

**Project Title:** **Management of Training & jobs Service**

## Submitted By:

Ahmad Abu ghazal- 201820056

Abdullah Majed Abughali– 201410848

## Supervisor:

Dr. Muhammad Al-Tayee

January – 2022

# Approval

We certify that we have read the project titled “**Management of Training & jobs Service**”, and as a members of project evaluation committee we had examined the students in the content of this document and knowledge related to it, and we certify that it is adequate with standings as a project for partial fulfillment of the requirements of B.Sc. in Software Engineering department.

Chairman: Member:

Name: Name:

Date: Date:

Signature: Signature:

# Certificate

It is certified that this project has been prepared and written under my direct supervision and guidance. I also would like to certify that this document is approved for submission and evaluation.

Supervisor:

Signature:

Date:

# Dedication

THIS PROJECT IS DEDICATED TO ALL OUR FAMILY MEMBERS OF

MUNES BANI FAWAZ & IBRAHIM AL HAMMAD ESPECIALLY

OUR PARENTS, BROTHERS AND SISTERS**.** WE LOVE YOU ALL**.**

ALSO, WE WOULD LIKE TO DEDICATE THIS PROJECT TO OUR TEACHERS

FOR ALL THE GOOD TIMES WE HAD WITH THEM THROUGH OUR TIME IN

THE UNIVERSITY AND FOR ALL THE GOOD AND HARD WORK THEY DID BY TEACHING US, MAY THEIR KNOWLEDGE CONTINUE THROUGHOUT ALL GENERATIONS**.**

SO, WE DEDICATE THIS PROJECT TO:

DR**.** LAMIS AL-QORA'N

DR**.** ABDULRAHMAN OBIDAT

DR**.** ENAS AL-NAFFAR

DR**.** MUHAMMAD AL-TAYEE

DR**.** RAWAN ABU AL-LAIL

# Table of Contents

Approval.......................................................................................................................................2

Certificate.....................................................................................................................................3

Dedication....................................................................................................................................4

List of Figures..............................................................................................................................7

List of Tables.............................................................................................................................. 7

Glossary of Terms...................................................................................................................... 8

Chapter 1 - System Conception................................................................................................. 9

1.1 - Who is the application for? ....................................................................................... 9

1.2 - What problems will it solve? …………………………........................................... 9

1.3 - Why is it needed? .......................................................................................................9

1.4 - Where will it be used? ............................................................................................... 9

1.5 - How will it work? .......................................................................................................9

1.6 - When is it needed? ....................................................................................................10

Chapter 2 - Domain Analysis................................................................................................... 11

2.1 - Classes (Brainstorm)................................................................................................ 11

2.2 - Bad Classes............................................................................................................. 11

2.3 - Good Classes........................................................................................................... 11

2.4 - Data Dictionary....................................................................................................... 11

2.4 - Data Dictionary...................................................................................................... 12

2.5 - Association’s & Attributes...................................................................................... 13

2.6- dependency …………………………………………………………..................... 14

Chapter 3 - Application Analysis........................................................................................... 15

3.1 - Introduction about application analysis................................................................. 15

3.2 - Functional requirement .......................................................................................... 15

3.3 - Non-Functional requirement .................................................................................. 16

3.4 - Use Case Diagram.................................................................................................. 17

3.5 - Use Case Requirements Specification .................................................................. 18

Chapter 4 - System Architecture............................................................................................ 23

4.1 - Introduction about software Architecture ............................................................. 23

4.2 - What is software architecture? ............................................................................... 23

4.3 - What is your software Architecture? ..................................................................... 23

4.4 - Why did you choose it? ......................................................................................... 23

4.5 - Benefits of three-tier architecture ........................................................................... 23

4.6 - Description of the three tiers in detail............................................................... 23, 24

Chapter 5 - System Design .................................................................................................... 26

5.1 - Introduction about System Design ........................................................................ 26

5.2 - Class diagram..........................................................................................................26

5.3 - Relational Database Table...................................................................................... 27

5.4 - User Interface Design ........................................................................................... 30

Chapter 6 - Implementation................................................................................................... 34

6.1 - Introduction about Implementation....................................................................... 34

6.2 - Implementing the Login Method........................................................................... 34

6.3 - Implementing the Login Authentication Method.................................................. 35

6.4 - Implementing the Apply for training .................................................................. 36

6.5 - Implementing the Admin...................................................................................... 38

6.6 - Implementing the courses .................................................................................... 39

6.7 - Implementing the Registration.............................................................................. 40

Chapter 7 - System Testing................................................................................................... 41

7.1 - Introduction about System Testing........................................................................ 41

7.2 - Black Box Testing ................................................................................................ 41

7.3 - Testing by Using ................................................................................................... 43

7.4 - Testing by Using Results....................................................................................... 45

Chapter 8 - Conclusion & Future work.................................................................................. 46

8.1 - Conclusion.............................................................................................................. 46

8.2 - Future work............................................................................................................ 46

References.............................................................................................................................. 47

Appendix ......................................................................................................................... 48-49

Project Submission........................................................................................................ 50

# List of Figures

Figure (2.1): Class Diagram..................................................................................................... 13

Figure (3.1): Use Case Diagram............................................................................................... 17

Figure (4.1): Three Tier Architecture....................................................................................... 25

Figure (5.1): Class Diagram...................................................................................................... 27

Figure (5.2): Relational Database Table.................................................................................... 29

Figure (5.3): User Interface Design (Home Page)..................................................................... 30

Figure (5.4): User Interface Design (About Page)..................................................................... 31

Figure (5.5): User Interface Design (About Page) ..................................................................... 31

Figure (5.6): User Interface Design (services Page) .................................................................. 32

Figure (5.7): User Interface Design (University Page)…………............................................... 32

Figure (5.8): User Interface Design (Company Page)................................................................. 33

Figure (6.1): Implementation (Login Page Design & Code)....................................................... 34

Figure (6.2): Implementation (Login Page Design & Code)....................................................... 35

Figure (6.3): Implementation (Login Authentication Code)....................................................... 36

Figure (6.4): Implementation (Applying Design)....................................................................... 37

Figure (6.5): Implementation (Applying Code) ......................................................................... 38

Figure (6.6): Implementation (Admin) ..................................................................................... 38

Figure (6.7): Implementation (Courses Design & Code) ......................................................... 39

Figure (6.8): Implementation (Courses Design & Code) .......................................................... 39

Figure (6.9): Implementation (Registration Design & Code) ................................................... 40

Figure (6.9.1): Implementation (Registration Design & Code) ................................................ 40

Figure (7.1): Black Box (Testing Case #1)................................................................................ 42

Figure (7.2): Black Box (Testing Case #2)................................................................................ 42

Figure (7.3): Black Box (Testing Case #3)................................................................................ 43

Figure (7.4): Black Box (Testing Case #4)............................................................................... 44 Figure (7.5): Black Box (Testing Case #4)............................................................................... 44

**List of Tables**

## Table (1): Glossary of Terms…………………………………………………………………...…8

## Table (3.1) Register ………………………………………………………………………….. 17

## Table (3.2) Login ………………………………………………………………………….….. 17

## Table (3.3) Logout ……………………………………………………………….….. 18

## Table (3.4) Apply for training ……………………………………………………….. 18

## Table (3.5) Send Form ……………………………………………………………….. 19

## Table (3.6) View Details ……………………………………………………………... 19

## Table (3.7) View Submitted Form ……………………………………………….…... 20

Table (7.1): Black Box (Testing)**……………………………………………………….39**

Table (7.2): Testing by Using (Samples)**…………………………………… …………...**41

Table (7.3): Testing by Using (Results)**……………………………………….….......**43

Table (7.4): Testing by Using (Results)**……………………………………….….......**43

**Glossary of Terms**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| HOD | Head of Department |

## Table (1): Glossary of Terms

**Chapter 1 - System Conception**

Here we are going to answer 6 questions in regards to understand the application and its usage and what it is intended for.

**Chapter 1 - System Conception**

Here we are going to answer 6 questions in regards to understand the application and its usage and what it is intended for.

**Q1) who is this application for?**

For the students who still in the university or they graduate andlookingfora place to take courses and to get experience inthebest companies and for the universities and companies

**Q2) what problem it will solve?**

The biggest problems that will get solved is that this project:

Gives the advantage of not needing to go to the companies to apply for a certain courses and jobs   
it will save a good amount of time and effort for students and companies and universities.

As a student you will gain experience before you start working with a huge companies

It will help the student knowledge about the requirement of labor market.

**Q3) where will it be used?**

This web application can be used by freshly graduated students or current students on

Google Chrome or Microsoft Edge

**Q4) when is it needed?**

It is needed when a student needs to apply for a certain jobs or courses and students will be able to gain experience before applying for jobs. The project will be considered done on the 5th of February approximately.

**Q5) why is it needed?**

It will save a decent amount of time and effort for students when it comes to looking for jobs and courses, alongside meeting a lot of people with a lot of experience they could share with the freshly graduated or current students   
companies can look for qualified candidate to work with them.

**Q6) how will it work?**

The Companies will be able to see student’s information and their CV’s.

Students can apply for training in the elite companies and they could upload their CV’s and information on the website

Universities give priorities to students applied for training through our website and universities are mediators between students and companies.

**Chapter 2 - Domain Analysis**

Here we are going to explain the Classes and what they are needed for in order to understand the project furthermore by analyzing the classes and to explain what each class is.

### **2.1 - Classes (Brainstorm)**

(users),(company),(university),(usertype),(CodeUnivesity),(transaction),(Code nationality),(Courses),(Student master),(Code Company),(Admin)

### **2.2 - Bad Classes**

(User type), (Code University), (Code nationality), (Code Company)

### **2.3 - Good Classes**

(Users), (company), (university), (Courses), (transaction), (Student master),(Admin).

### **2.4 - Data Dictionary**

**Users**: A user is a person who utilizes a computer or network service.

A user often has a user account and is identified to the system by a username .Other terms for username include login, account name,

**Company**: A company, abbreviated as co., is a legal entity representing an association of people, whether natural, legal or a mixture of both, with a specific objective. Company members share a common purpose and unite to achieve specific, declared goals. Companies take various forms:

Voluntary associations, which may include nonprofit organizations

Business entities, whose aim is generating profit

Financial entities and banks

Programs or educational institutions

**University**: Is an institution of higher (or tertiary) education and research which awards academic degrees in several academic disciplines. Universities typically offer both undergraduate and postgraduate programs in different faculties of learning.

**Courses**: Course (education), a unit of instruction in one subject, lasting one academic term

Course of study, or academic major, a program of education leading to a degree or diploma

**Transaction**: an agreement, communication, or movement carried out between users and the University or the Company.

**Student master**: the student who will enter for the website to search for a job or a courses

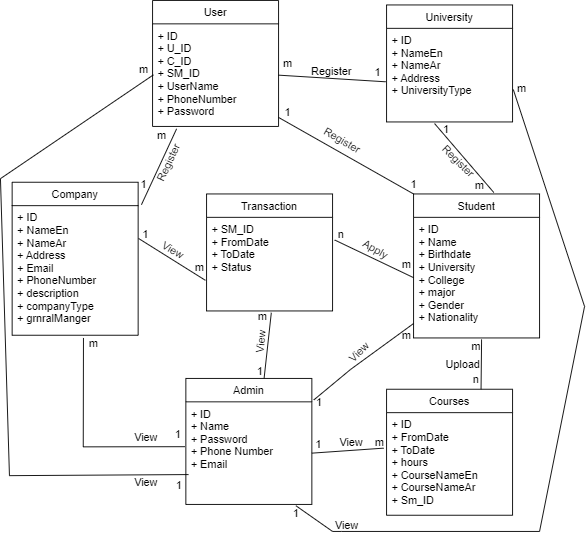
**Admin**: a person who has all the authorities to add/delete/edit Companies, Universities and Students on the website.

### **2.5 - Association’s & Attributes Class Diagram:**

Here we will see the class diagram for the project that is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application and for detailed modeling thus it can be used for data modeling. All that is demonstrated in Figure (1) on the next page.

**Here we will explain the Uml diagram:**

User’s will be able to register or login to the website and will be able to see the courser and jobs submitted by the companies , university can help the student through suggesting suitable courses to take or help them with job opportunity , with transaction the student can apply for a likable course to take.



**Figure (2.1): Class Diagram**

**Chapter 3 - Application Analysis**

### **3.1 - Introduction about application analysis**

Requirement is a condition or capability possessed by the software or system component in order to solve a real-world problem. The problems can be to automate a part of a system, to correct shortcomings of an existing system, to control a device, and so on. **IEEE** defines requirement as a condition or capability needed by a user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. Requirements describe how a system should act, appear or perform. For this, when users request for software, they provide an approximation of what the new system should be capable of doing. Requirements differ from one user to another and from one business process to another.

Different Types of Software Requirements:

1. Functional Requirements.
2. Non-Functional Requirements.

Find Actors:

1. Student.
2. Company
3. University
4. Admin

### **3.2 - Functional requirement**

**1) Register:** the admin can registrant the new student or company and university

**2) Login:** the Students, admin, university and Company can use their username and password to log in to the website

**3) Logout:** the Students, admin, University and Company can logout from the website using a button.

**4) Apply for training**: The Student after Signing in will fill the form he needs

**5) Send Form**: After filling a form the student sends the form to the Companies.

**6) View Details**: the Students, admin, university and Company can view all the information in the forms.

**7) View Submitted Form:** The Company, admin can view the forms that students applied for.

### **3.3 - Non-Functional requirement**

1. **Performance**: defines how fast a software system or its particular piece responds to certain users’ actions under certain workload.

