**Development of a Web Application for finding and Applying for Jobs**

**Summer Training Report**

**CMSE 400**

**Talal Mahdy - 147139**

**Number of working days: 40**

**Period of Training: June 13, 2017 - August 1, 2017**

**Company Name: EBLA Computer Consultancy**

**Company Address: 14th floor, Behbehani Building,  
 Sharq, Kuwait**

**Computer Engineering Department**

**Eastern Mediterranean University**

# ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to EBLA Corporation, to my Supervisor Eng. Rami Jumah and to all of the amazing employees for giving me the opportunity to do this project of developing an extremely useful Web Application, which also helped me in doing a lot of research and I came to know about so many new interesting things.

# ABSTRACT

The principal objectives of the summer training internship were to create a well-designed, well-functioning Web Application using the .NET Framework, and at the same time, to learn about the new languages and technologies that are not taught in University programs. The purpose of the application is to assists its users in finding and applying for new Jobs. The project was started due to the shortage of similar Web Applications for finding and applying for Jobs, especially in Kuwait and the Gulf Region. The summer training began with the first two weeks being learning, practicing and experimenting weeks. After that, the focus was shifted towards designing the application and building an initial prototype. After that, the development of the application began and this stage took most of the time of the summer training. At the end of the summer training period, a final test of the application was performed and the application was ready for deployment. This summer training performed is going to help me a lot in choosing a correct career path in Software Engineering, help me in finding a Job quickly after graduation and it made me very interested in learning all the Web Development languages and technologies necessary to succeed in my future Job.

**Keywords:** Web Application, HTML, ASP.NET, C#, Job finder.TABLE OF CONTENTS Page

[ACKNOWLEDGEMENT 2](#_Toc495993447)

[ABSTRACT 3](#_Toc495993448)

[LIST OF FIGURES 5](#_Toc495993449)

[1. INTRODUCTION 1](#_Toc495993450)

[2. COMPANY OVERVIEW 2](#_Toc495993451)

[3. OBJECTIVES / PROBLEM DEFINITION 3](#_Toc495993452)

[4. WORK DONE AND METHODOLOGY APPLIED 4](#_Toc495993453)

[5. RESULTS AND DISCUSSION 12](#_Toc495993454)

[6. NEW SKILLS AND KNOWLEDGE GAINED THROUGH THE EXPERIENCE AND THE IMPORTANCE OF LIFELONG LEARNING 26](#_Toc495993455)

[7. CONCLUSIONS 28](#_Toc495993456)

[APPENDIX A 29](#_Toc495993457)

# LIST OF FIGURES

Figure 5.1. Solution Explorer.......................................................................................12

Figure 5.2. Home Page.................................................................................................13

Figure 5.3. Contact Us..................................................................................................14

Figure 5.4. Sign Up......................................................................................................15

Figure 5.5. Login..........................................................................................................16

Figure 5.6. User Profile................................................................................................17

Figure 5.7.Jobs.............................................................................................................18

Figure 5.8. Job Details..................................................................................................19

Figure 5.9. Resume Builder..........................................................................................20

Figure 5.10. Add New Job...........................................................................................21

Figure 5.11. View/Edit/Delete Jobs.............................................................................22

Figure 5.12. View/Delete/Download Resumes............................................................23

Figure 5.13. Jobs Table - Microsoft SQL Server.........................................................24

# 1. INTRODUCTION

On June 13, 2017, I started my first summer training at EBLA Computer Consultancy. After a meeting with my Supervisor Eng. Rami Jumah, I was given a project that was not crucial for the company, and that project was to develop a Job Portal Web Application using the .NET Framework. But before that, I was given some tasks to brush up my skills on various topics and actually learn the basics first. So the aim of the summer training was to learn some new technologies and develop an application at the same time. The purpose of developing a Job finding Web Application is due to the shortage of such types of applications especially in Kuwait and in the neighboring Gulf Countries. So an application like this could prove very useful to the people living in these countries as it will make it easier to find and apply for the Jobs they are interested in. In this report, I will start by giving an overview about the company, and then I am going to explain the objectives of the work given. I will then explain all about the project and the work done. After that, I will provide the results after the project was done using some figures and tables. And finally, I am going to write about the new knowledge gained from the summer training experience.

# 2. COMPANY OVERVIEW

EBLA Computer Consultancy Company was established in 1993 with offices in Kuwait, Qatar and the UAE (Abu Dhabi & Dubai). EBLA is one of the largest professionally managed IT companies in the Middle East and a leading systems integrator company in the Arab Gulf region today. EBLA’s key focus is to sell and support a full range of Enterprise software solutions from two of the leading software vendors in the world – IBM and Microsoft.

EBLA has over 70% share of the Enterprise Content Management (ECM) market in Kuwait and Qatar. EBLA also has Enterprise Agreements (EA) for Microsoft software solutions with the Governments of Kuwait and Qatar.

With over 300 professional and highly experienced employees, EBLA is financially strong and viable with a closed Shareholder status, and over the last 10 years EBLA has achieved consistent profitability and business growth of over 10% year on year.

Since its inception, EBLA continues to be managed by the same team of the initial owners and managers, thus retaining the Company’s internal intellectual property rights within the organization. Additionally, EBLA is an ISO 9001:2008 quality certified Company. This certification further strengthens its commitment to customer satisfaction and its dedication to continuous improvement of its business operations.

Address of the Company:

14th floor, Behbehani Building, Sharq, PO Box 1070, Dasman 15461, Kuwait

Contact Information:

Mohammed El-Sayed (Senior HR Officer) – Email: hr.corporate@eblacorp.com

# 3. OBJECTIVES / PROBLEM DEFINITION

The main aim of the summer training at EBLA was to enable me to learn some new technology that I did not learn at my University, and use that knowledge to develop a useful Web Application. The company is involved with many advanced projects with many government organizations and companies like the KNPC, the KOC, the Ministry of Information and Technology, etc. But the project given to me was not crucial for the company, which means that I did not have anything to worry about in case I did a mistake. The project given to me was to develop a Web Application to assist the people, especially in Kuwait and the neighboring Gulf Countries in finding and applying for Jobs. The application should be well designed using the HTML and CSS languages and I used the Adobe Dreamweaver tool to do the design activities. The development of the application must be made using the ASP.NET Framework along with the C# Programming language which is part of the .NET Platform and I used Microsoft Visual Studio as the main development tool. The database of the application could be constructed using any suitable technology such as MySQL, Oracle Database, or Microsoft SQL Server. I chose Microsoft SQL Server to manage the database. The application shall consist of a section for normal every day users and a separate section for the Administrator. The main features of the User Section of the application shall consist of an Account Creation, Login, Jobs Section, Profile, Resume Builder, Password Recovery, Contact Us, About Us, and Terms. The main features of the Admin Section shall include the addition, deletion, and editing of Jobs, and to view, delete and download of resumes.

# 4. WORK DONE AND METHODOLOGY APPLIED

During my first few days at the Company, I was in a bit of a state of shock since I was observing all those employees working on many advanced things that I did not know about. My supervisor and other employees comforted me and told me that everyone was once a beginner like me. I was given the task of developing a Job Portal Application, but before that, my supervisor asked me to learn all the basics before actually beginning in order for me to feel comfortable while working on the application. So during the first two weeks, my focus was mostly on learning C#, HTML, CSS, and brushing up on SQL.

C# was a somewhat easy language to learn since it felt similar to the C++ Programming Language which I was already familiar with. I started by brushing up on the Object Oriented Concepts, then I started learning the basic topics such as data types, variables, basic math operations, constants,overflowing,scope,type conversion, operators (arithmetic, logical, conditional), looping (for, while, do while, foreach),switch, break, continue, strings and string functions, string formatting, stringbuilder, working with dates, timespan, arrays and lists, enums, reference types and value types, exception handling, classes and objects, getters, setters, constructors, object initializers, methods, fields, access modifiers, properties, static, class methods, overloading methods, indexers, inheritance, override class methods, polymorphism/abstract class, operator overloading, generics, enums, structs, anonymous methods, lambda expressions, file I/O. I also revised many of the SQL topics that I already learnt from University.

After learning most of the important C# concepts, I started learning HTML and CSS and it was not very difficult too, it just needed some practice. I was learning and experimenting with many topics in HTML such as elements, comments, attributes, headings, paragraphs, styles, formatting, quotations, colors, styling with CSS, links, images, tables, lists, blocks, classes, exploration of different layouts, HTML forms such as text inputs, button, radio button, drop-down list, textarea, checkbox, exploration of different attributes, etc. And in CSS, I was experimenting with topics such as syntax, linking CSS to HTML, colors, backgrounds, borders, margins, padding, height/width,box model,outline,text,fonts,links,lists,tables,position,align, etc.

After around two weeks of learning the basics, I started the next stage which is to create a good design for the Web Application. I conducted a meeting with my supervisor and his assistant and we were discussing some design ideas. Since I was not going to use a Scripting language like JavaScript, the design was not going to look very advanced. Since designing on Visual Studio is not beginner friendly, the plan was to design most of the webpages for the application on Adobe Dreamweaver, and then converting those design documents to real ASP pages for use in Visual Studio. So I started out by writing the main functions and reasonable requirements of the user and admin panels of the website, researching navigation ideas (like a menu bar/side bar), color ideas (background, text color), picture ideas, researching design ideas from the Internet, looking through similar website designs, etc. I realized it was easier to use tables to organize the various sections of the page. I splitted the Home Page to four parts, the header section, topmenu section, body section (top picture, left sidebar area, and main body area on the right), and the footer section. The design of this page is important as it will be used as a template for the other pages as well. I then Implemented the main navigation bar with various options such as: Home, About, Contact, Jobs, Sign Up, Login. I also chose a professional looking picture and placed it at the top picture section of the body and did some editing on it. I also applied various CSS classes to various elements of the Page. After that, I started working on the Sign Up, Login and Contact Us pages where I added many relevant HTML Controls and applied various CSS Classes to these pages.

