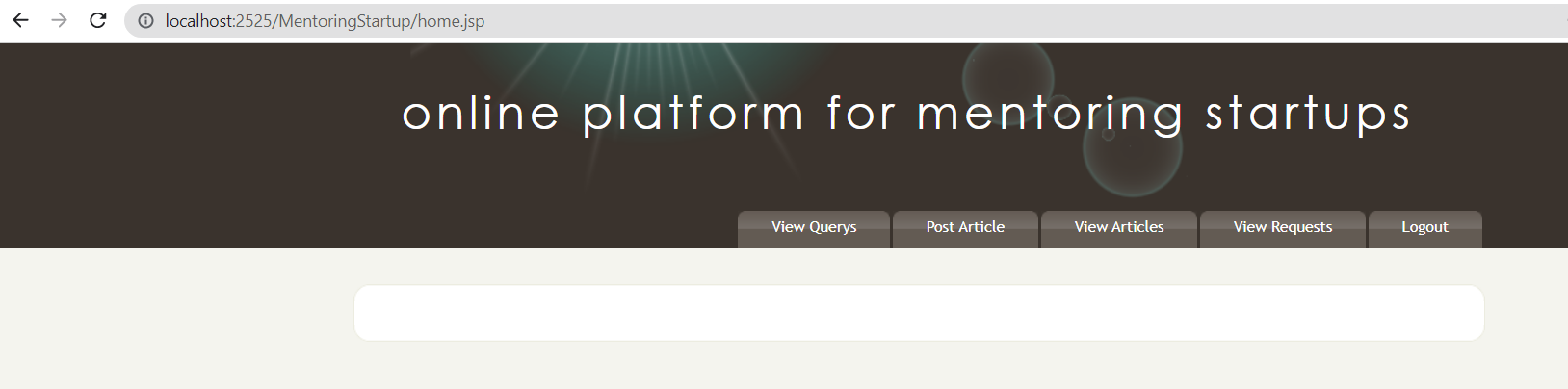
Screenshot 6.2: user registration

The following screenshot 6.3 shows us the mentor registration, where it gives the details of mentor

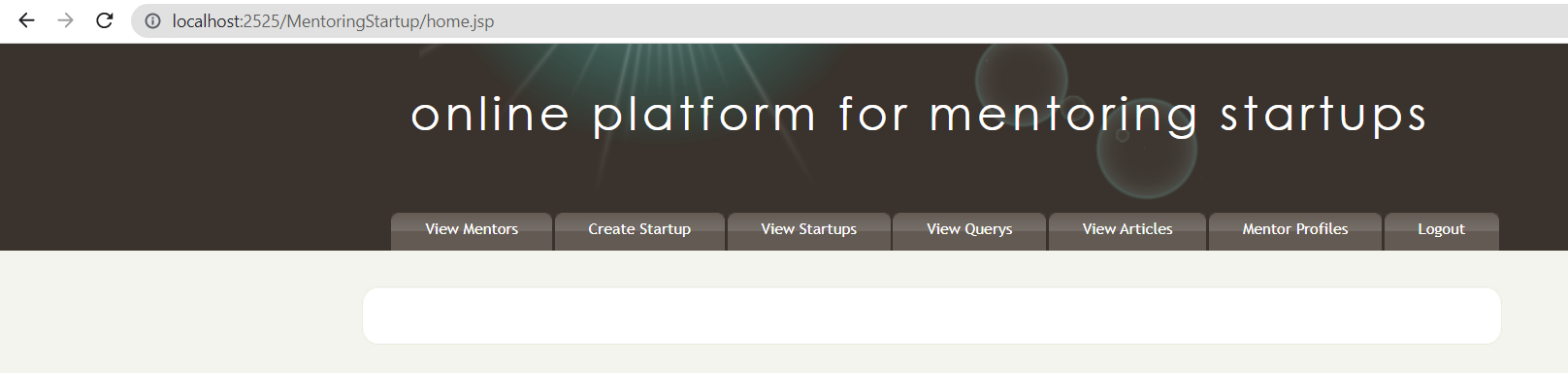


Screenshot 6.3: mentor registration page

The following screenshot 6.4 shows us the mentor activities



Screenshot 6.4: mentor activities

The following screenshot 6.5 shows us the user activities

Screenshot 6.5: user activities

**7. CONCLUSION**

**7. CONCLUSION & FUTURE SCOPE**

**7.1 PROJECT CONCLUSION**

The conclusion on the project is Mentoring of students through Online, we can reduce time to access the data of students. Here calculations and comparisons of students academics year to year is done automatically by Bar graphs & Pie charts i.e., which is user friendly Interface. Information of every student can be stored in same database by which we can access students information. Searching an individual students information provides quick access. All the manual mentoring defects are improvised by smart mentoring.

**7.2 FUTURE SCOPE**

In future, we aim to have this platform for the business people to grow in their field of business.

**8. BIBLIOGRAPHY**

**8. BIBLIOGRAPHY**

**8.1 REFERENCES**

[1] Dietel and Dietel,”Advanced java programming”,text ref

[2] john zukowski,”Masterinh JAVA”,text ref

[3] Roger S Pressman,”Software engineering”,text ref

[4] Senn,”Analysis & Design of information System”,text ref

**8.2 GITHUB LINK**