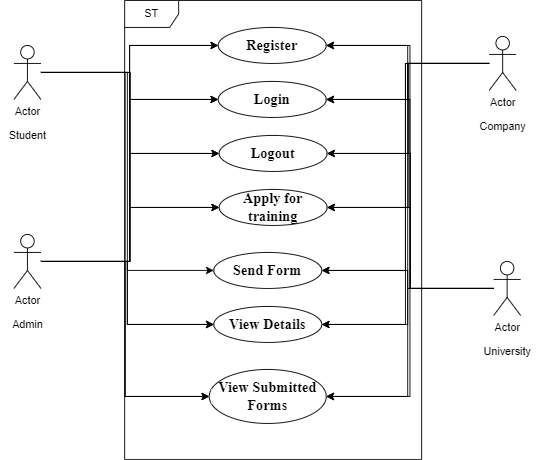
* Based on response time which is less than 7 seconds.

**b) Authorization:** is a non-functional requirement, which **relates to how system functionality works between the actors (admin, university, company, students)**

* Based on testing different logged in accounts

### **3.4 - Use Case Diagram**

Here we will see the use case that represents a written description of how users will perform tasks on the website. It outlines from a user's point of view the system's behavior as it responds to a request. All that demonstrated in Figure (2).



**Figure (3.1): USE CASE DIAGRAM**

### **3.5 - Use Case Requirements Specification**

Here we will talk about the use case specifications that is typically created in the analysis and design phase in an iterative manner. At first, only a brief description of the steps needed to carry out the normal flow of the use case (what functionality is provided by the use case) that is written.

**Table 1) Register.**

|  |  |
| --- | --- |
| Overview | |
| Title | Register |
| Description | the admin can registrant the new student or company |
| Actor | Student, Company, University ,admin |
| Initial Status | None |
| Basic Flow | |
| Step1: Enter the URL of the website.  Step2: Click on Registration in login page, then fill the forms of step 1 and step 2.  Step3: Click Create. | |
| Post Condition | |
| After clicking create you can login. | |
| Alternative Flow | |
| None | |

**Table 2) Login.**

|  |  |
| --- | --- |
| Overview | |
| Title | Login |
| Description | Use username and password to enter the website |
| Actor | Student, Company, University ,admin |
| Initial Status | None |
| Basic Flow | |
| Step1: Enter the URL of the website.  Step2: Choose Student Portal or Admin Portal (Company). Step3: Enter username and password.  Step3: Click login.  Step4: The portal will open | |
| Post Condition | |
| After clicking login the portal will open. | |
| Alternative Flow | |
| In Step 3 when you enter an invalid password an alert will appear. | |

**Table 3) Logout.**

|  |  |
| --- | --- |
| Overview | |
| Title | Logout |
| Description | Used to exit the portal in the website |
| Actor | Student, Company, University ,admin |
| Initial Status | Logged in |
| Basic Flow | |
| Step1: Click the logout button at the top of the website.  Step2: The portal will be closed. | |
| Post Condition | |
| After clicking logout the portal will close. | |
| Alternative Flow | |
| None. | |

**Table 4) Apply for training.**

|  |  |
| --- | --- |
| Overview | |
| Title | Apply for training |
| Description | The Student after he Signed in, he will fill the form he needed |
| Actor | Student |
| Initial Status | Logged in |
| Basic Flow | |
| Step1: Log in.  Step2: Choose the form needed to fill.  Step3: After filling the form click submit.  Step4: Notification will appear | |
| Post Condition | |
| When you submit you will get a notification counter in the top screen of the website. | |
| Alternative Flow | |
| Enter non-valid data cause alert to appear. | |

**Table 5) Send Form**

|  |  |
| --- | --- |
| Overview | |
| Title | Send Form |
| Description | Used after filling a form to send it on the system |
| Actor | Student |
| Initial Status | Logged in, Form Filled |
| Basic Flow | |
| Step1: Click the submit button after filling the form.  Step2: The companies will be able to view submitted forms. | |
| Post Condition | |
| After clicking submit the form will be sent. | |
| Alternative Flow | |
| In Step1 when clicking submit if a form field is not filled it will prompt to be filled. | |

**Table 6) View Details**

|  |  |
| --- | --- |
| Overview | |
| Title | View Details |
| Description | The Student, Companies can view form details |
| Actor | Student, Companies, admin, university |
| Initial Status | Logged in |
| Basic Flow | |
| Step1: Log in.  Step2: Click details beside your forms to view details. | |
| Post Condition | |
| After clicking details the form information’s will appear | |
| Alternative Flow | |
| None. | |

**Table 7) View Submitted Form.**

|  |  |
| --- | --- |
| Overview | |
| Title | View Submitted Form |
| Description | Show received form by enter student id |
| Actor | Company |
| Initial Status | Logged in |
| Basic Flow | |
| Step1: Log in.  Step2: Choose Student id from the drop-down list.  Step2: After clicking view all the forms will appear to the companies. | |
| Post Condition | |
| Submitted forms will appear. | |
| Alternative Flow | |
| None. | |

**Chapter 4 - System Architecture**

### **4.1 - Introduction about software Architecture**

The software architecture of a system depicts the system’s organization or structure, and provides an explanation of how it behaves. A system represents the collection of components that accomplish a specific function or set of functions. In other words, the software architecture provides a sturdy foundation on which software can be built. A series of architecture decisions and trade-offs impact quality, performance, maintainability, and overall success of the system. Failing to consider common problems and long-term consequences can put your system at risk. There are multiple high-level architecture patterns and principles commonly used in modern systems. These are often referred to as architectural styles. The architecture of a software system is rarely limited to a single architectural style. Instead, a combination of styles often makes up the complete system.

#### **4.2 - The software architecture is** **mvc**: **Model View Controller**

#### **4.3 - we choose it because:**

The actual purpose of MVC is to separate your views from your controller and model. In other words, it is a design pattern is a structure for keeping display and data separate to allow each to change without affecting the other. By saying so, it is mostly used for GUI stuffs.

You should use an architecture that separates logic from your views. If needed, you should use an architecture that utilizes a controller (such as MVC) if there is logic required that doesn't necessarily fit into a model

### **The Benefits of MVC architecture**

1. Development of the application becomes fast.
2. Easy for multiple developers to collaborate and work together.
3. Easier to update the application.
4. Easier to Debug as we have multiple levels properly written in the application.
5. Re-usability of code, easy to maintain code and maintenance.
6. The best thing is the developer feels good to add some code in between the project maintenance.

**4.4 - Description of the MVC** **in detail**

Three important MVC components are:

* Model: It includes all the data and its related logic
* View: Present data to the user or handles user interaction
* Controller: An interface between Model and View components

Let’s see each other this component in detail:

### **Model**

The model component stores data and its related logic. It represents data that is being transferred between controller components or any other related business logic. For example, a Controller object will retrieve the customer info from the database. It manipulates data and sends back to the database or uses it to render the same data.

It responds to the request from the views and also responds to instructions from the controller to update itself. It is also the lowest level of the pattern which is responsible for maintaining data.

Model, contain classes and within those classes are data stored in them.  
and the models we used in the project : (users),(company),(university),(usertype),(CodeUnivesity),(transaction),(Code nationality),(Courses),(Student master),(Code Company)

### **View**

A View is that part of the application that represents the presentation of data.

Views are created by the data collected from the model data. A view requests the model to give information so that it resents the output presentation to the user.

The view also represents the data from charts, diagrams, and tables. For example, any customer view will include all the UI components like text boxes, drop downs.

The view will show the page in the website and the views we used in the project:

(users),(company),(university),(usertype),(CodeUnivesity),(transaction),(Code nationality),(Courses),(Student master),(Code Company)

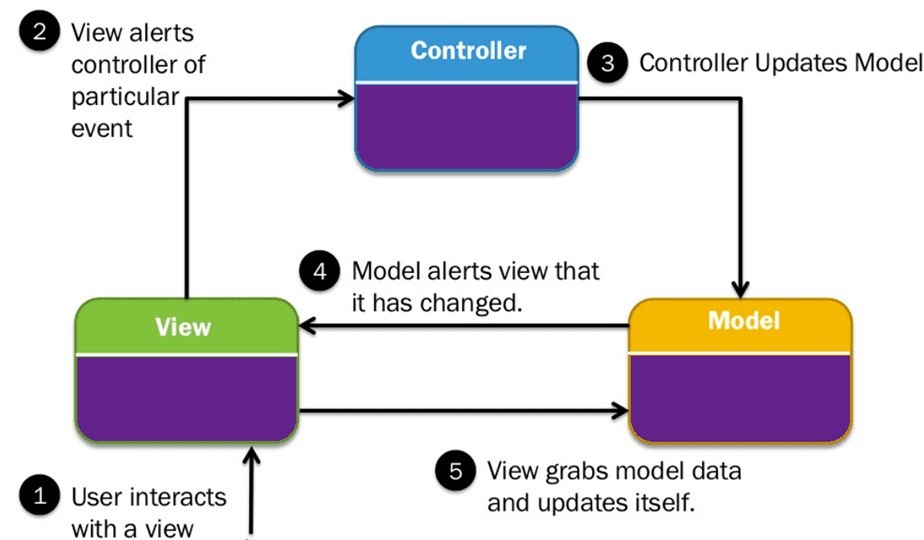
### **Controller**

The Controller is that part of the application that handles the user interaction. The controller interprets the mouse and keyboard inputs from the user, informing model and the view to change as appropriate.

A Controller send’s commands to the model to update its state (E.g., Saving a specific document). The controller also sends commands to its associated view to change the view’s presentation (For example scrolling a particular document).

The controller is the part where we use the backend where we use methods to create or delete or edit and here is some of the controllers we used in the project:

(users),(company),(university),(usertype),(CodeUnivesity),(transaction),(Code nationality),(Courses),(Student master),(Code Company)

****

## Figure (4.1): **(MVC)** Architecture

**Chapter 5 - System Design**

### **5.1 - Introduction about System Design**

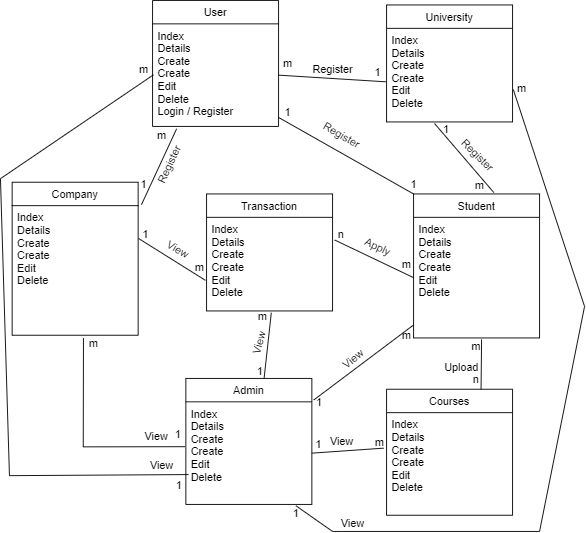
Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. So here we are going to talk about the class diagram, database table, and user interface design.

### **5.2 - Class diagram**

As before here will see the class diagram for the project that is the main building block of object-oriented modeling. It is used for general conceptual modeling of the structure of the application and for detailed modeling thus it can be used for data modeling, as demonstrated in Figure (4).

**Here we will explain the Uml diagram:**

User’s will be able to register or login to the website and will be able to see the courser and jobs submitted by the companies , university can help the student through suggesting suitable courses to take or help them with job opportunity , with transaction the student can apply for a likable course to take.

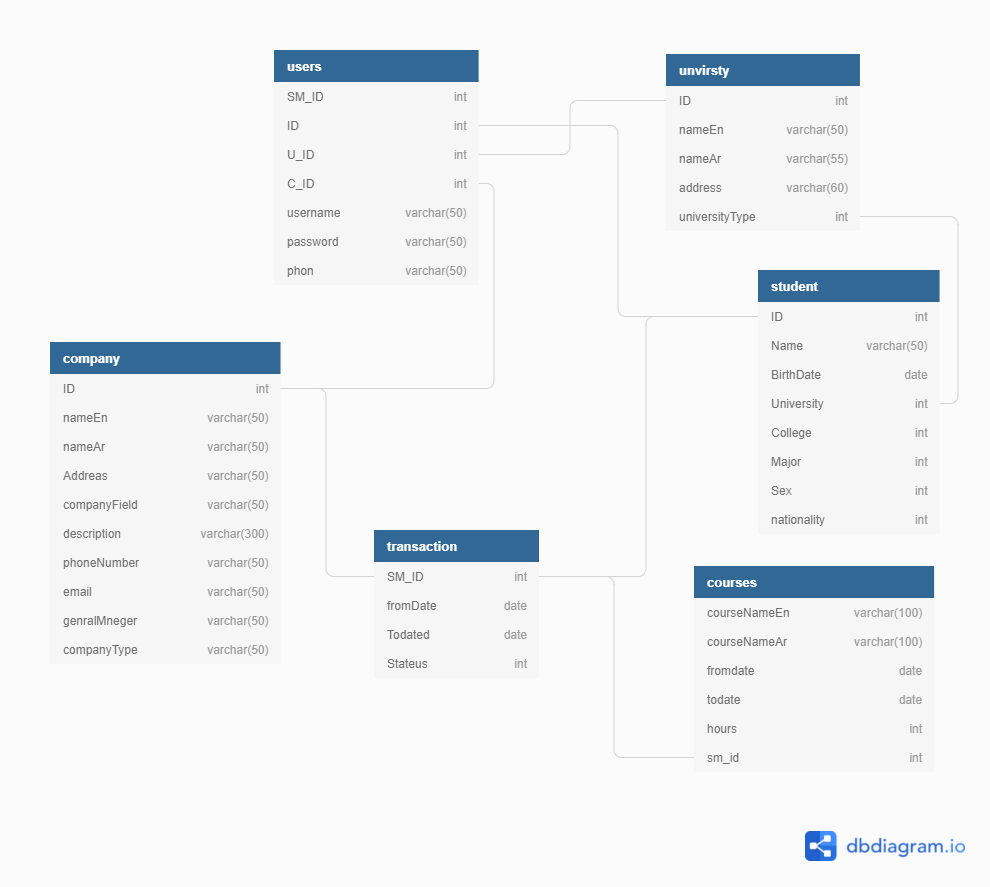


**Figure (5.1): Class Diagram**

### **5.3 - Relational Database Table**

A relational database is a digital database based on the relational model of data. A software system used to maintain relational databases is a relational database management system. That said we will see the relational database for this project in Figure (5).