After finishing up most of the design documents, I imported all the HTML pages, the CSS file, and the images to the Visual Studio Environment. I started reading and learning more about ASP.NET and doing working on the application along the way. I learnt about many useful ASP topics such as viewstate, session state, application state, server control events, IsPostBack, Internet Information Services (IIS), TextBox Control, RadioButton, Checkbox, Hyperlink, Button, LinkButton, Dropdown list, CheckBoxList, FileUpload Control, Calendar Control, Server MapPath Method, Validation, RequiredField Validator, CompareValidator, CustomValidator, ValidationSummary, ResponseRedirect, QueryString, Sessions, Exception Handling, Error Events, Custom Errors, Sending Emails, Master Pages, ContentPlaceHolder, Menu Control, Sitemap.

To create a Master Page, I used the Home Page as a reference. I removed all the unnecessary elements from it, added the necessary code to make it function as an ASP Page, and placed a ContentPlaceHolder in a strategic position on the Body Section for other pages to work with. The Master Page has many advantages since it reduces the amount of code repetition for each Page. I then chose the Home.aspx as my default page. I then started creating similar pages like this which use the Master Page as a reference. In the Master Page, I had to change some of the coding from HTML to ASP since HTML pages will not work anymore.

After working on the Home Page, I started working on the Sign Up Page. Even though the design was ready, it looked different on Visual Studio and I had to fix many small errors. I had to also convert all the HTML Controls I created to ASP Controls. HTML Controls do not run on the Server, and they do not have a PostBack and a ViewState feature. So using HTML Controls will not be ideal since we have to keep writing separate code to make it PostBack to the Server. An ASP Page consists of two main parts, a .aspx file for design and .aspx.cs file for writing the code behind of the page. This code can be written in either C# or VB.NET (.aspx.vb). I wrote the code behind using C#. There is also a .aspx.designer.cs file, but this file is mostly generated by Visual Studio. I had to learn to connect my application to the Microsoft SQL Server Database. I created a Database using Microsoft SQL Server Management Studio and then I created a new table for Users. I edited the web.config file and wrote a proper connection string which is very useful to avoid repeating the same code in all the pages that I am going to work with. Then I had to write the code behind for the Sign Up page. I also applied many validation controls to force the user to fill out all the required fields and agree to the terms and conditions. I also applied an ErrorLabel that summarizes the errors for the user so that he would be able to quickly fix them. After the user clicks on the Sign Up button, the user data entered in the form should be submitted to the database in a proper format.

After finishing the Sign Up page, I started working on the Login page. Again, I had to convert the HTML Controls to ASP Controls. I also had to write the code behind, create user sessions for a logged in user, and making sure login works. Then I had to think about what happens after login is successful. I created a page called UserProfile which is also linked to the Master Page. On that page, I created a Label with a helpful Message informing the user that he successfully logged in. After login is successful, the users are directed to their respective UserProfile page. I had to also create validations to make sure the user actually enters a username and a password and whether those were correct. If they were not correct, a message is displayed informing the user that the username/password are incorrect. I also faced some problems like whenever I am on the Login or Sign Up page and I tried to visit the other page by clicking on the page link from the menu bar above, the validations on the page would pop up and will not allow me to visit the other page. This is because whenever I click on the Login/SignUp LinkButton Control, the control causes the page to postback to the Server. So I had to make some validation groups to distinguish between the unrelated validation fields. I also faced another problem where a user could enter http://localhost:53080/UserProfile.aspx directly into the address bar and the UserProfile page would open. Since the page is not deployed to an actual server, Visual Studio uses a local host as the Server. To fix this, in the UserProfile code behind file, I should check if the "user" session has a value. If it does not have a value, it means that the user is not logged in and he has to be redirected to the login page with a message requesting him to do so. There was another small problem that if the user successfully logged in, the links in the navigation bar above will not change to reflect the successful log in. The Login link should change to display a Logout link and the Sign Up link should change to display a Profile link. To fix this, I had to replace the menu bar links in the Master Page with ASP LinkButton Controls to distinguish between Login/Logout and Signup/Profile.

After finishing the Login page, I had to work on the UserProfile page. On this page, I had to retrieve the user data which is stored in the database according to which user is logged in. I created an ASP DetailsView Data Control and I had to take many things into account when working with this control. There are many ASP Bound Fields that retrieve data from the database in this control. Since I already defined a successful login as a Session State called “user”, I can use this as a basis for which user’s data I should obtain. I also added a section where I displayed an edited Photo on the right of the Page for better design. I am planning to update this section with a proper section for uploading a Profile Picture later.

After working on the UserProfile page, it was time to start working on the Job sections of the Application. First of all, I created a new table in the database for all the Jobs. Then I created a Jobs page which is linked to the Master Page. On the Jobs page, I wanted to display all the Jobs that are stored in the database. Therefore, I created an ASP GridView Data Control and edited many of its properties. There were many columns that do not fit on this page so I created another Page called JobDetails. I wanted to enable the users to view more details about each posted Job by clicking on a respective link next to each posted Job. This was possible since I made a unique column in my database for the Jobs, and using the DataBinder.Eval() method, I was able to create unique Details links for each posted Job. When a user clicks on any Details link, he will be redirected to the JobDetails page, where a DetailsView ASP Controls shows the user all the Details about the respective Job he wants to find more details about. After the JobDetails problem was over, it was time to actually enable the users to apply for a Job. I created a new table for all Resumes in the database. After that, I wanted to use the same technique that I used to display all the unique hyperlinks about the details of each posted Job. I wanted to display similar links to apply for that Job. I created a new Page called ResumeBuilder. Whenever a user clicks on the Apply hyperlink for a Job, the user would be redirected to that Page. The ResumeBuilder page was challenging to develop as it had many factors to take into account. I had to retrieve the user data from the database in an easy to edit manner. I faced a problem where the date of birth was not being retrieved and I fixed that. Then I had to add more optional fields for the user to enter. I had to also work on the FileUpload ASP Control to enable the users to upload their resumes. Since I am working on a locally hosted server, I had to create a Folder called Resumes in the project solution to save all the resume files in it. I used the Server.MapPath() method to save the Resume files. Then I created validations to force the user to type in all required fields. There were some issues with the ResumeBuilder page. A user can type in the address bar http://localhost:53080/ResumeBuilder.aspx without logging in and the page would open, this problem was fixed in a similar way as the UserProfile page was fixed, by checking whether the “user” session has a value and that the user is actually logged in. Another problem was that there was a possibility that two or more users uploaded a Resume file with the exact same name. To eliminate this possibility, I chose to save the Resume file name with a prefix of the Username of the user. Another problem was that a user could upload a Resume file which is not in a proper format like a .doc or a .pdf format and he could take advantage of this vulnerability and upload a malicious file like a Malware or a Virus. To fix this, I added some validations to allow the users to only upload files of certain types using the Regex and Match classes. Now, the ResumeBuilder page is ready and I wanted to enable the users to apply for a Job from the JobDetails page too. There were two ways where I could do this, I could use the same unique hyperlink method that I used in the Jobs page, or I can create an Apply Now Button that required a code behind which redirects us to the ResumeBuilder page along with the JobID. I chose Apply Button method to learn more about it.

After completing the most important section which was the Jobs Section, I started working on the Contact Us page. On this page, I wanted to enable the users to communicate with the administrator about any inquiry they may have. There were two ways to solve this problem, the first way was to create another table for Contacting in the database and save the Message and all the user entered information from the Contact page in the table and then create a page in the Admin Section for the Administrator to view all the Messages. The second method was to actually send the Message and the entire user entered information directly to the Email Address of the administrator. I already implemented something similar using the first method, so I chose the second method to learn more about this concept. Sending Emails was possible by using the Simple Mail Transfer Protocol (SMTP). Since I require a host to send Emails, and this project is not a big project, I opted to use my own personal Email as the administrator Email and the host of my personal Email is   
smtp-mail.outlook.com. Also, many of the top Email host provider companies require that SSL (Secure Sockets Layer) be enabled in order to successfully send Emails. This is necessary to have an encrypted and a secure connection between the Web Server and the Client. Also, sometimes an error may occur in sending the Email such as if the user is disconnected from the Internet, or the host is temporarily unavailable, or some other issue. If an issue like this occurs, it can cause a runtime error and the application may crash. To prevent errors like these, I implemented some exception handling by using the try and catch blocks.

The Users Section of the Web Application is almost ready, and it was time to start developing the Administrator Section of the Application. I started by going back to Adobe Dreamweaver and doing some edits on the User Home HTML Page which I am going to use a reference to create main Administrator Master Page. After doing many edits and removing many unnecessary things for an administrator, the Master Page and the Home Page of the Admin Panel were ready. The Admin Panel will consist of pages to Add a New Job, View Jobs, Edit Jobs, Delete Jobs, View Resumes, Delete Resumes, Download Resumes. I created a New Folder in the Project Solution which is dedicated to storing all the Administrator related files.

