EASY PC BUY

A website to compare computer parts price

INSTITUTE OF INFORMATION TECHNOLOGY
University of Dhaka

EASY PC BUY

Technical Report And User Manual

Supervised by:

Md. Nurul Ahad Tawhid
Assistant Professor
Institute of Information Technology,
University of Dhaka.

Submitted by:

Taslima Akbar Keya (BSSE 0901)
Saif Kamal Chowdhury (BSSE 0924)
BSSE Session : 2016-2017

Date: 29th May, 2019

IIT
University of Dhaka

Contents

LETTER OF TRANSMITTAL

29th May, 2019

Dr. Mohammed Shafiul Alam Khan

Associate Professor
Institute of Information Technology ,
University of Dhaka.

Subject: Submission of final report on Software Project Lab 2.

Dear Sir,

With due respect, we are pleased to submit the final report on software project named Easy PC Buy. Although this report may have shortcomings we did try our level best to produce an acceptable software. We would be highly obliged if you overlooked our mistakes and accepted our effort we put in this project.

Sincerely yours,
Taslima Akbar Keya (BSSE 0901)
Saif Kamal Chowdhury (BSSE 0924)
BSSE 9th batch
Institute of Information Technology
University of Dhaka

ACKNOWLEDGEMENT

At first, we would like to thank almighty for helping us complete this project.

We would like to express our deepest gratitude to all those who provided us the support and encouragement to complete this project. Thanks to our Supervisor teacher Md.Nurul Ahad Tawhid, assistant Professor, Institute of Information Technology, University of Dhaka, whose continuous suggestions and guidance has been invaluable to us. Also special thanks to Dr. Mohammed Shafiul Alam Khan, Dr. Kazi Muheymin-Us-Sakib sir,Nadia Nahar mam,and Saeed Siddik sir for their valuable opinions which helped us to improve our insight. Without such stimulating suggestions, the project could not have made this much progress.

We are grateful to the Institute of Information Technology for giving us the opportunity to do such a project.

Lastly, we would like to thank our classmates. Although they were not on our team, they have always been helpful and provided valuable insights from time to time.

ABSTRACT

This report is about the final report of project “Easy Pc Buy”. It contains background, scope and objective of the project. It also has problem description, technology we have used, source code description and user manual. This report has followed SRS report that we have been following. Hope it will help to understand the workflow of the website.

CHAPTER 1: Project Definition

This is a comparison website which deals with computer component parts from two specific websites. The project is entitled to the name “Easy PC Buy”. People can view this website to compare prices of different components and get the best prices. People also have the option to create accounts. The website is user friendly and it will make easier for people to select the best components for their personal computer according to their desired budget.

1.1 Background

People in our country who are in the market to buy new computer usually do not have sufficient knowledge about the pricing and often get deceived by the sellers. Different sellers/shops sell the same component at different prices which is not convenient. From this website consisting prices from multiple shop, one can find the best price available for the product.

With this web application, members can easily enter to their account anywhere..They can easily find any specific components as there are lot of categories to help to search the song

1.2 About the Project

The knowledge of SRS, OOP, web technology and SQL has been implemented in this project. General users can search components and also can view price and description of the component. A registration system has been provided and the members can login into their account from anywhere to get their wishlist through this application. Members can also rate the site and can write review.

CHAPTER 2: Implementation Overview

This chapter aims to describe the implementation process of "Easy PC Buy". Here the technologies that have been used to develop this system and the testing that have been done during this system development will be described in brief. Implementation is the final and important phase. Implementation is the stage in the project where the theoretical design is turned into a working system.

2.1 Problem Description

There are two tasks here. First we have to parse the two websites to get the values of computer components to populate our database. Then we load up the data in the website for the users to view . There is a common homepage for all the users. There they can search for components and can authenticate for a personal account. A non-member can register to be a member. There is a different way to search a component. Member can rate on the product and can write review.

2.2 Technology and Tools Used in implementation

Development technologies are growing very rapidly with the increase of requirements. The technologies that have been used to develop this system is the most recent technologies and also very much appropriate to it.