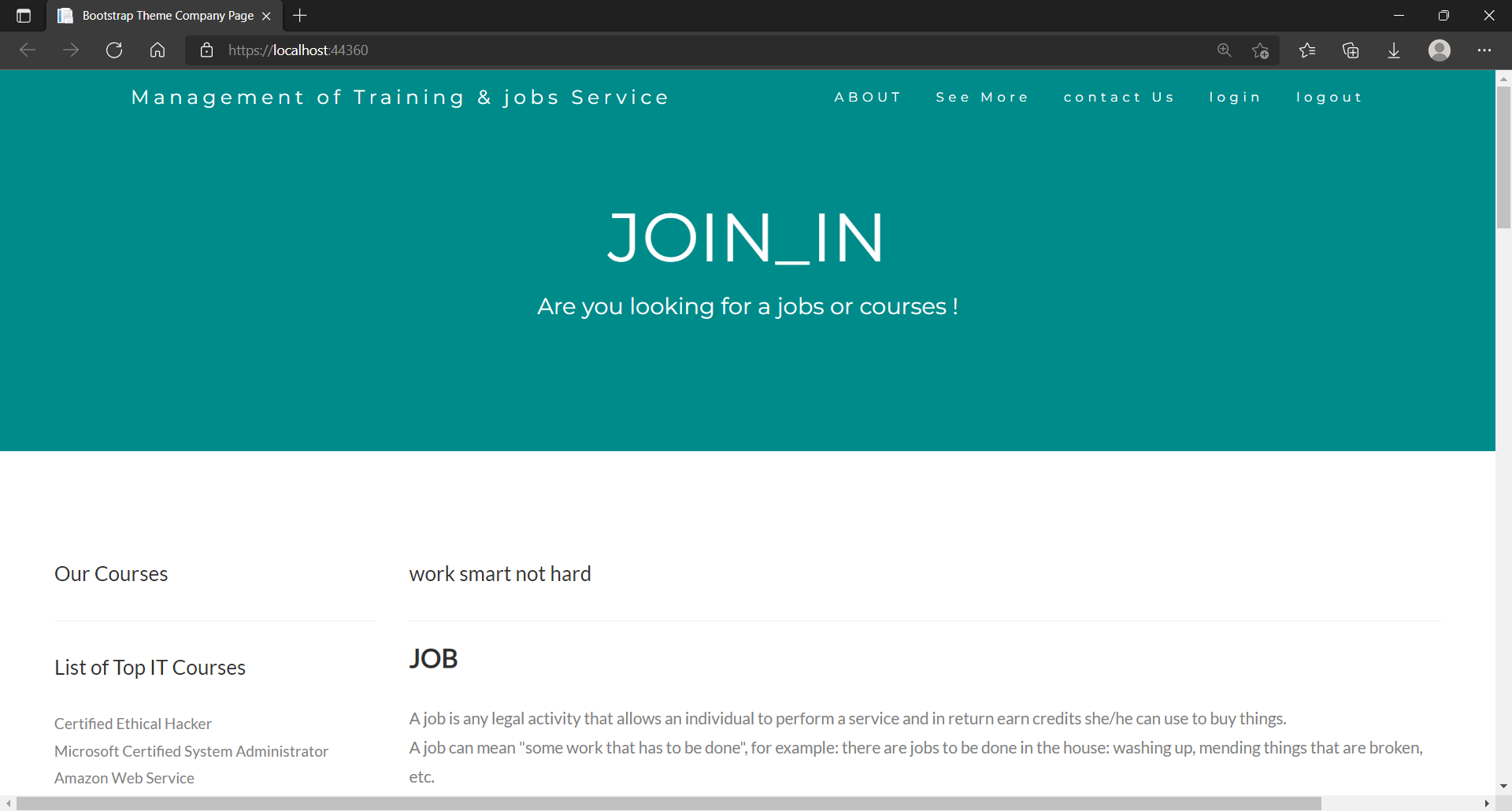
This figure explain the Relational Database: A relational database is a type of database that stores and provides access to data points that are related to one another. Relational databases are based on the relational model, an intuitive, straightforward way of representing data in tables. In a relational database, each row in the table is a record with a unique ID called the key



## Figure (5.2): Relational Database Table

### **5.4 - User Interface Design**

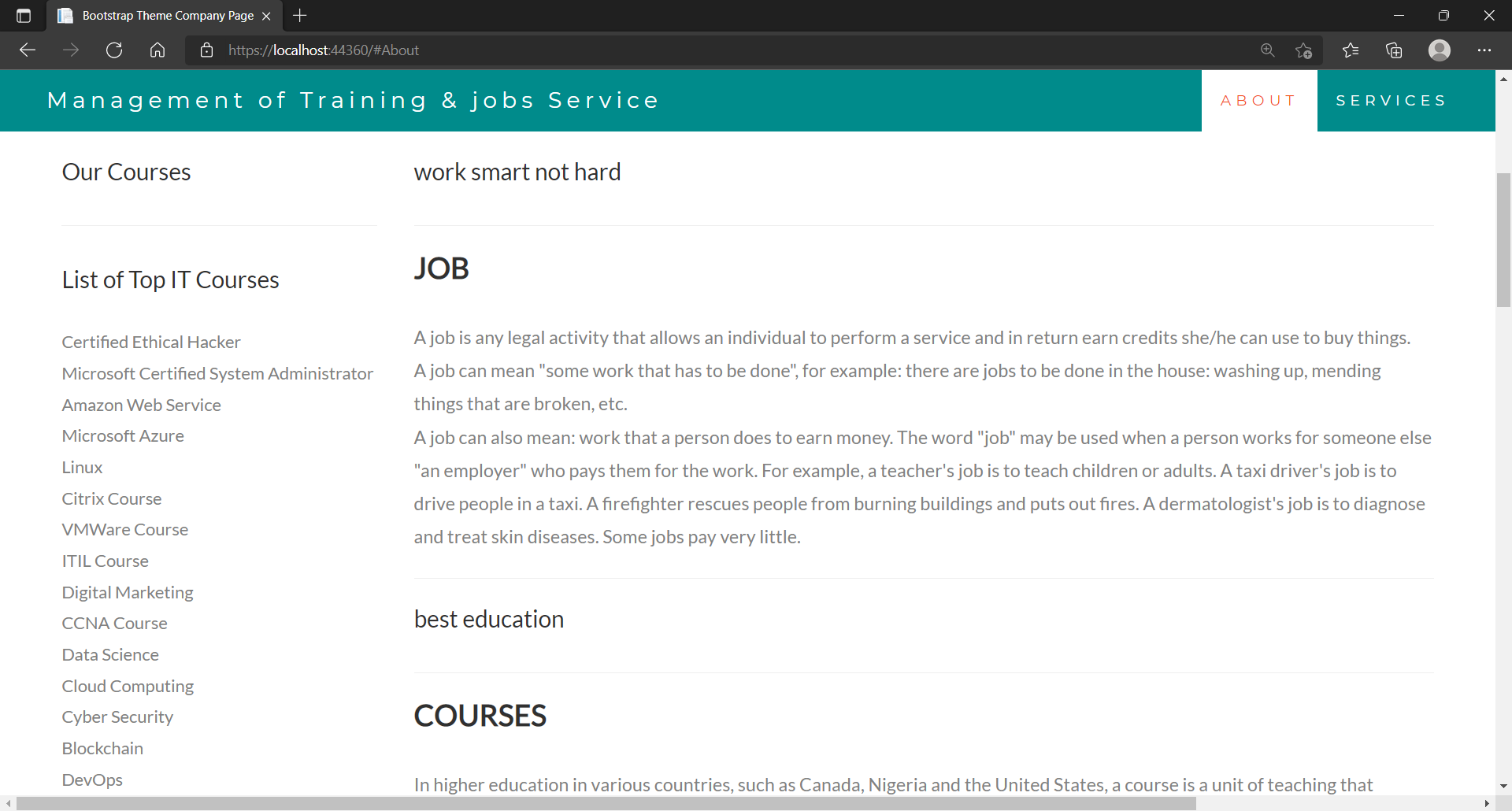
User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand and to use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture as demonstrated in Figures (6-11): User Interface Design.



## 

## Figure (5.3): User Interface Design (Home Page)

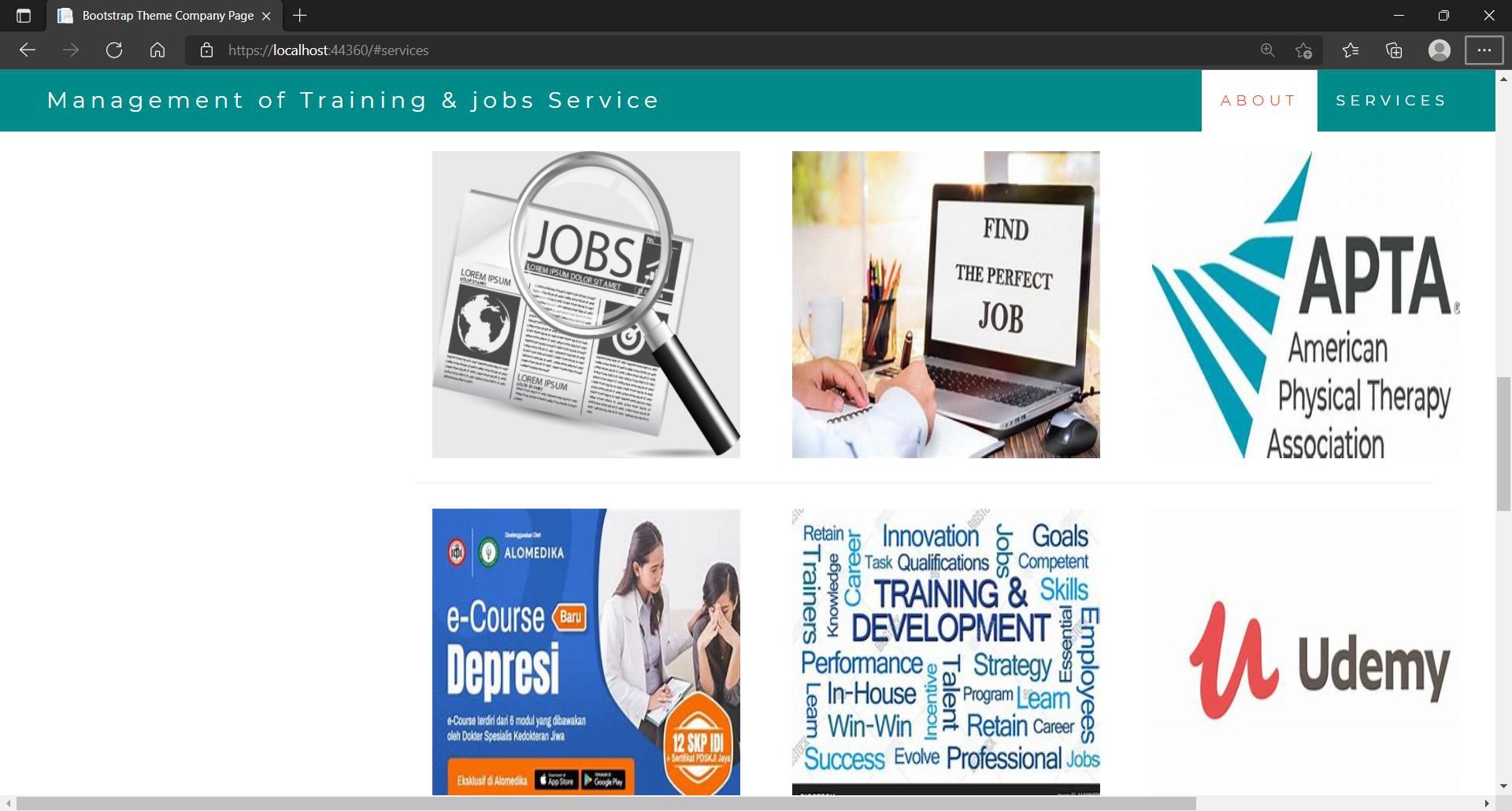
This is the home page to the website where all menus and interactions are on.