First of all, I started working on the View Jobs page in the Admin Panel which is linked to the AdminMasterPage. I created an ASP GridView Data Control, edited many of its properties and created many ASP BoundFields to retrieve the data from the Database using my Connection String. After all the Jobs were successfully retrieved from the database, I wanted to add unique Edit Job hyperlinks to all the rows where there are Jobs. I used the same DataBinder.Eval() Method used before to do that. I then created a separate page called EditJobs which is linked to the AdminMasterPage. In the EditJobs page, I wanted to create all the Controls necessary and then retrieve the data into those controls in a well formatted manner. I also added validations and an Error Label at the top of the page to make sure that the admin enters all the information in the required fields. In the code behind, I wrote an update query that updates the data based on what the administrator entered but there was a small problem that when the Update Button is clicked, the same page is posted back causing the old values to be taken again from the server. Therefore, in the Page\_Load method, I wrote a statement if(!IsPostBack) to check if the page was not posted back and in order not to run the code in the Page\_Load again. Furthermore, I added a delete link to all the rows where the Jobs are shown using the same DataBinder.Eval() Method and then I wrote a code to display a confirmation message before a Job is actually deleted in case the delete button was pressed by mistake.

The next page to develop was the Add New Jobs page. This page was somewhat easily to develop as it is similar to a page that I have developed before which is the Sign Up page for Users. I created all the ASP Controls necessary to add a new Job along with a Submit Button Control. I also added all the necessary validations and an Error Label to summarize the errors for the administrator. I also added some useful labels to inform the Administrator that a Job was successfully added or deleted or updated. In the code behind, I used the try and catch exception handling for more reliability in case a wrong format was entered.

The final page to develop in the Admin Panel was the ViewResumes page. In this page, I used an ASP GridView Data Control with many BoundFields to display all the data from the Resumes table of the Database. I then created a delete link for each Resume using the same technique of DataBinder.Eval() and then I wrote the code behind for the page. I then added download links for all available Resumes of job applications for the administrator to download using the same DataBinder.Eval() method. I also added a useful label to inform the Administrator that a Resume was successfully deleted and I wrote a code to display a confirmation message before a Job is actually deleted in case the delete button was pressed by mistake.

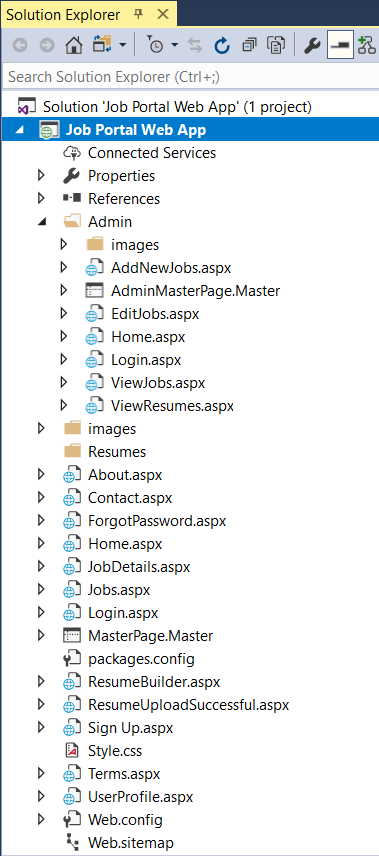
After the development of the Admin Panel is over, I had to work on some other small features. The first one is a Password Recovery feature. This feature was similar to the Contact Us page where the SMTP was used. The users could enter their Username and their Password would be sent to the respective email they used when Signing Up. Then I had to apply some security measures to prevent normal users from accessing the Admin Panel. A user could manually enter localhost:53080/admin/Home.aspx for example and he would have access to the Admin Panel. I removed this ability by creating a Login page which is linked to the AdminMasterPage. Whenever a user tries to access any Admin page, and if there is no Session or no login was recorded, then the user will be automatically redirected to this login page. I added code to prevent users from accessing any other Admin Page without passing the Login page first. The last I created was a useful SiteMap feature for the Users using the ASP SiteMap control and by creating a file with a .sitemap that contains hierarchal links to all the pages in the User Panel. This will help the Users in navigation and it will enable them to know at which point they are on the Website and how they arrived at this point. Finally, I tested the Application as a whole and discussed it with my Supervisor where I prepared some known issues list and an update list.

# 5. RESULTS AND DISCUSSION

In this section of the report, I will provide many screenshots of the Jobs Web Application and I will discuss these screenshots.

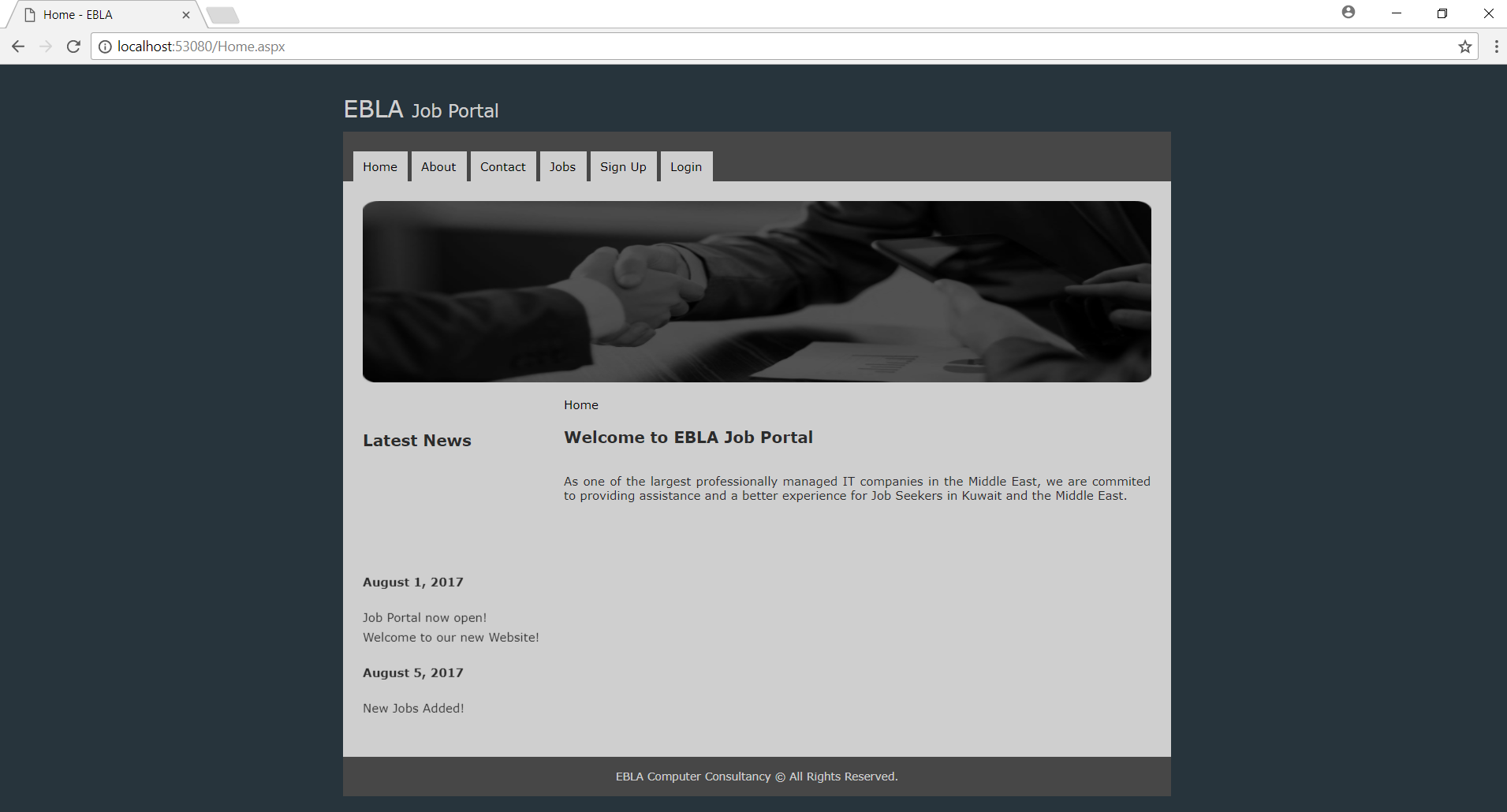
**5.1. Solution Explorer:**

A solution is a group of projects. I have only one project in my solution which is the Job Portal Web App. These are all the files included in the project.