- **Hyper Text Markup Language (HTML)**
  Hyper Text Markup Language (HTML) is the main markup language for web pages. HTML elements are the basic building-blocks of a webpage. 3 The latest HTML5 has been used for developing this system
- **Cascading Style Sheets (CSS)**
  Cascading Style Sheets (CSS) is a stylesheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML. CSS3 have been used for developing this system. CSS3 is the latest standard for CSS. It is completely backwards-compatible with earlier versions of CSS.
- **JavaScript(JS)**
  JavaScript is a high-level, dynamic and interpreted run-time language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production. The majority of websites employ it and all modern Web browsers support it without the need for plug-ins. It has an API for working with text, arrays, dates and

regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

- **Python 3.7**
  Python is an interpreted, high-level, general-purpose programming language.Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

- **ASP.NET Core 2.0**
  ASP.NET Core is a new version of ASP.NET by Microsoft. It is an open-source web framework which can be run on Windows, Mac, or Linux.

- **Visual Studio 2017**
  Visual Studio is a Microsoft-integrated development environment (IDE) that can be used for developing consoles, graphical user interfaces (GUIs), Windows Forms, Web services and Web applications.
  Visual Studio is used to write native code and managed code supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

- **Sublime Text**
  Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and its functionality can be extended by users with plugins, typically community-built and maintained under free-software licenses.

CHAPTER 3: Source Code Description

This chapter has been categorized into two parts.

3.1 Front-end

Since the main web application was designed in ASP.Net Core 2.0 the front-end was mostly Razor Views generated from the Controller class which was built upon the Models generated from the Database Tables. These views are mostly comprised of HTML tags and attributes. Following is an example of the front-end code snippet from the project.

```
class="container">
@foreach (var item in Model)
{
    <div class="flip-card">
        <div class="flip-card-inner">
            <div class="flip-card-front">

                <img src="@item.PicLink" alt="Avatar" style="width:200px;height:200px;">

            </div>
            <div class="flip-card-back">
                <h5>@item.Name</h5>
                <p>Startech Price: ৳ @item.StarPrice</p>
                <p>Ryans Price: ৳ @item.RyansPrice</p>

                <button class="btn-default"><a asp-action="Details" asp-route-id="@item.MoboId">Details</a>
            </div>
        </div>
    </div>
}
.v>
```

Figure 2.1: Font-end code 1

The views can be stylized with CSS. ASP.NET Core 2.0 by default comes with bootstrap 3. So very little customized CSS was needed for the project. An open source bootstrap theme was added to the project for different look and feel. This theme is called “bootstrap-hero.css”. However some CSS still had to be done. Following is a snippet of a CSS code.

```css
.flip-card-front, .flip-card-back {
    position: absolute;
    width: 100%;
    height: 100%;
    backface-visibility: hidden;

}

.flip-card-front {
    background-color: #FFFFFF;
    color: black;
    border-radius: 10px;
}

.flip-card-back {
    background-color: dodgerblue;
    color: white;
    border-radius: 10px;
    transform: rotateY(180deg);
}
td{
    padding:10px;
    font-size:1.2em;
}
```

Figure 2.2: Font-end code 2

For front-end data validation default functionalities of ASP.NET Core was used. So there was little need of JavaScript code.

3.2 Back-end

The main core of the project was collecting data from the two computer store websites which were done by python scripts. This python scripts contain a library called BeautifulSoup which can be used to parse HTML websites by going through the whole document. By crawling through different classes of HTML elements we gathered the data of multiple components. We got the data of Processors, Motherboards, Rams and Hard Disk Drives.

```
storage = []
descriptions = []
brands = []
pic_links = []

for a in soup.select('.container h2 a'):
    links.append(a['href'])
    model_name, brand_name, description, cap = load_product_model(a['href'])
    brands.append(brand_name)

    names.append(a.get_text(strip=True))
    models.append(model_name)
    descriptions.append(description)
    storage.append(cap)
price_list = soup.find_all('p', attrs={"class":"special-price"})
for a in soup.find_all('a', attrs={"class": "product-image"}):
    if a.img:
        #print(a.img['src'])
        pic_links.append(a.img['src'])

for j in price_list:
    prices.append(j.get_text()[18:])

rows = zip(brands,names, models, links, prices, pic_links, storage, descriptions)

for row in rows:
```

Figure 3.1: Back-end code 1

The collected data was inserted into Microsoft SQL Server Database. Following are few of the data tables:

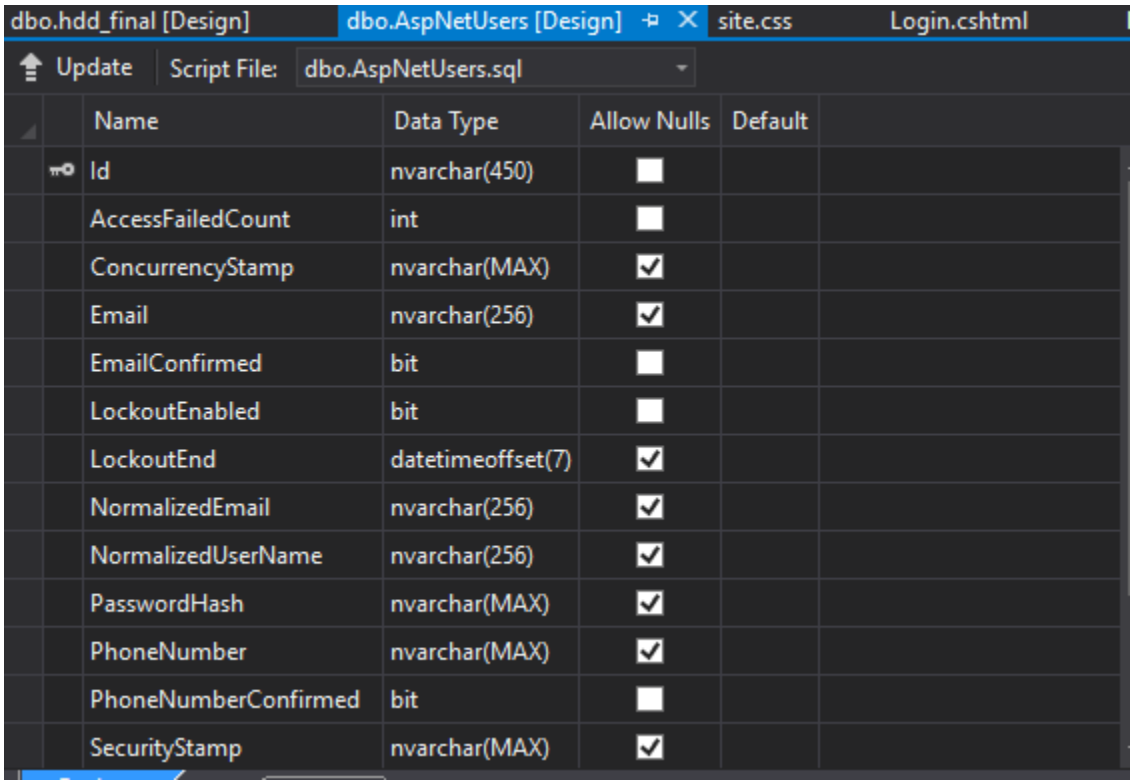


Figure 3.2: User data table generated from Asp.net Identity

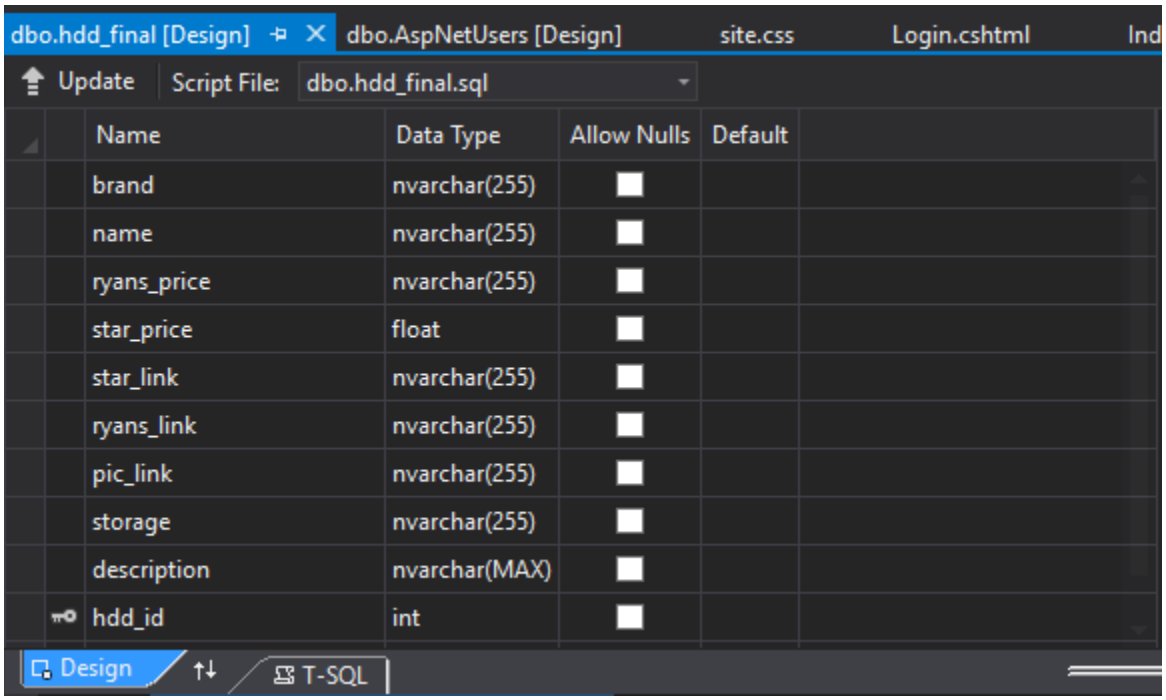


Figure 3.3: Data Table for component hard disk drive