## 

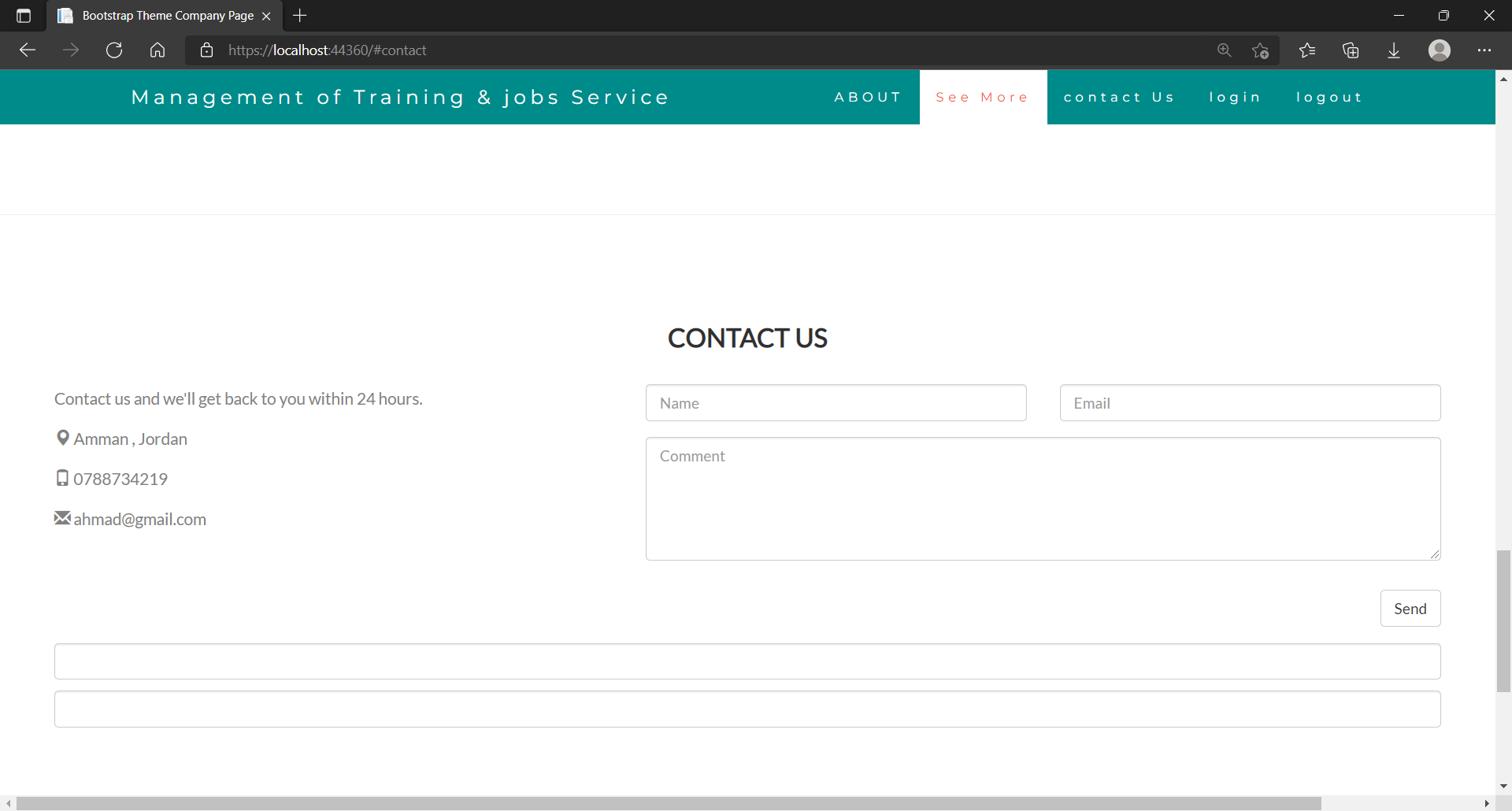
## Figure (5.4): User Interface Design (About Page)

In this page you can read more about our website and get more information about the activities we have available on the website with a lot more information you might want to take a look at.



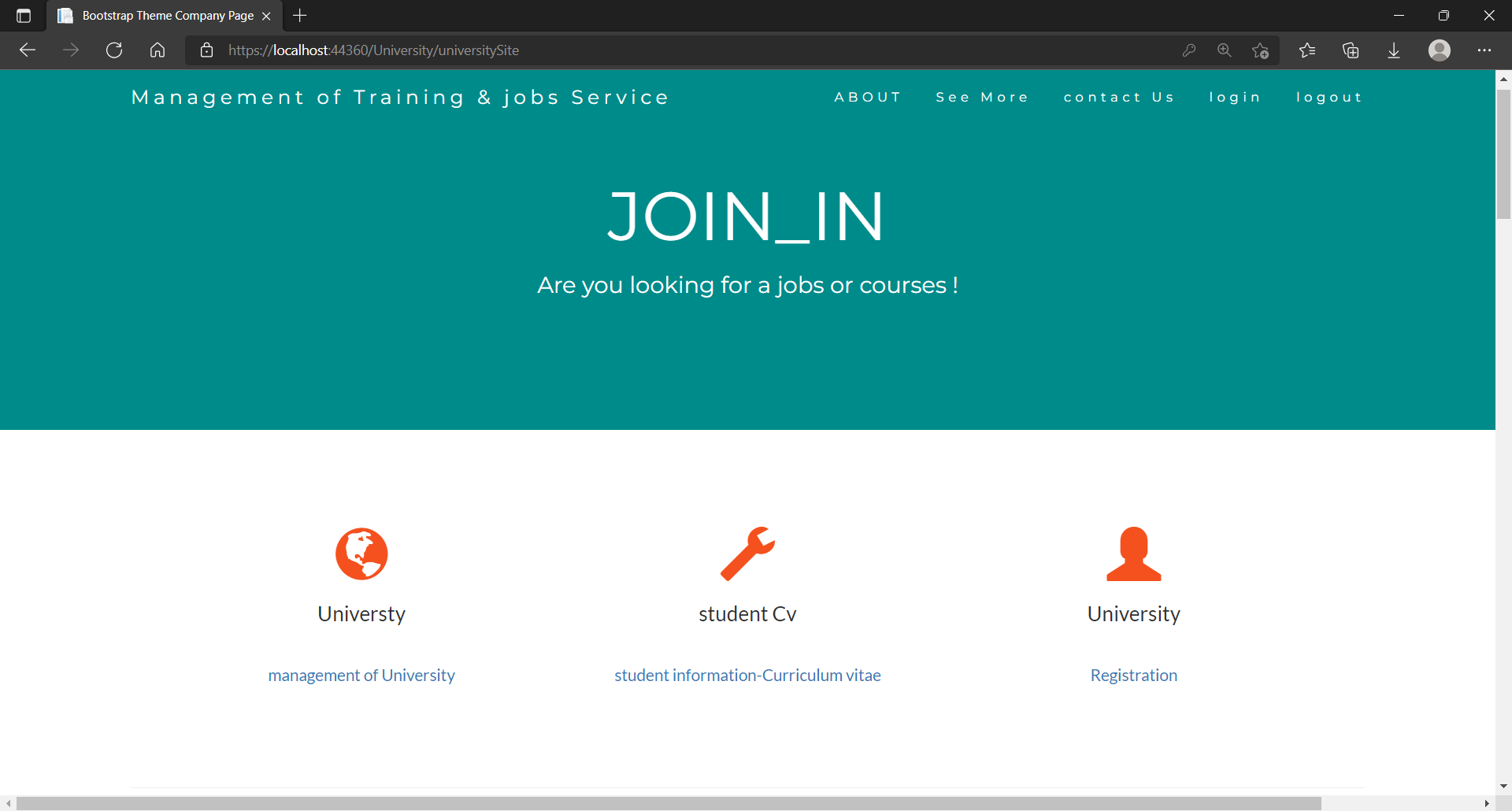
## Figure (5.5): User Interface Design (About Page)

In this page you can read more about our website and get more information about the activities we have available on the website with a lot more information you might want to take a look at.



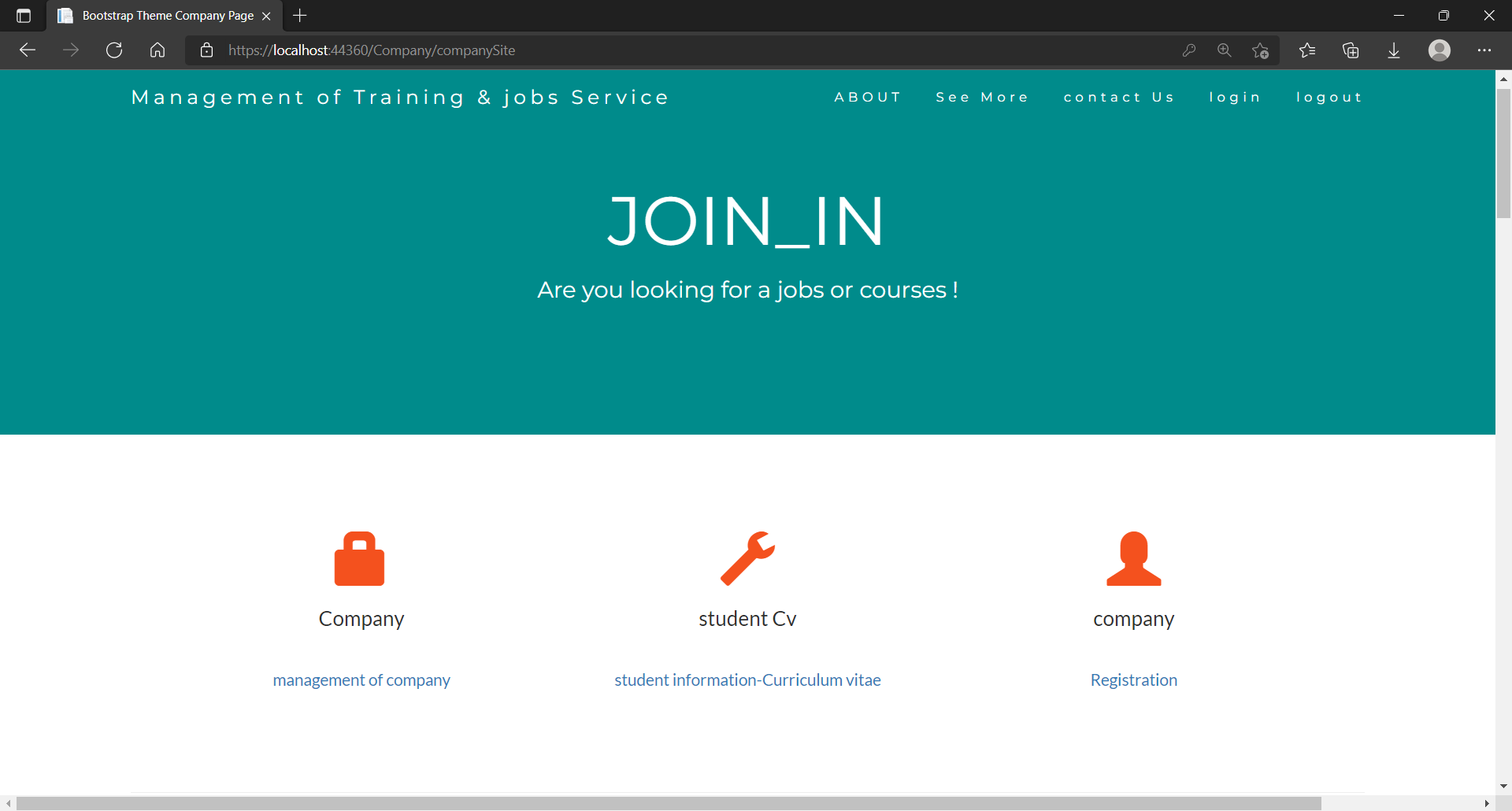
## Figure (5.6): User Interface Design (contact us Page)

In this page you can contact us directly with any concerns or requests.



**Figure (5.7): User Interface Design (University Page)**

In this page you can see all the universities registered to our website.



## Figure (5.8): User Interface Design (Company Page)

In this page you can see all the Companies registered to our website.

**Chapter 6 - Implementation**

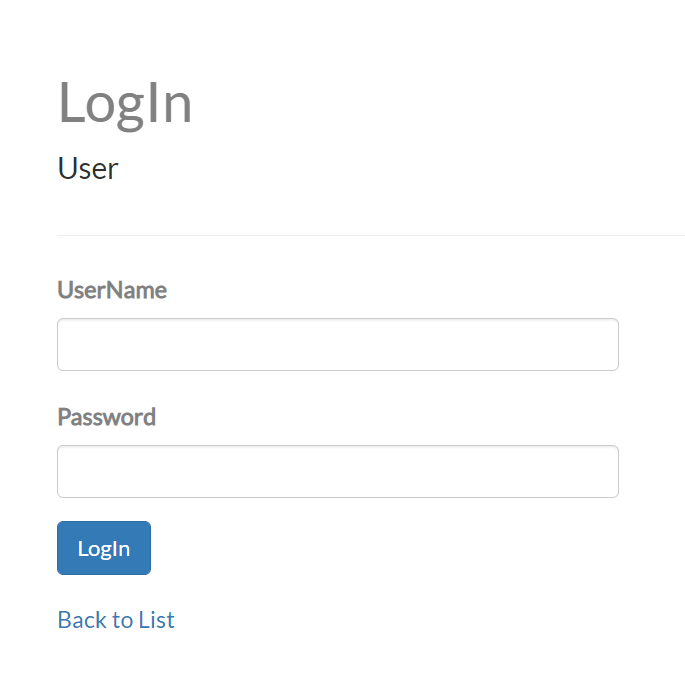
### **6.1 - Introduction about Implementation**

Software implementation refers to the process of adopting and integrating a software application into a business workflow. Implementation of new tools and software can be complex, depending on the size of the software so in this chapter we are going to demonstrate some of the implementation used in our project.