**Figure 5.1: Solution Explorer**

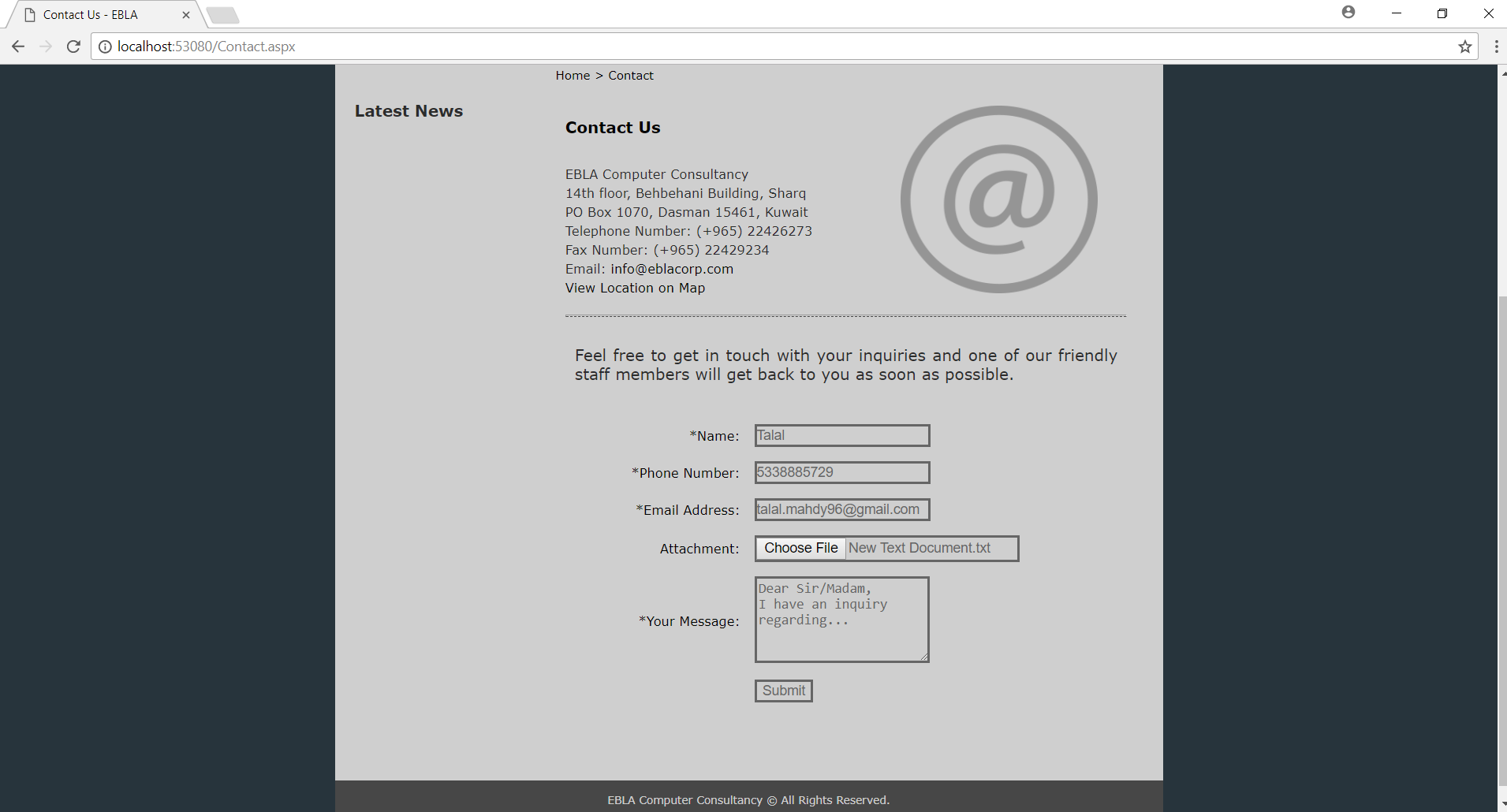
**5.2. Home Page:**

This is the Startup Page of the Project. On the left, there is a Latest News Section with a Marquee Control which scrolls from bottom to top. This page consists of 4 main sections, the header section, topmenu section, body section (top picture, left sidebar area, and main body area on the right), and the footer section.

## 

**Figure 5.2: Home Page**

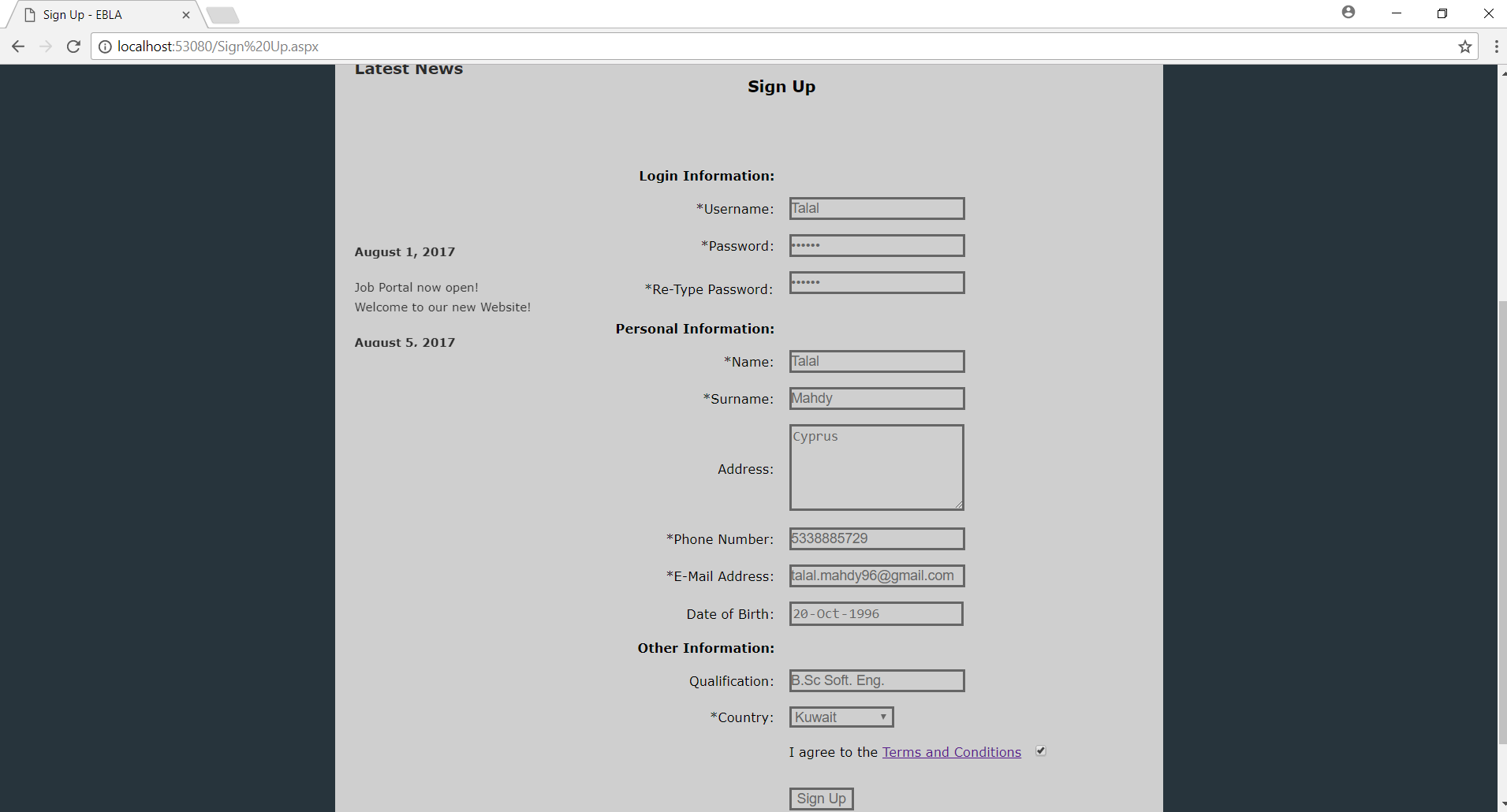
**5.3. Contact Us:**

This is the Contact Us page of the Application. When a user fills out this form, an email will be sent to the Administrator with all these details. The fields marked with an ‘\*’ are required fields. A user cannot submit the form without filling out those fields first.

## 

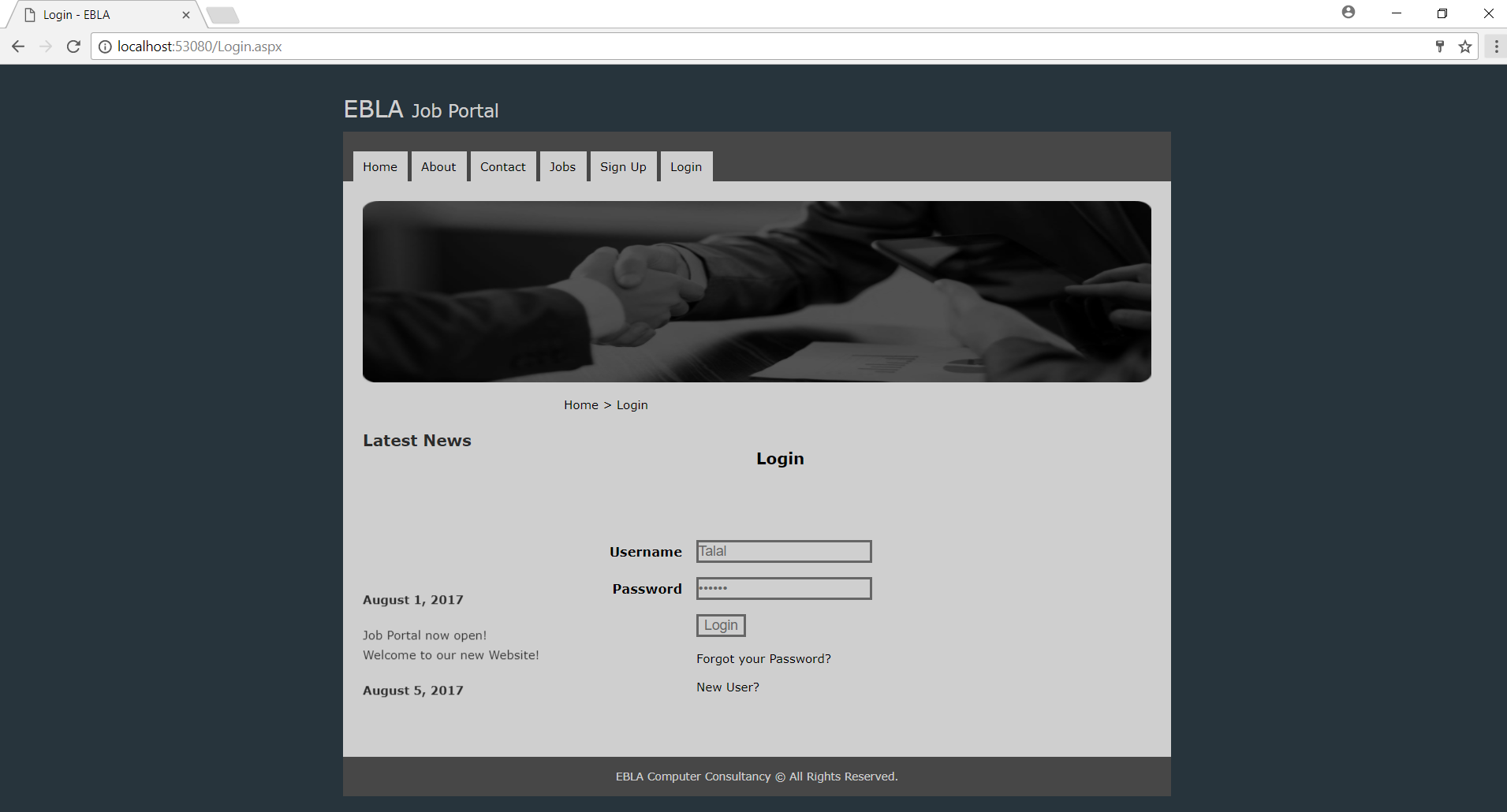
**Figure 5.3: Contact Us**

**5.4. Sign Up:**

This is the Sign Up page of the application. A user should create an account before applying for a Job. A user should fill out all the fields marked with an ‘\*’ and accept the terms and conditions before creating an account.