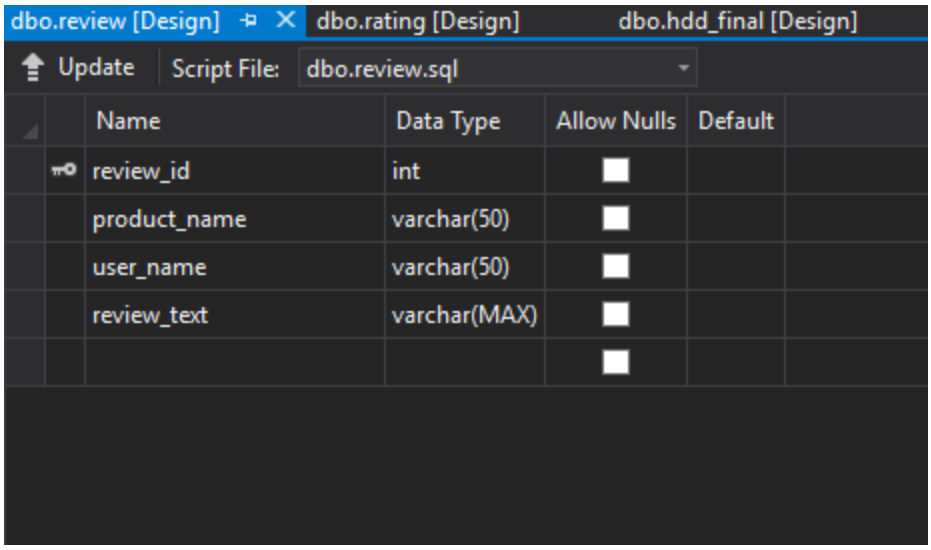


Figure 3.4: Data Table for review

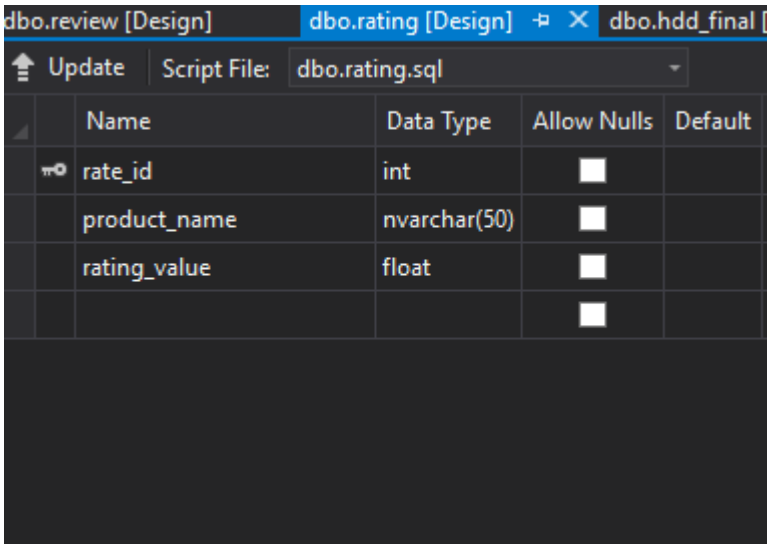


Figure 3.5: Data Table for rating

Now we have used the following classes:

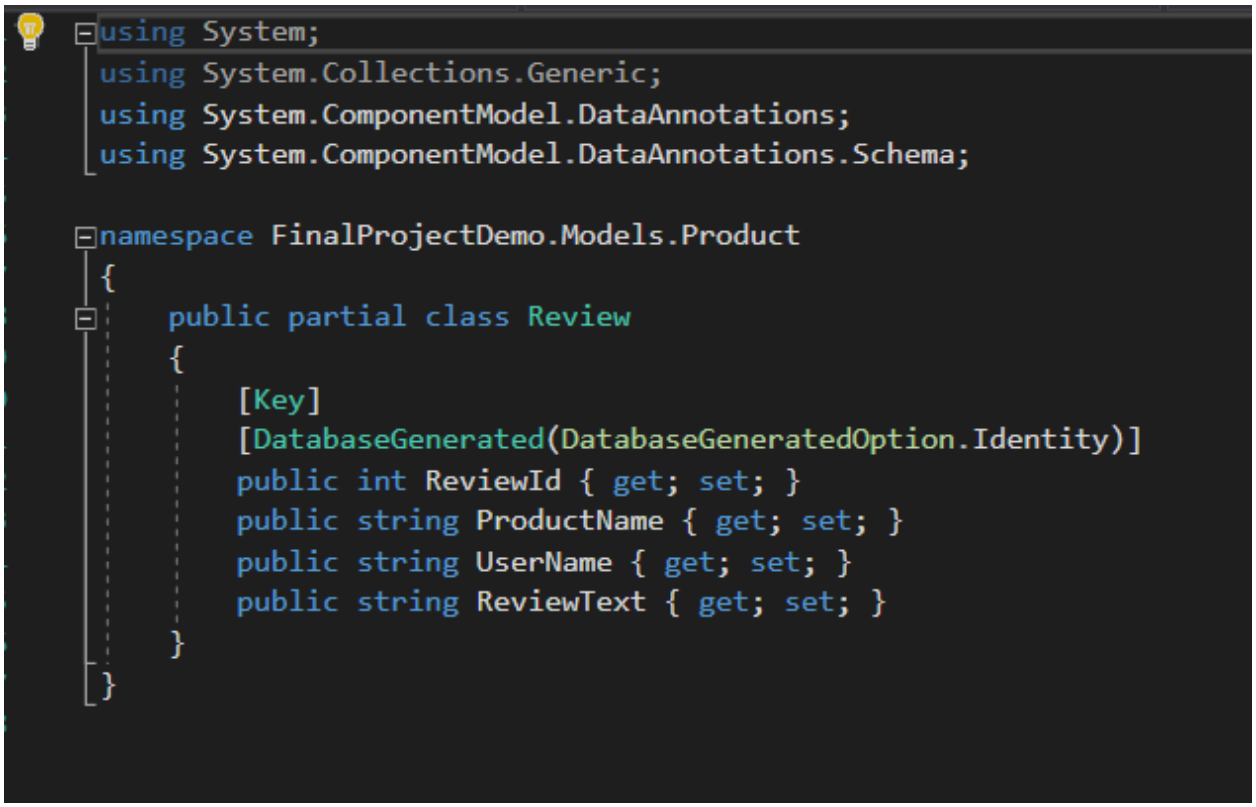
1. User (generated from ASP.NET Core Identity)
2. ProductEntities
   Related to this DbContext class we have, five partial classes:
   1. ProcessorFinal
   2. MoboFinal
   3. HddFinal
   4. Review
   5. Rating

```
using System;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;

namespace FinalProjectDemo.Models.Product
{
    public partial class ProductEntities : DbContext
    {
        public virtual DbSet<HddFinal> HddFinal { get; set; }
        public virtual DbSet<MoboFinal> MoboFinal { get; set; }
        public virtual DbSet<ProcessorFinal> ProcessorFinal { get; set; }
        public virtual DbSet<Rating> Rating { get; set; }
        public virtual DbSet<Review> Review { get; set; }


        public ProductEntities(DbContextOptions<ProductEntities> options)
       : base(options) { }
```

Figure 3.6: DbContext Class ProductEntities

```
using System;
using System.Collections.Generic;

namespace FinalProjectDemo.Models.Product
{
    public partial class ProcessorFinal
    {
        public string Brand { get; set; }
        public string Name { get; set; }
        public double RyansPrice { get; set; }
        public double StarPrice { get; set; }
        public string StarLink { get; set; }
        public string RyansLink { get; set; }
        public string PicLink { get; set; }
        public string Description { get; set; }
        public int ProId { get; set; }
    }
}
```