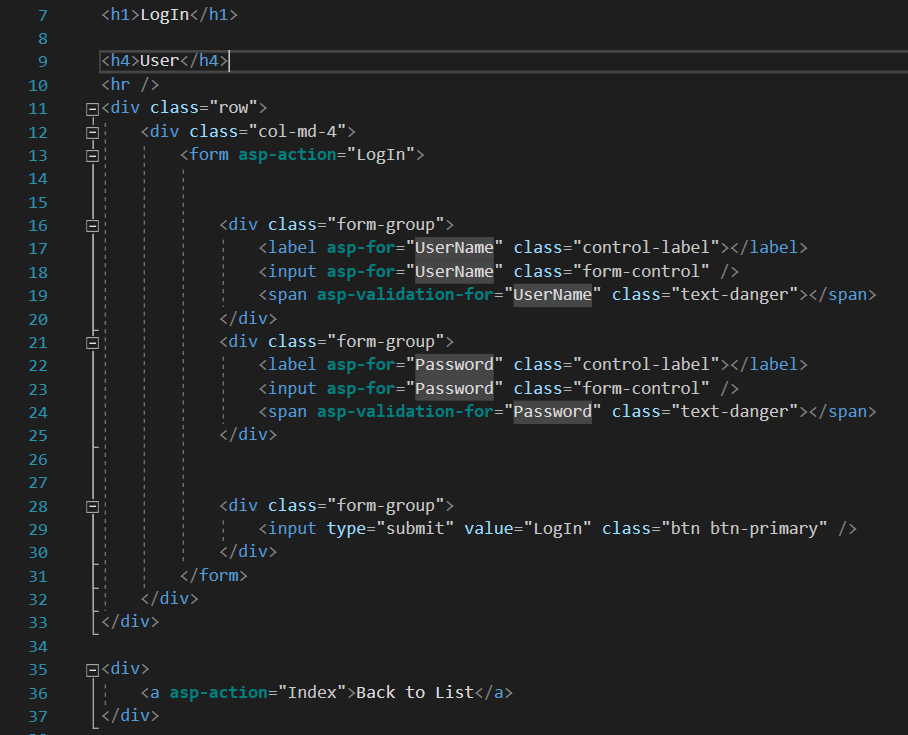
### **6.2 - Implementing the Login Method**

We need a login form for our websites users to interact with and enter their details. So first we

Have our index.html file that will be the access portal to the user



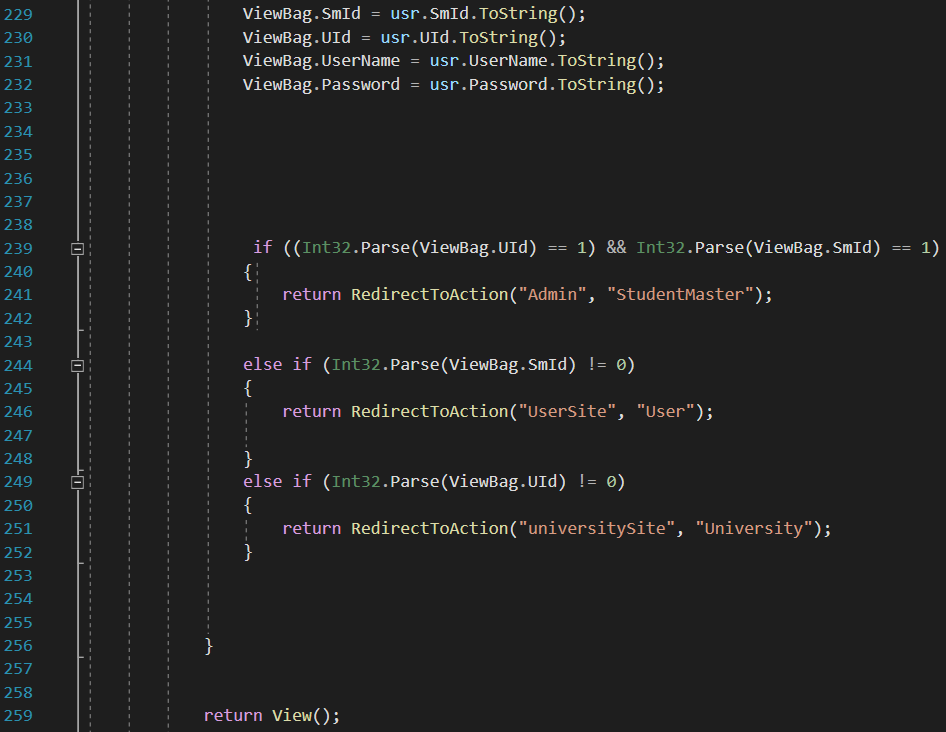
**Figure (6.1):** **Implementation (Login Page Design)**



**Figure (6.2):** **Implementation (Login Page Code)**

#### **6.3 - Implementing the Login Authentication Method**

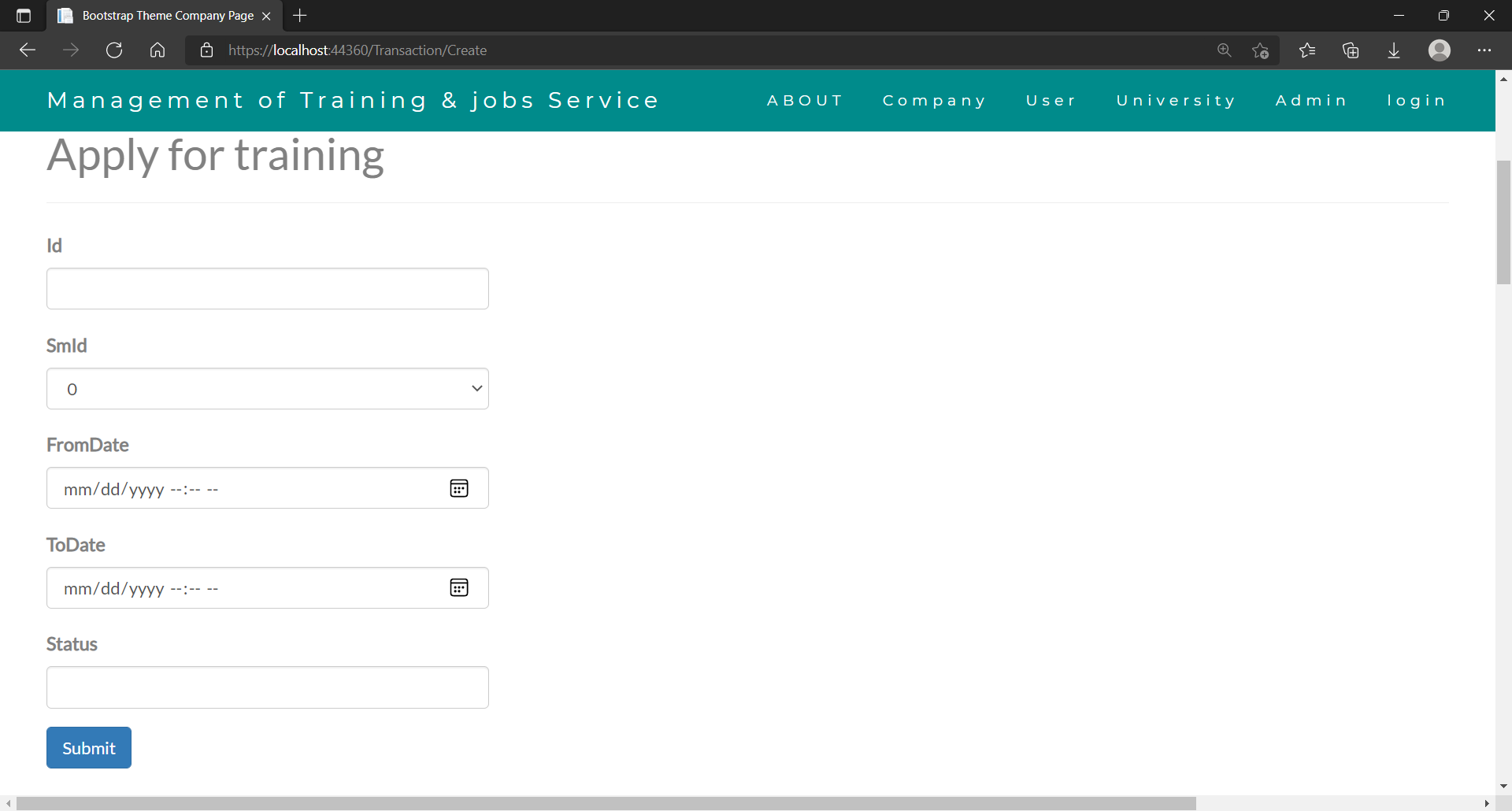
Now after the user enters his login data a login controller will handle the request in order to see if the user’s access is granted.



## Figure (6.3): Implementation (Login Authentication Code)

### **6.4 Implementing Apply For Training Form Method**

Now the main purpose of the project is that the student applies for a training form to the elite companies through our website



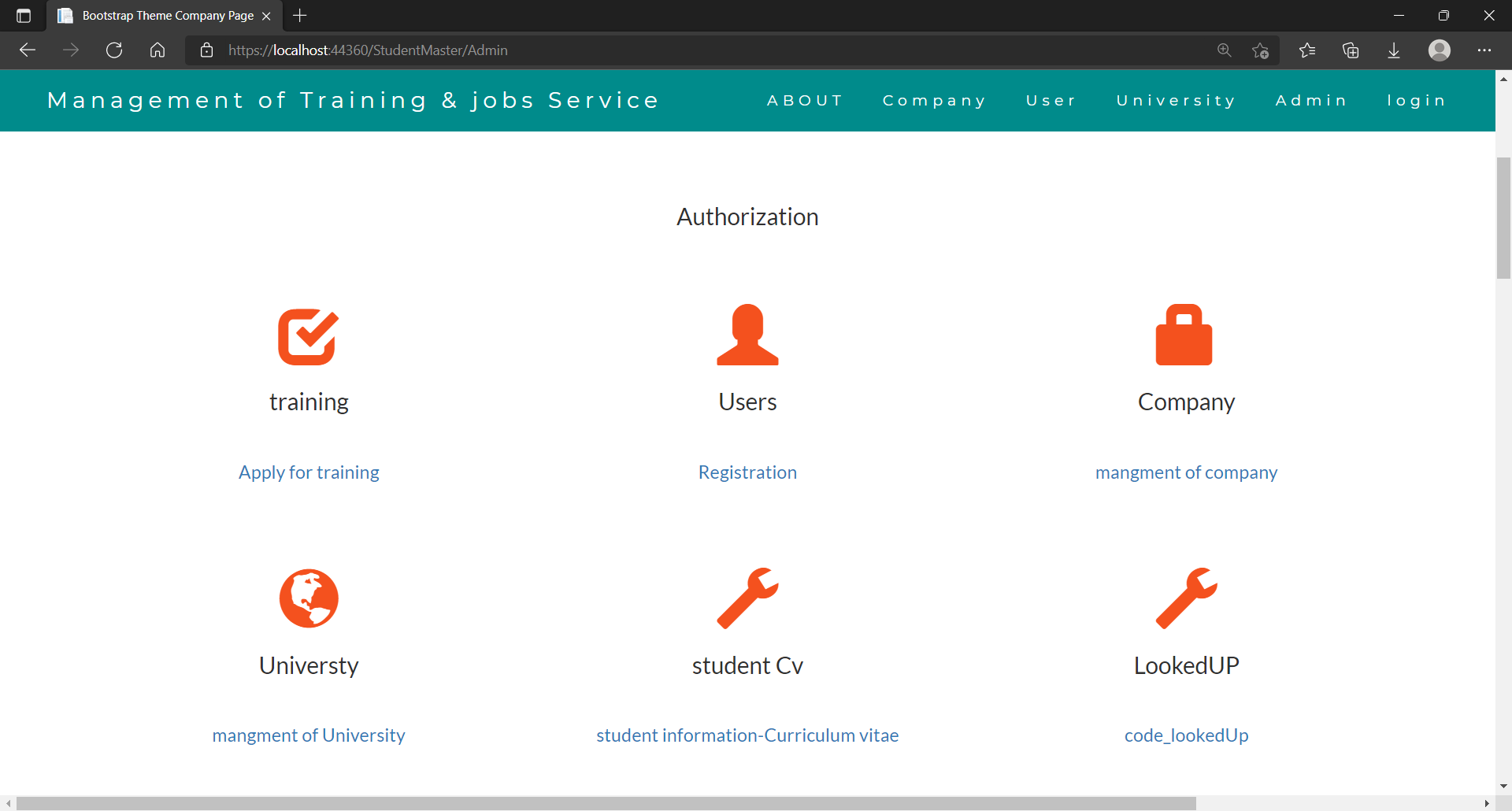
**Figure (6.4): Implementation (Form Applying Design)**