**Figure 5.4: Sign Up**

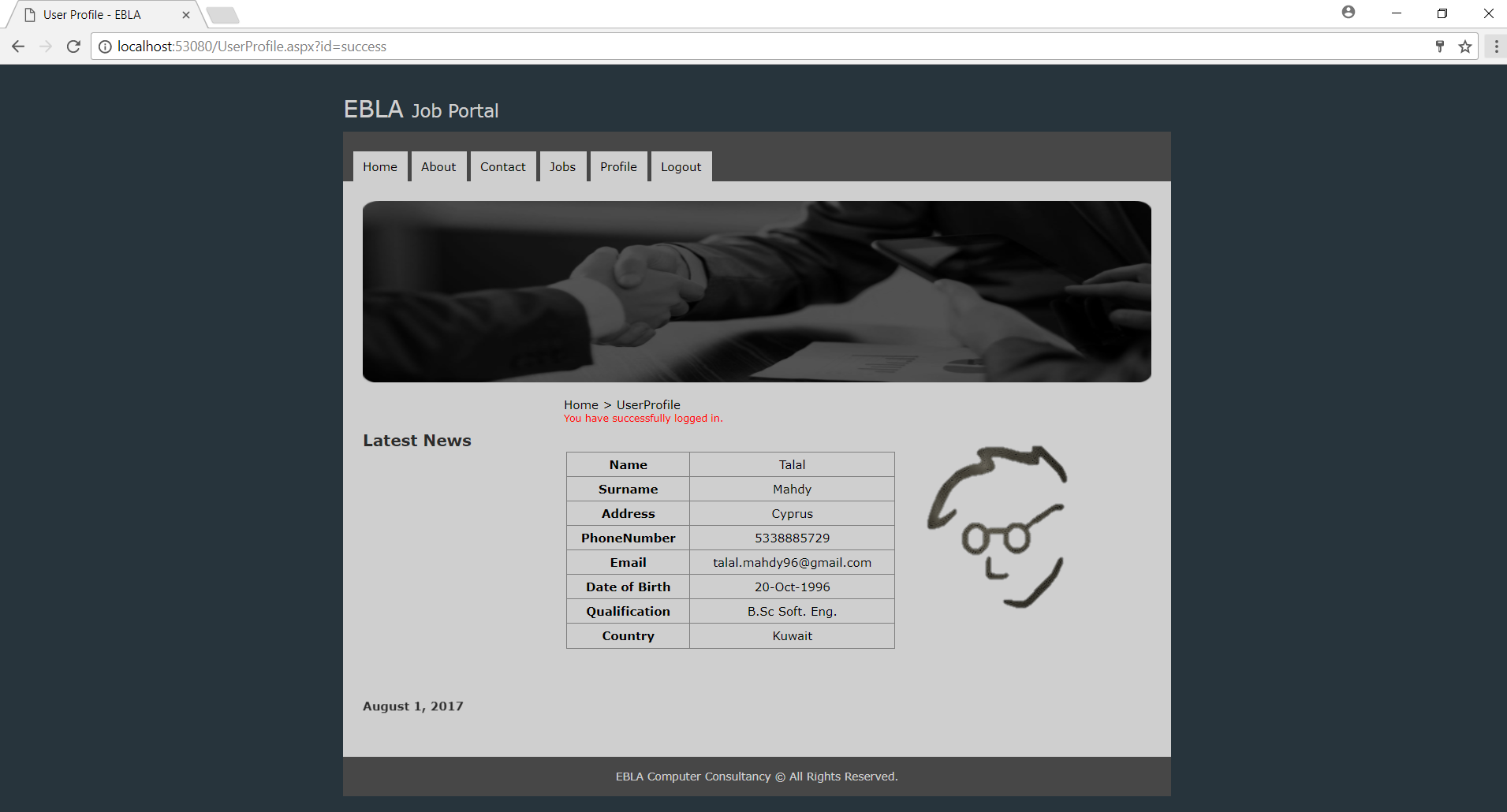
**5.5. Login:**

This is the Login Page. After creating an account, users can login, view their profile, build their resumes, and apply for a Job.

## 

**Figure 5.5: Login**

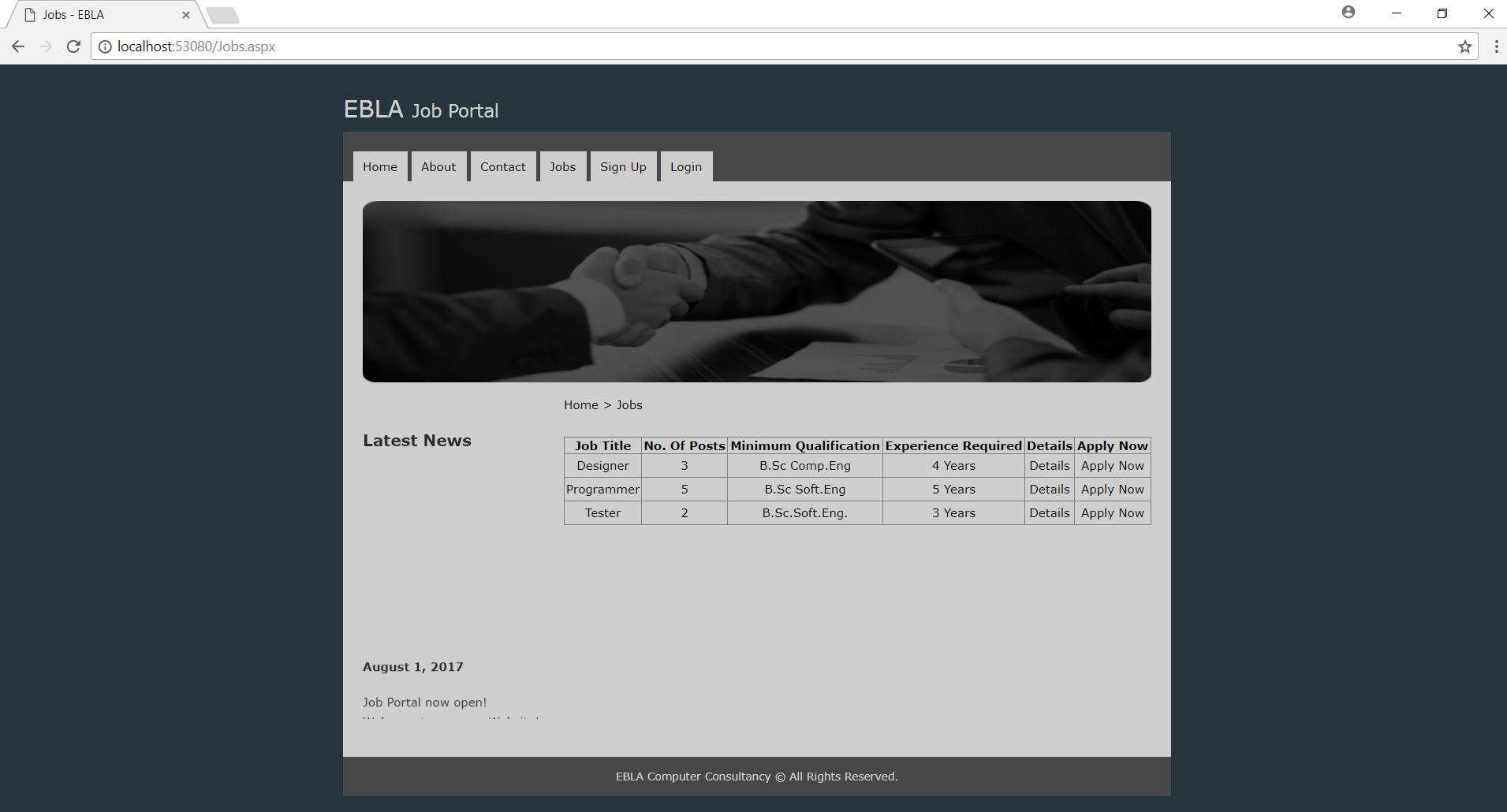
**5.6. User Profile:**

This is the User Profile page. After a successful login, this page will be displayed by default.