Figure 3.7: ProcessorFinal Class



```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace FinalProjectDemo.Models.Product
{
    public partial class Review
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public int ReviewId { get; set; }
        public string ProductName { get; set; }
        public string UserName { get; set; }
        public string ReviewText { get; set; }
    }
}
```

Figure 3.8: Review Class

Next we have the Controller Classes

1. AccountController
2. HomeController
3. ProcessorFinalController
4. MoboFinalController
5. HddFinalController
6. ReviewController
7. RatingController

Some of the classes and the methods are written below:

AccountController Class

Method	Description
LogIn()	This method is used to log a user into the system.
Register()	This is used for registration of a new user.
ConfirmEmail()	This method generates the confirmation email for the new registered user.
Logout()	This method is used to log out of the system.
ForgotPassword()	This method is used for handling forgotten password scenario.
ResetPassword()	This method generates the reset password token for resetting the password.

ProcessorFinalController Class

Method	Description
Index()	This method generates the index View for the processors page.
Index_Search()	This method is used to search with keywords.
Details()	This method generates the details View of a specific processor.

CHAPTER 4: User Manual

The major part of the project is the User Interface and experience.

4.1 User-Side

There are five major module for user side. They are- homepage, search, authentication, registration, giving review and rating, create own wishlist and personal account of a member.

4.1.1. HOMEPAGE

Both the member and non-member users get this homepage when they first enter to this website. In figure 4.1.1, we can see the interface of the homepage.

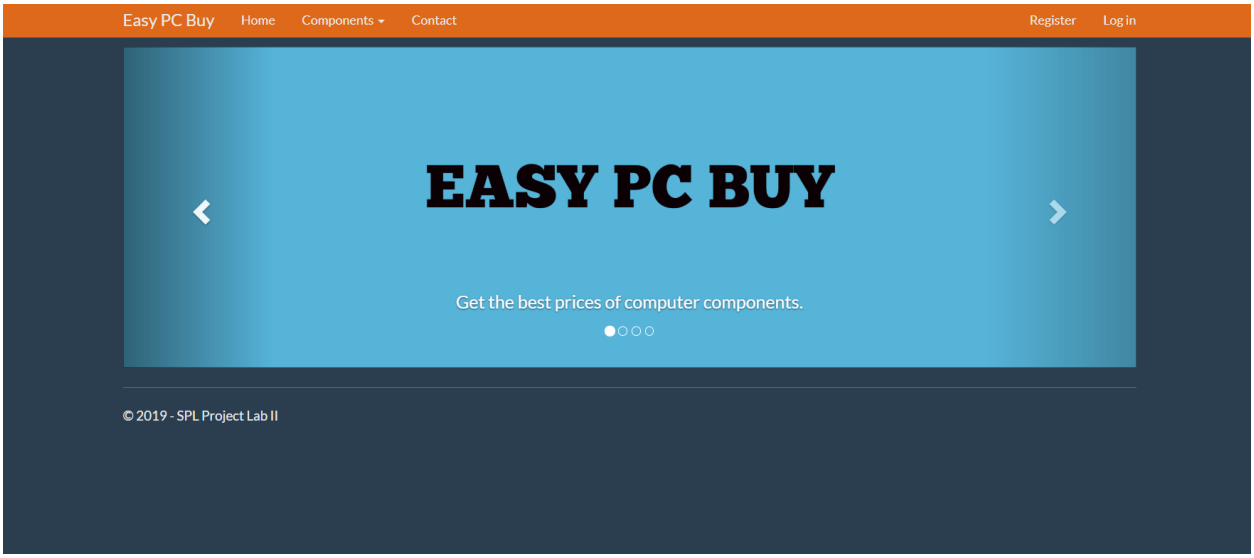


Figure 4.1.1: Homepage of Easy PC Buy

This page has some buttons-

- **Home:** This buttons leads to the main page or homepage of this website.
- **Contact:** With this button user may contact with the admins. In figure 4.1.2, we can see the contact page.