**Figure (6.5): Implementation (Form Applying Code)**

### **6.5 Implementing the Admin Portal Home Page**

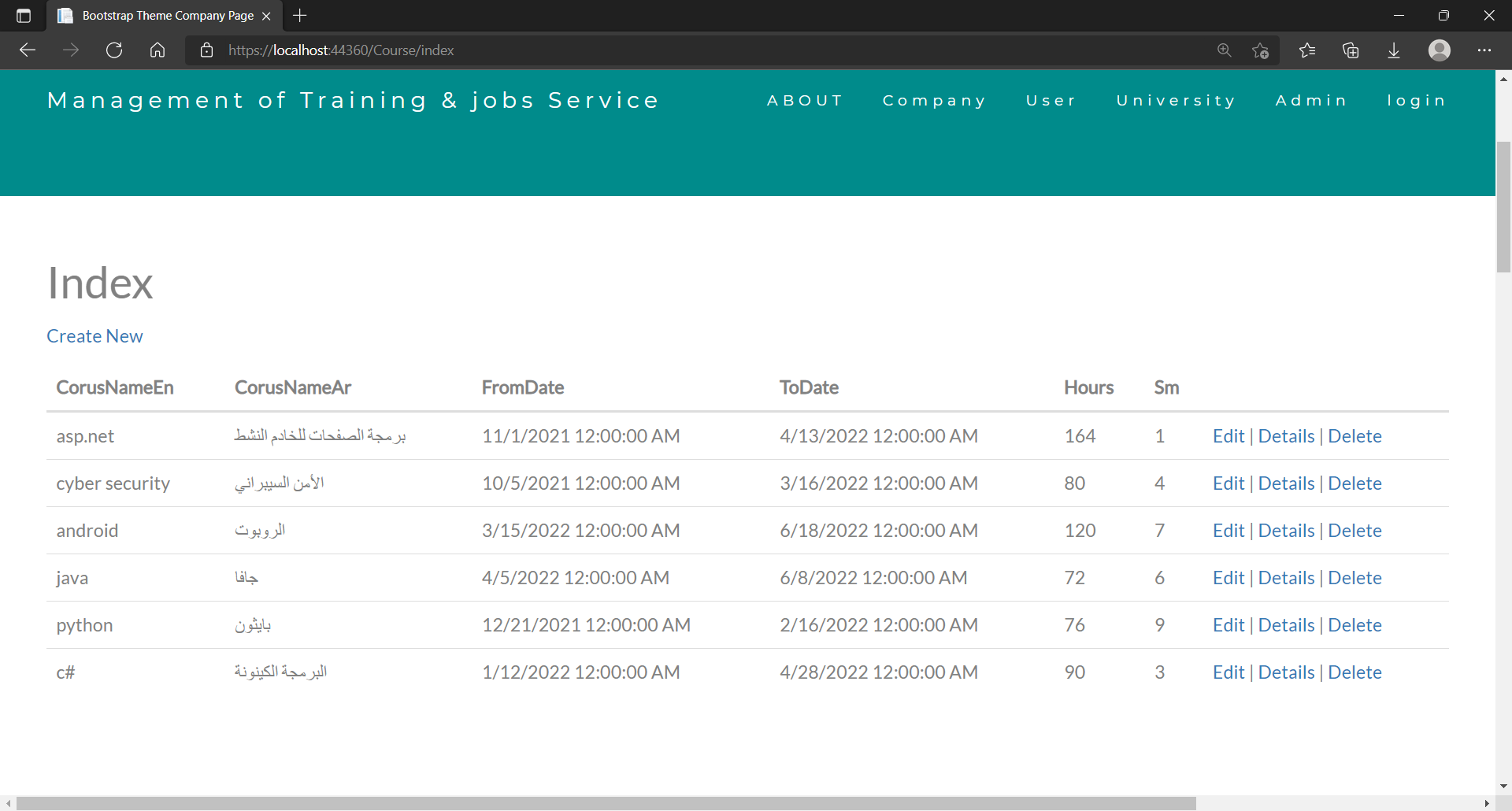
Through this form we can view what type of authorizations the admin has on our website.



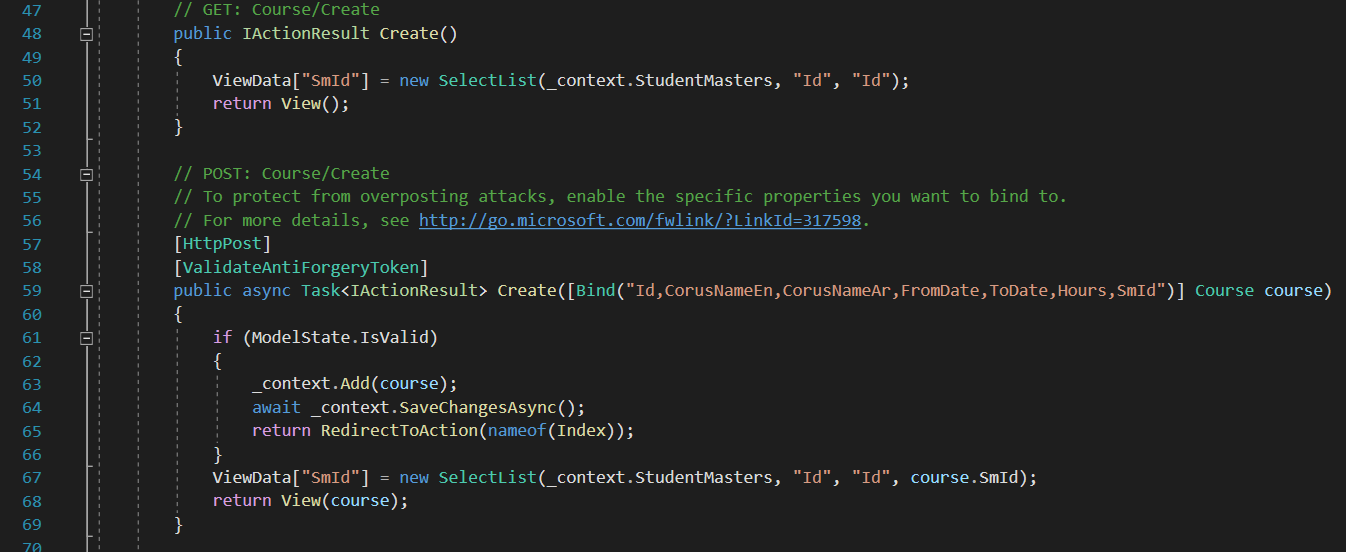
**Figure (6.6): Implementation (Admin Portal)**

### **6.6 Implementing the Courses Portal**

Through this form we can view what type of courses the students have.



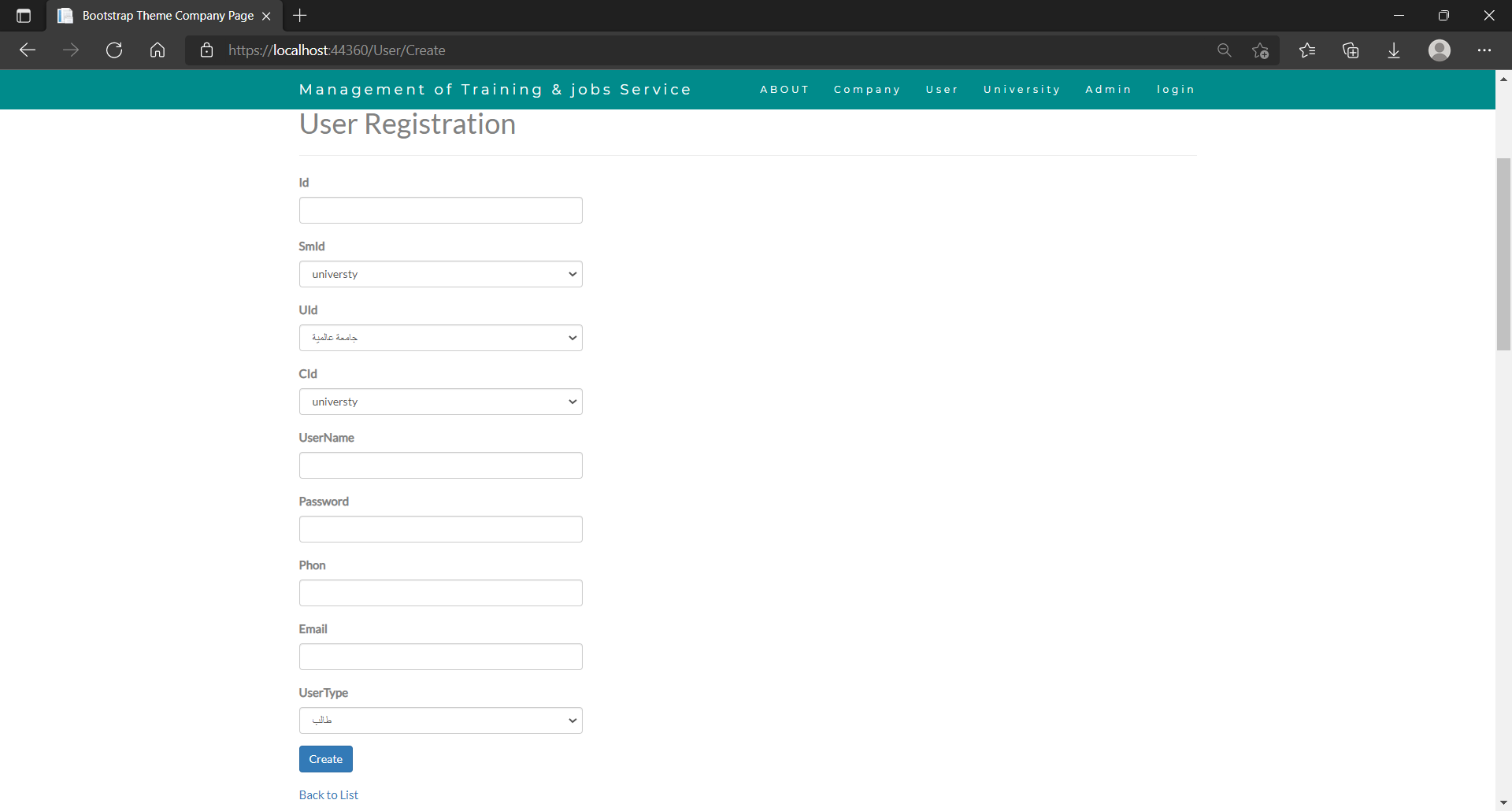
**Figure (6.7): Implementation (Courses** **Portal)**



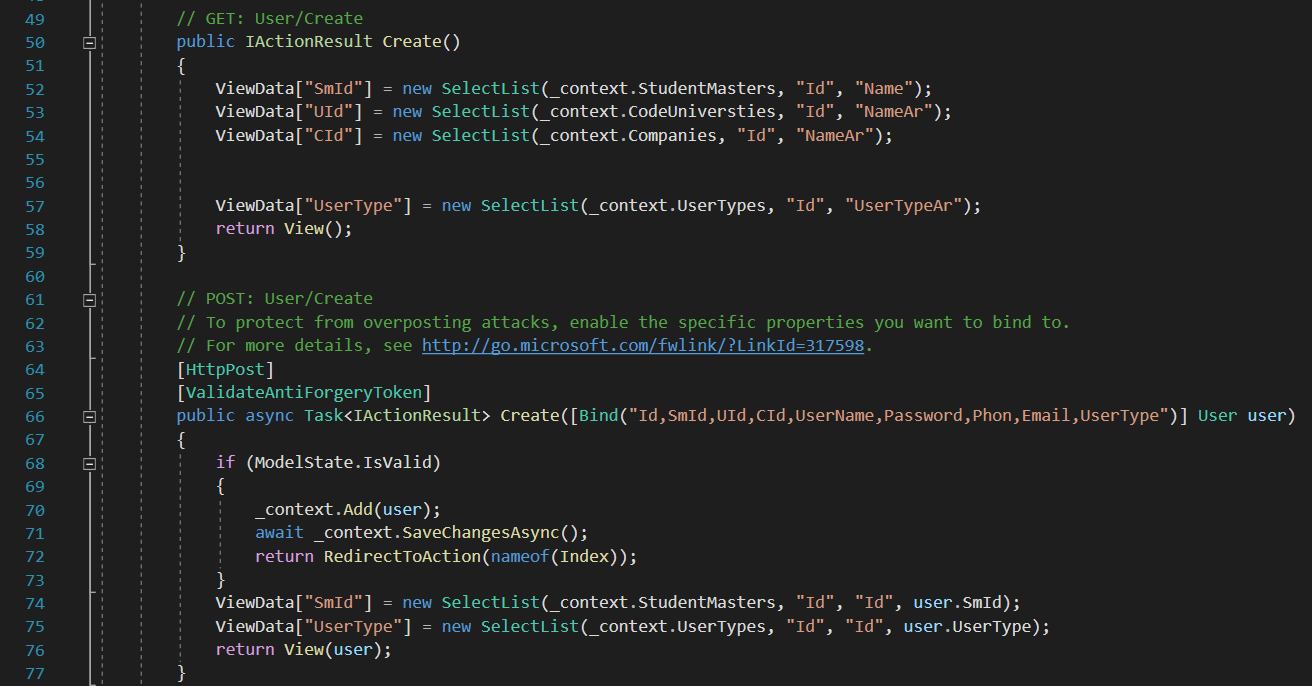
**Figure (6.8): Implementation (Courses** **Portal)**

### **6.7 Implementing the Registration page**

We need a Registration in form for our websites for the new users.