## 

**Figure 5.6: User Profile**

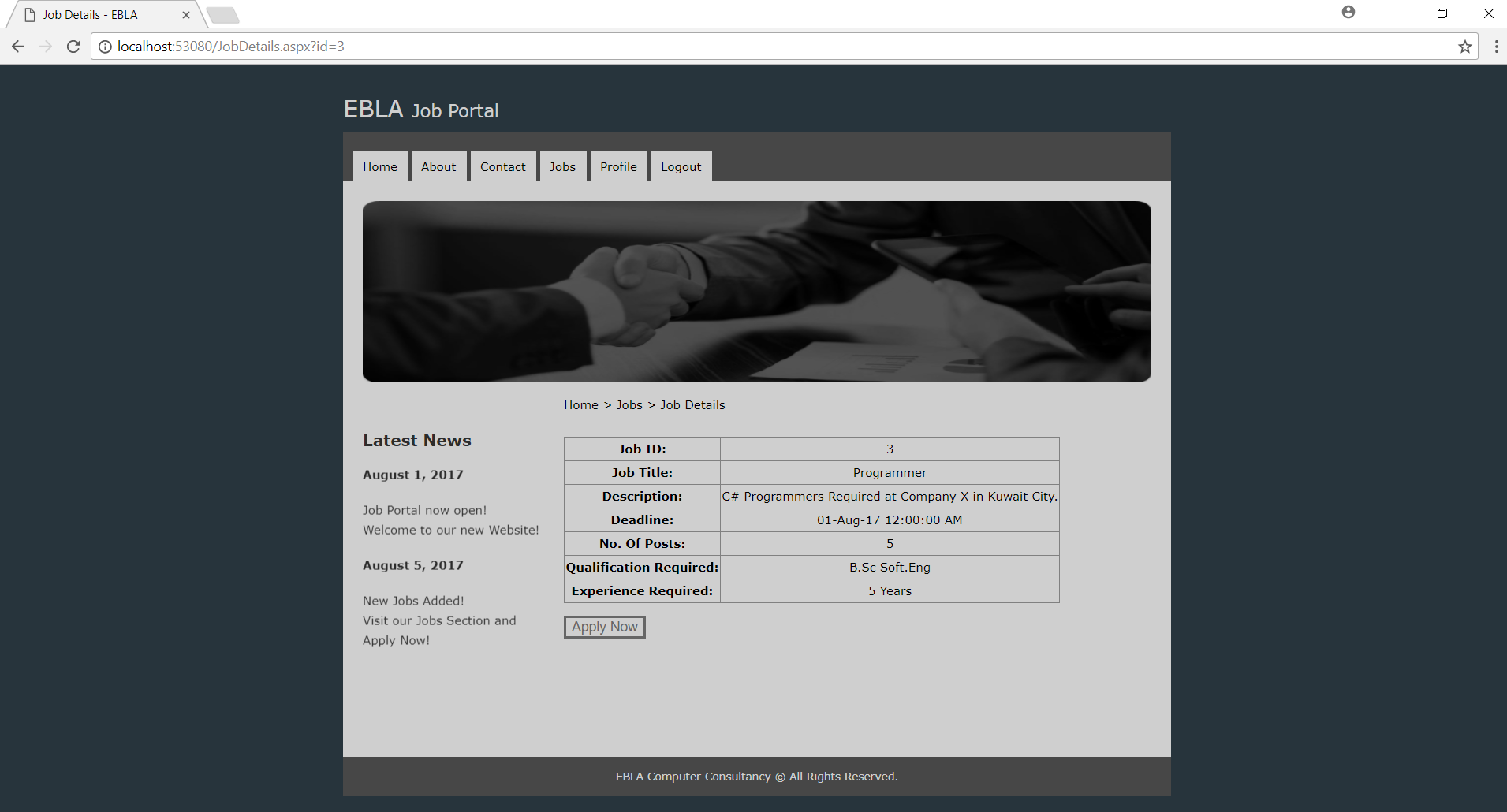
**5.7. Jobs:**

This is the main Jobs page. A user may apply to any posted Job. If users were not satisfied with the information provided on a Job posted on this page, they may visit the respective Details page of a posted Job and apply from there.

## 

**Figure 5.7: Jobs**

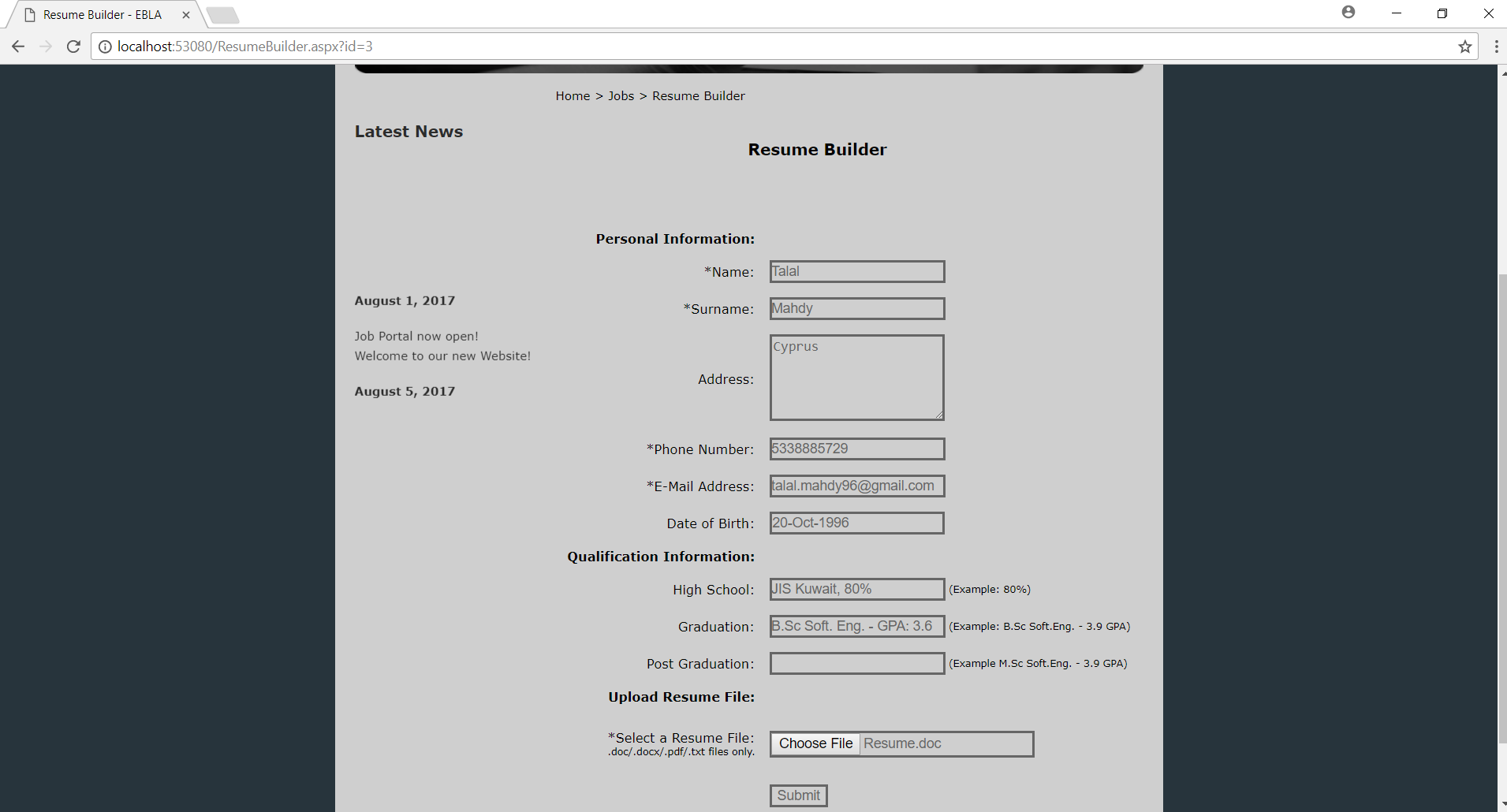
**5.8. Job Details:**

This is the Job Details page for one of the posted Jobs. A user can also apply to the Job through this page.