**Figure 6.9): Implementation (user Registration page)**



**Figure (6.9.1): Implementation (user Registration page)**

**Chapter 7 - System Testing**

### **7.1 - Introduction about System Testing**

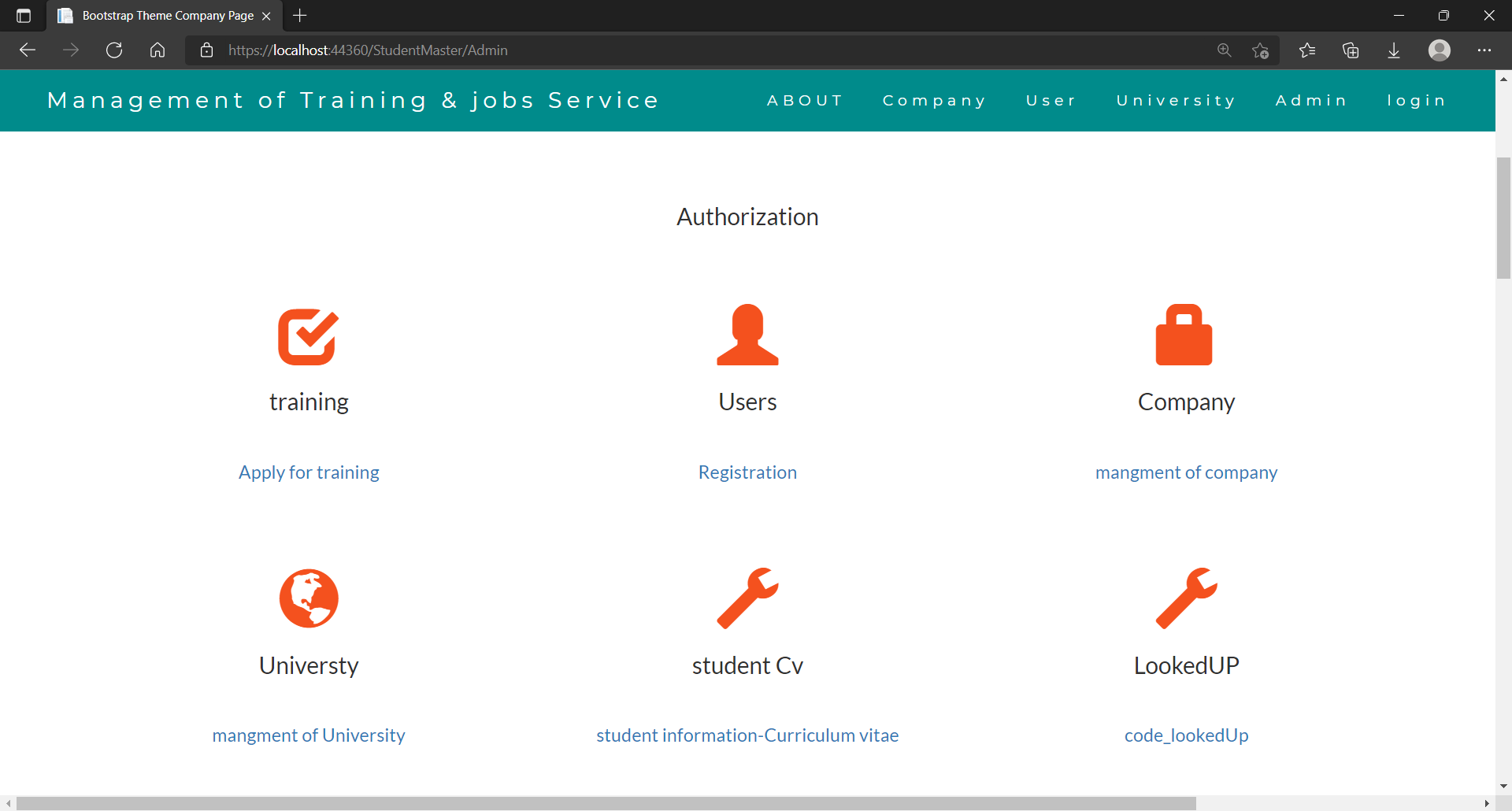
Software testing is nothing but an art of investigating software to ensure that its quality under test is in line with the requirement of the client. Software testing is carried out in a systematic manner with the intent of finding defects in a system.

### **7.2 - Black Box Testing**

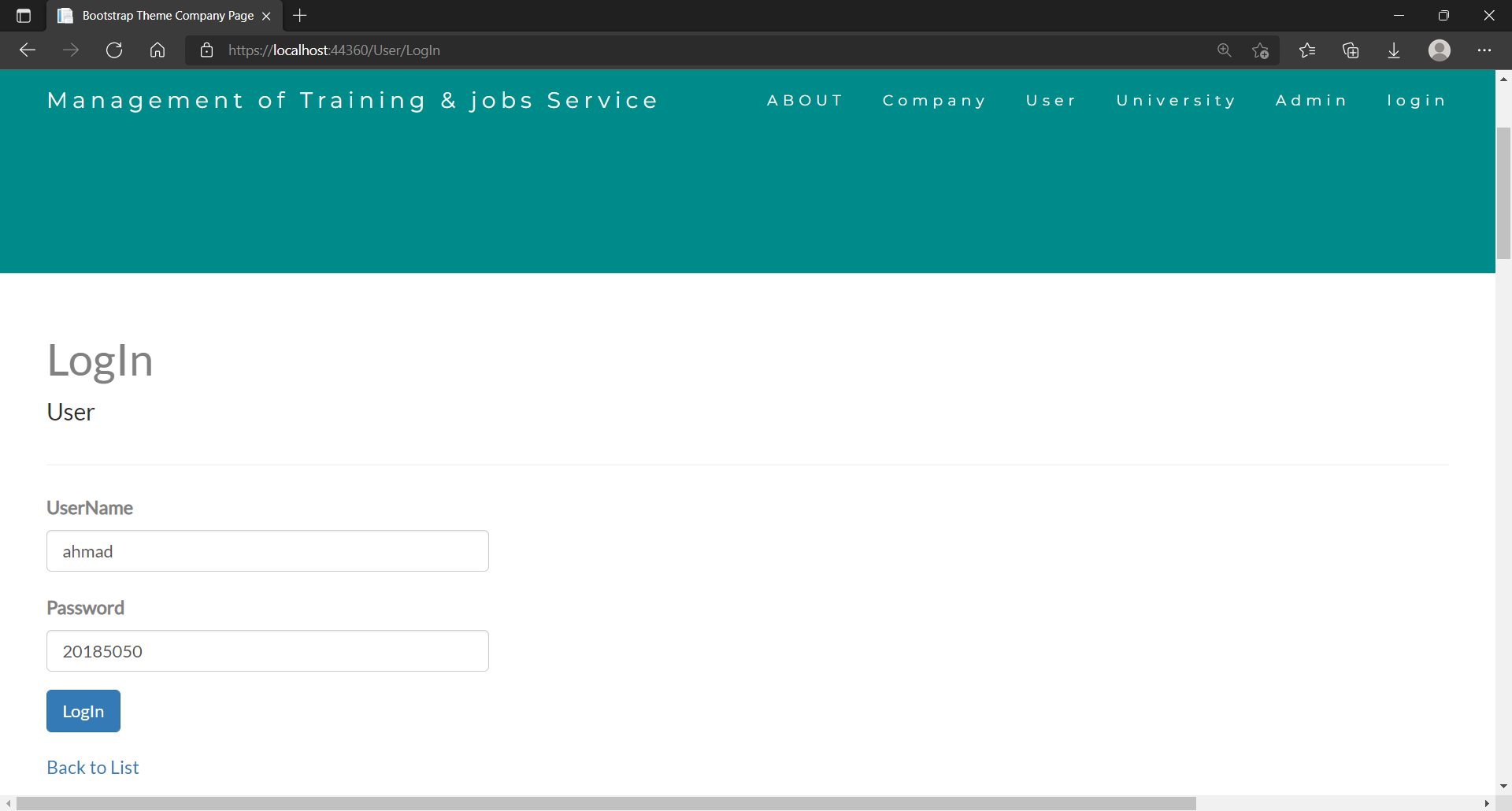
Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. Black box testing is a powerful testing technique because it exercises a system end-to-end. In the next table we will see some tests performed on the website followed with the images of the expected outputs as written in the table.

**Table (7.1): Black Box (Testing)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **Description** | **Input** | **Expected Output** | **Pass / Fail** |
| **1** | Testing Login | Username: Ahmad  Password: 2018200 | Condition (Login  Successful) | Passed |
| **2** | Testing Login | Username: Ahmad Password: 20185050 | The admin page will not open | Passed |

****

## Figure (7.1): Black Box (Testing Case #1)



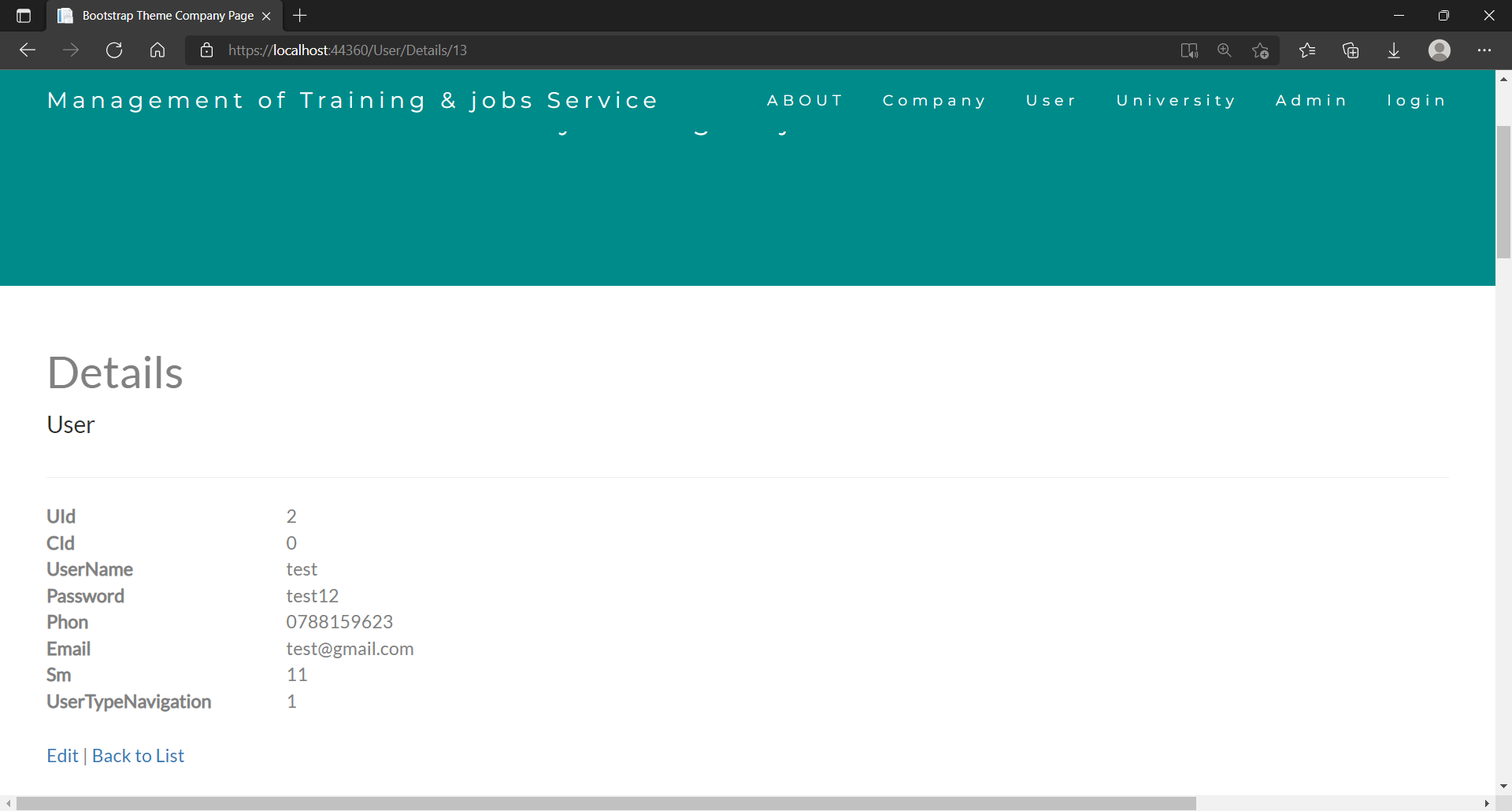
## Figure (7.2): Black Box (Testing Case #2)