## 

**Figure 5.8: Job Details**

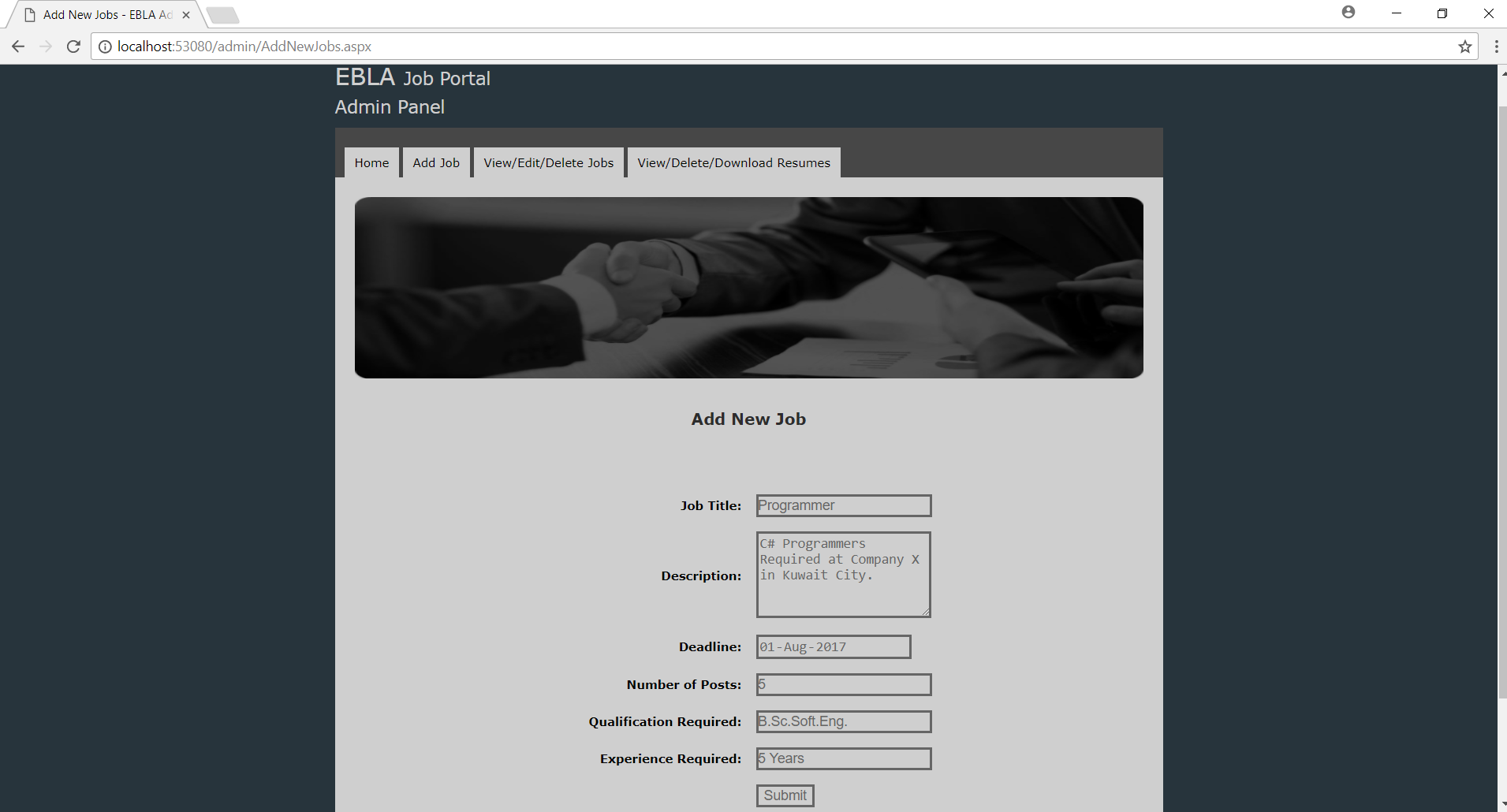
**5.9. Resume Builder:**

This is the Resume Builder page. After users choose a Job to apply to, they will be forwared to this page. Some information they wrote out before will be already written for them and they may choose to provide some more information. After submitting the form, users will be forwared to a page confirming successful application for a Job.

## 

**Figure 5.9: Resume Builder**

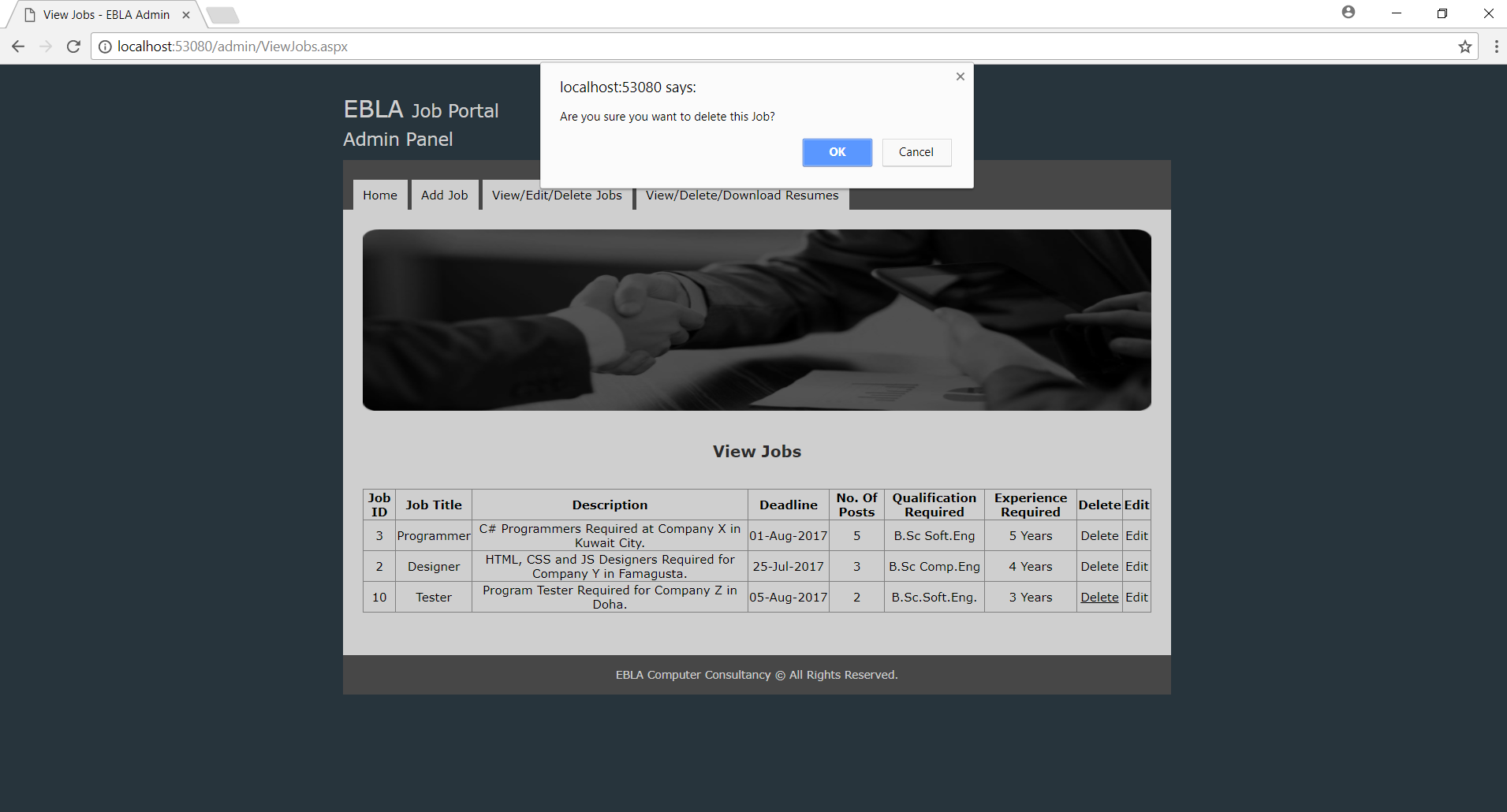
**5.10. Add New Job – Admin Panel:**

This is the Admin Panel section. It can only be accessed by the administrator by typing in host/admin/”any admin page.aspx” in the address bar. Before accessing any admin page, an admin login should occur from the admin login page. An admin may post new Jobs using this page.

## 

**Figure 5.10: Add New Job**

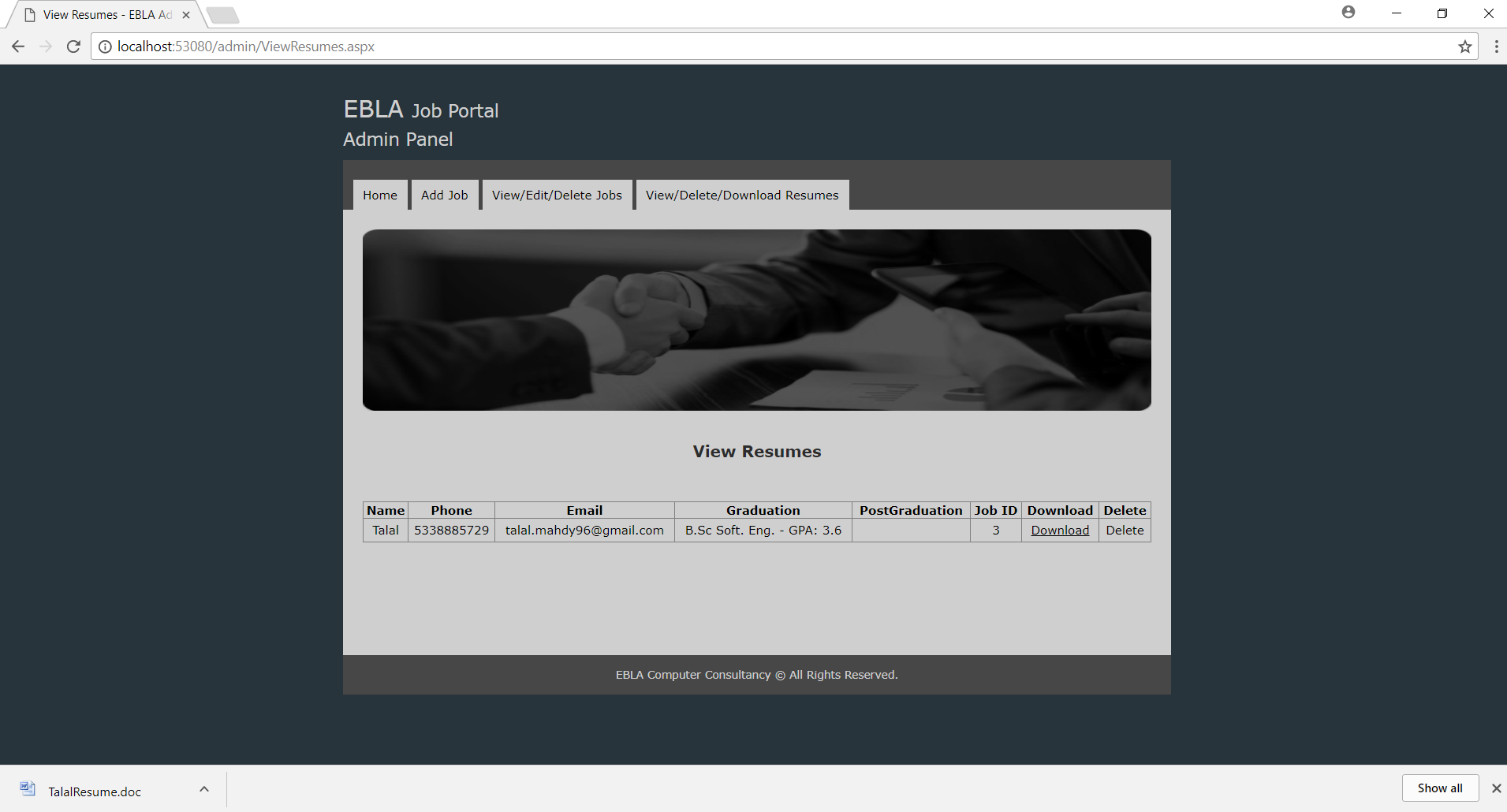
**5.11. View/Edit/Delete Jobs – Admin Panel:**

An admin may use this page to delete a posted Job. If an admin clicks chooses to edit a posted Job, they will be forwared to a separate page called EditJobs.aspx. On that page, the Job data is automatically retrieved in Text Boxes and the Admin may easily edit that data.

## 

**Figure 5.11: View/Edit/Delete Jobs**

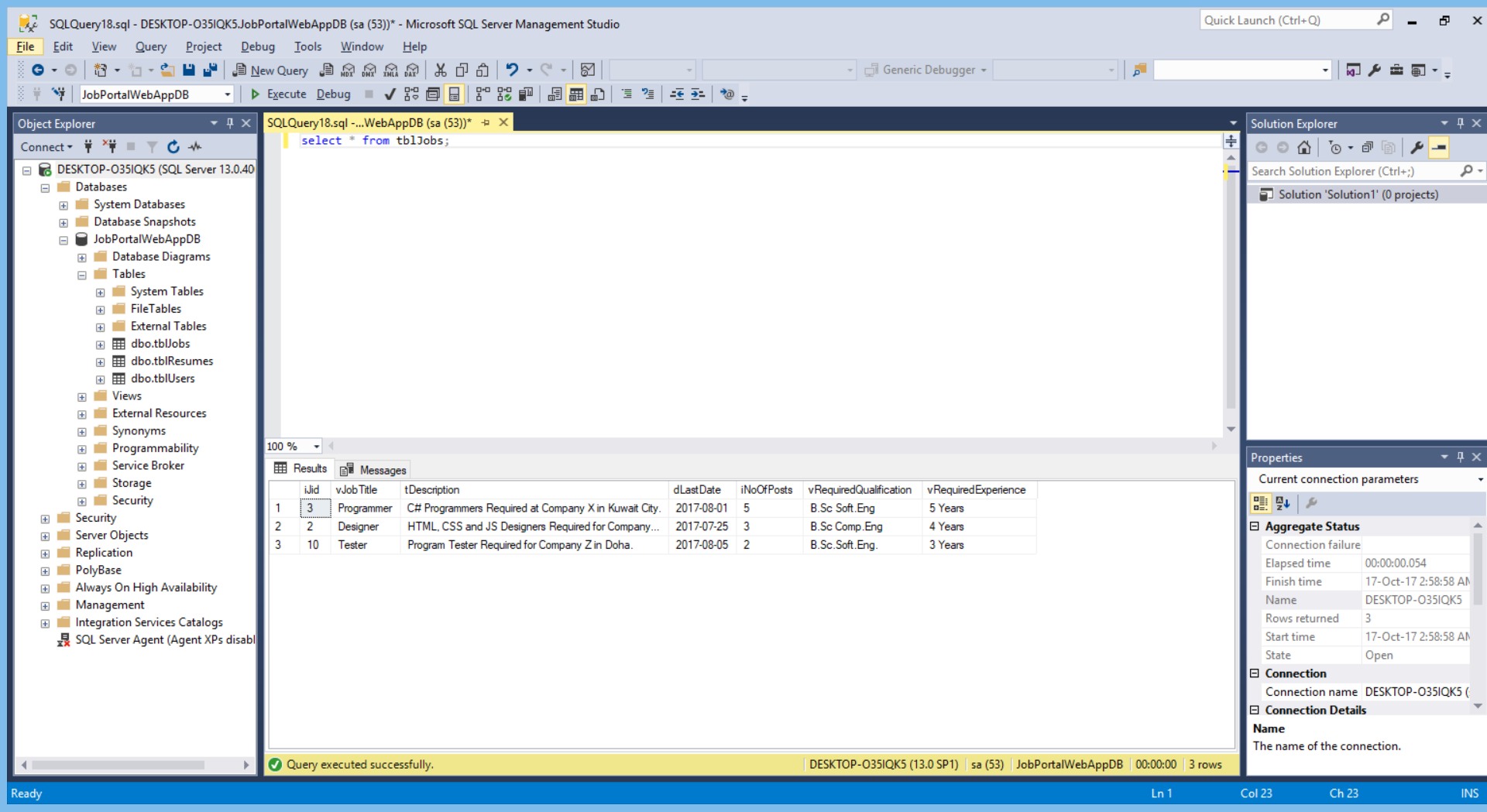
**5.12. View/Delete/Download Resumes:**

An admin may download the Resumes of the people who applied for a Job from this page. An admin may also delete unwanted resumes from this page.

## 

**Figure 5.12: View/Delete/Download Resumes**

**5.13. Jobs Table – Microsoft SQL Server:**

There are three tables in the Database of the application. A table containing the User data, a table containing the Jobs data, and a table containing the Resume data.

## 

**Figure 5.13: Jobs Table – Microsoft SQL Server**

**5.14. Suggestions and Improvements:**

The Web Application that I created is not perfect. I can do many things to improve it. Some known issues and areas of improvement are mentioned below:

A user can apply to a Job after the deadline has passed, and before an admin removes it from the list of Jobs.   
A user can apply to a Job more than once.  
Two or more users can register with the same username.  
After users log in, they are able to visit the login page again by typing it in the address bar, and without logging out first.  
There are no restrictions on the password, a user could provide an insecure password for his account.  
A ‘remember me’ feature is absent from the login page.  
A profile picture feature is absent.   
The design of the Web Application is not advanced.

6. NEW SKILLS AND KNOWLEDGE GAINED THROUGH THE EXPERIENCE AND THE IMPORTANCE OF LIFELONG LEARNING

I learnt a lot of things during my Summer Training Experience. One of the most important things that I learnt is that technology changes and gets updated very rapidly. I also noticed that many of the topics that we learn in our University programs are not direct topics that we are going to apply directly in our Jobs. But instead, they are very important foundations to all the future topics that we are going to learn. In my opinion, engaging in lifelong learning is something that every Human Being should do if they want to keep up with today’s competitive Job Market. Students who complete their Bachelor’s Degree and whom do not learn any language, new technology, new skills, etc will very soon find themselves very inferior to other employees since their knowledge and skills will be out of date. The concept of lifelong learning does not only apply to the field of Computer Engineering, but to all other fields of study.

During my High School Education, I was very grateful that I was able to take Computer Science classes during the last two years. I learnt about many interesting topics in High School including the C++ Programming Language, Data Structures, SQL, some Computer Networks concepts, etc. So when I arrived at my University, I was ready for many courses since I already had some basics from High School. During my studies at the University, I have been actively working to further improve my C++ skills. I learnt many advanced C++ topics and practiced them by solving problems. I also joined the ACM Club at the Computer Engineering Department where I was meeting with my colleagues and solving some problems. I was also fascinated about the new field of Mobile Application Development and I wanted to learn more about this exciting new field that I chose to develop a Mobile App in one of my Software Engineering Course projects with the help of my Colleagues.

There are many career fields in Computer Engineering. Some of them are Mobile Application Development, Web Application Development, Network Administration, Systems Analysis, Computer Design, Robotics, Quality Assurance and Testing, etc.   
In my opinion, a Computer/Software Engineering graduate should focus mostly on learning just a few topics that are related to his field of interest, but he should always strive to perfection in those topics. For example, after doing my summer training, I became very interested in the field of Web Application Development. Therefore, after I complete my Bachelor’s Degree, I will not be wasting my time in learning the languages or frameworks that are not relevant to achieving my goal of becoming a Full Stack Web Developer. Instead, I will be working to improve my skills in PHP or ASP.NET along with the C# Programming Language. I will also try to master many other crucial languages in Web Development such as JavaScript, HTML, CSS, etc. I will also try to master working with Database Management Systems by learning the Structured Query Language in depth.

# 7. CONCLUSIONS

In Conclusion, I was really fascinated by the Summer Training that I have conducted with EBLA Corp. I am really grateful that Eastern Mediterranean University includes this training in our Program of Study. At the workplace, I designed and developed a Web Application that helps the people in finding and applying for Jobs. Development of this application was not possible without the guidance and mentorship of my Supervisor Eng. Rami Jumah. In my opinion, I think that the summer training was one of the most useful things that I have ever conducted and it is going to be very useful in doing my Graduation Project and other projects at my University. Not only did I develop an application, I also built a group of networks by meeting and contacting many amazing employees, I learnt about the Job Market and how work is conducted in Companies, I learnt about many new technologies and languages that I did not know about in University, and the most important of all, I learnt to Never stop learning, because life never stops teaching.

# 

# APPENDIX A

**Instructions:**

After I finished working on the application, I did not really work on the actual deployment of the application since the application was intended for educational purposes and was not a commercial product. Since the Web Application is locally hosted, it will be a bit difficult to copy it from one PC to another. Moreover, the database is also locally hosted and that means that Microsoft SQL Server Management Studio should be installed on the PC and it should contain the database tables. I included the complete project folder along with the backup of the Database in the CD included with the report.