### **7.3 - Testing by Using**

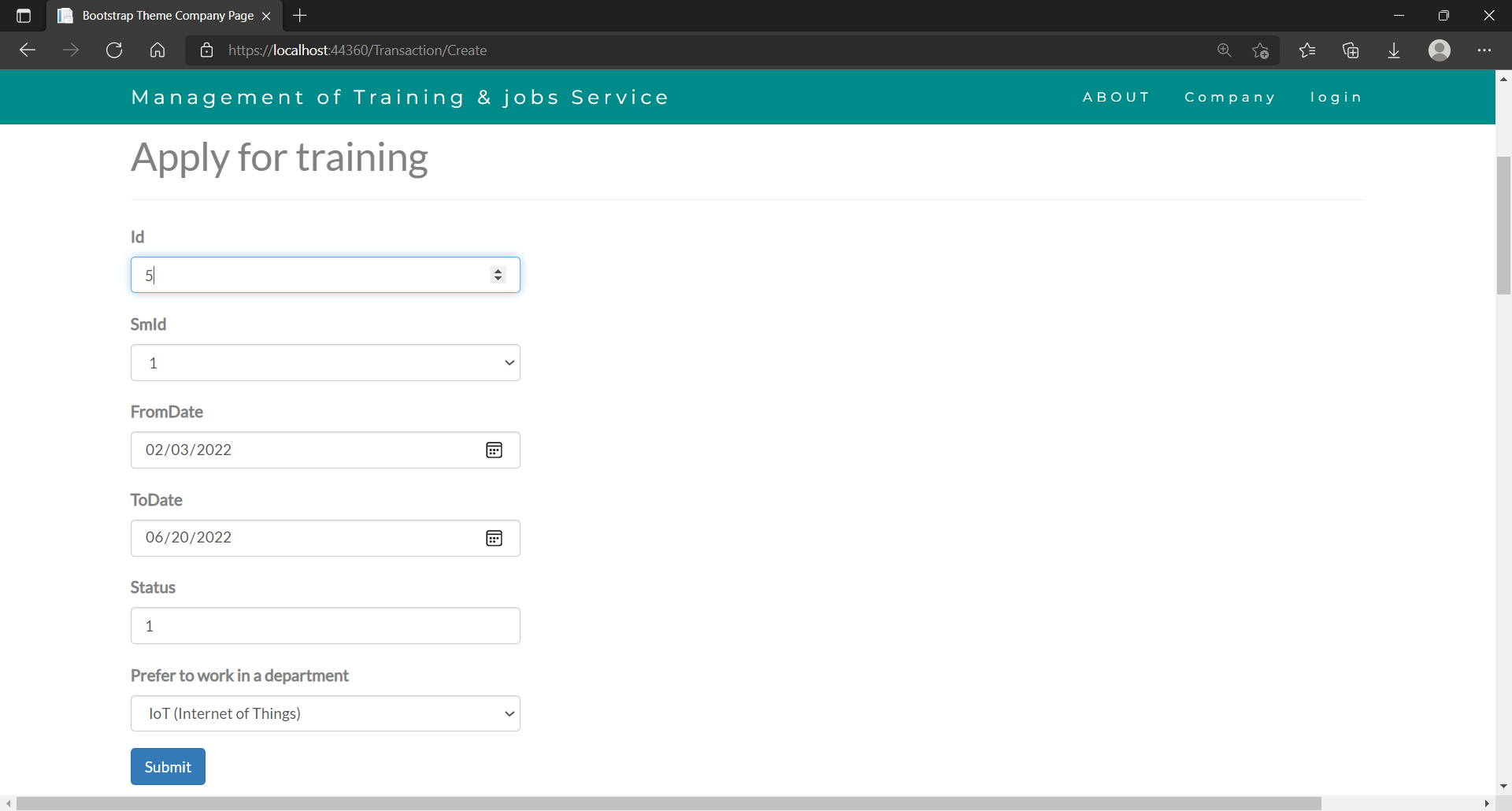
To perform this test, multiple usage scenarios were given and described. A set of testing data were written for each scenario and the data was matching the real-world examples. The operations were performed separately by different users on different machines. Following are samples of the testing data.

**Table (7.2): Testing by Using (Samples)**

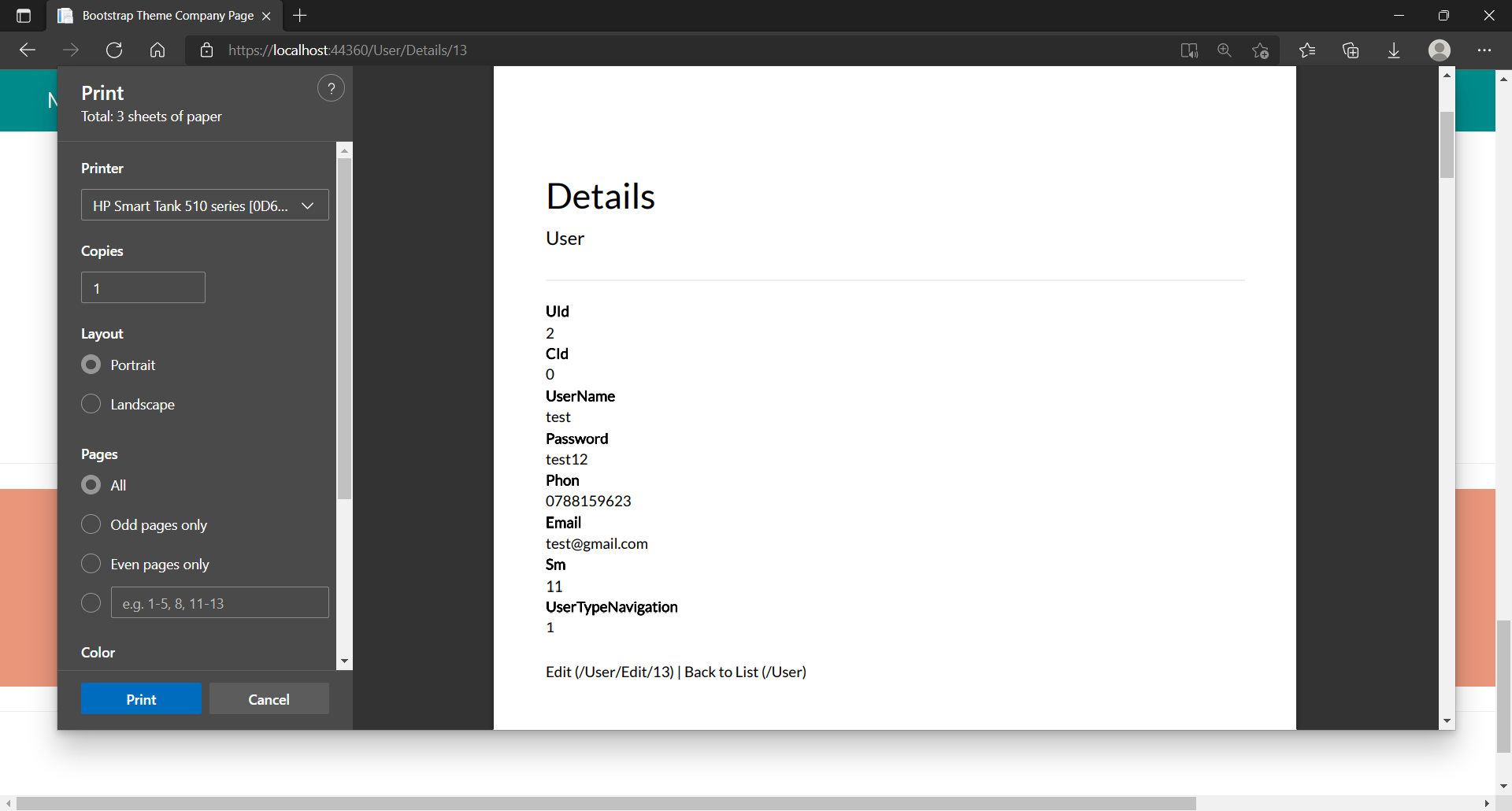
|  |  |
| --- | --- |
| **Operation** | **Test data** |
| Registrations | Subject: Registration  Description: Creating account named : Test  With password : Test12 |
| Apply for training | Course Name: IOT(internet of things) |
| Print Forms or page’s | Printing a filled form or Page |



**Figure (7.3) Test by Using**



**Figure (7.4) Test by Using**



**Figure (7.5) Test by Using**

#### **7.3.1 - Testing by Using Results**

## Table (7.3): Testing by Using (Results)

|  |  |
| --- | --- |
| **Operation** | **Result** |
| Registration | Registration Successfully |
| Apply for a Course | Applied Successfully |
| Print a form or a page | Printed Successfully |

**Table (7.4): Check List of Software Testing**

|  |  |  |
| --- | --- | --- |
| **#** | **Question** | **Answer** |
| 1 | Does it really work as expected? | Yes |
| 2 | Does it meet the user's requirements? | Yes |
| 3 | Is it what the users expect? | Yes |
| 4 | Do the users like it? | Yes |
| 5 | Is it ready for release? | As a beta yes, as a full release no |
| 6 | How well does it work? | Excellent |
| 7 | What does it mean to you that “it works”? | It met all requirement specifications and functions without errors |
| 8 | How do you know it works? What evidence do you have? | Testing done with excellent results |
| 9 | In what ways could it seem to work but still have something wrong? | N/A |
| 10 | What might cause it to not to work well? | N/A |

**Chapter 8 - Conclusion & Future work**

### **8.1 - Conclusion**

**Management of Training & jobs Service** is an effective and useful website it’s designed to help the students to apply for the elite companies to be trained at with ease, students can also upload their CV’s for companies to view. And the student have better chance’s to get trained in the elite companies by applying through our website because universities are connected to both registered students and available companies.

### **8.2 - Future work**

More work could be done to change on the home design also adding more than one language to the site as well as improve the security system to make sure that all the information's on the system are safe from hacking , more portals and pages will be added to the site . Develop the site to contain job for students, also, competitions and awards to support students' CV. And to make sure that the website will be used by a lot of people we going to do a mobile applications,

The Portals can also be upgraded to include a private chat system between the student and companies to insure a good communication, a page will be added that include all Books and references a student might need.

**References**

* Refsnes Data. “W3Schools.” Internet: https://www.w3schools.com/, 1998.
* Mark Otto, Jacob Thornton. “Bootstrap.” Internet: https://getbootstrap.com, August.19,2011.
* Brendan Eich. “JavaScript” Internet: https://www.javascript.com/, December.4,1995.
* John Resig, Brandon Aaron, Jörn Zaefferer. “jQuery.” Internet: https://jquery.com, August.26,2006.
* Jeff Atwood, Joel Spolsky. “Stack Overflow.” Internet: https://stackoverflow.com, September.15,2008.

**Appendix**

**Project Submission**

1. **Supervisor:** dr.mohammad Taye
2. **Project Title:** Management of Training & jobs Service
3. **Goals and Objectives :**
4. Enabling students to get training in the best companies
5. Enabling companies to train and appoint students
6. Enabling universities to implement their study plans by facilitating the training of their application
7. Providing jobs after training or graduation
8. enable students to take courses (self-learning) to made them ready for working in company
9. **Brief description of the project :**

(Web application) this project will enable the student to sign the best courses by allowing companies to offer the required specialties to the labor market

Then the student will know the requirement of labor market and what to choose  
and will Enabling universities to implement their study plans by facilitating the training of their application and student will easily find a job after graduation

1. **References**: Reid Hoffman’s. LinkedIn , Internet : https://www.linkedin.com/

May 5, 2003 [October 10th , 2021]

1. **Project Requirements (Hardware & Software)** : in this project we will use

Asp.net mvc : The ASP Technology

ASP and ASP.NET are server side technologies.

Both technologies enable computer code to be executed by an Internet server.

When a browser requests an ASP or ASP.NET file, the ASP engine reads the file, executes any code in the file, and returns the result to the browser.

And Database (using sql server):

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

1. **Company** **or organization (If applicable)** there is no Company or organization
2. **Prerequisite** Department approval + 90 hours
3. **Project Specialization (**Software Engineering)
4. **Time schedule :**

(2-3 months): the project will be ready at the 5th of February

Table work:

|  |  |  |  |
| --- | --- | --- | --- |
| Ahmad | Defining the requirements | Design | Implementation  Verification |
| Abdullah | Defining the requirements | Design | Validation |

**1. Defining the requirements**

At this stage of work the team conducts a meeting with the customer and discusses all his requirements to the final product. Waterfall methodology is based on strict documentation, so all client’s demands are written down. Waterfall teams do not contact the customer during the period of project realization, so the next meeting is conducted after the final product is ready.

**2. Design**

At this stage the developers should design the architecture of the future software. Unlike other software development methodologies, in Waterfall this process has a great significance for the future work. That is because of the fact that Waterfall teams do not change the plans of their projects during their realization. Hence, the architecture of the future software product is documented and the team is ready to next stages of the life cycle.

**3. Implementation**

This is the stage of code creation. It is the main stage of every software development project. For Waterfall projects it is extremely important because of the inability to return to the previous stages if something went wrong. That is why Waterfall developers should work hard to avoid bugs and other issues in the software product’s code.

**4. Verification**

This is the stage of product testing. The testing life cycle in Waterfall is what we call an example of product testing in software development. The code, developed at the previous stage of work is tested. After that the defects are fixed and it is tested one more time. If the code contains defects that cannot be fixed immediately, the developers should run their project from the very beginning.

Ahmad Yussor Abu Ghazal - 201820056 Abdullah Majed Abu Ghali - 201410848

Supervisor Signature: Date:

Note: This is completed by the supervisor, and submitted to the Graduation Project Committee