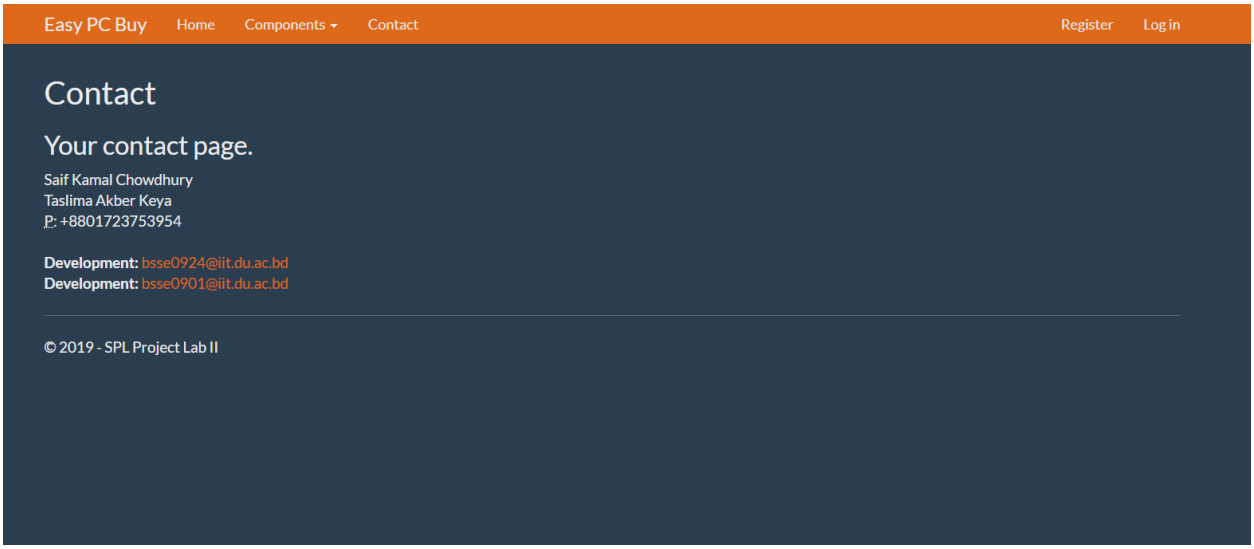


Figure 4.1.2: Contact page

- **Authentication/Registration:** To become a member of Easy PC Buy, users need to register. The registration module is shown in figure 4.1.3 It needs an email address and a password. It sends a confirmation message to the email to verify the email address, which is shown in figure 4.1.4

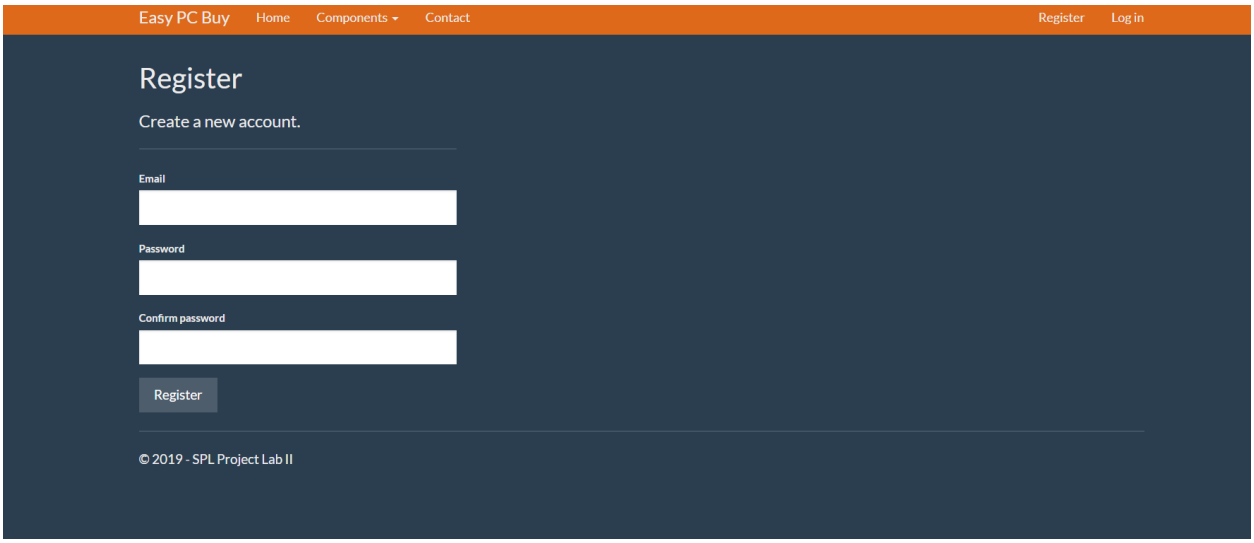


Figure 4.1.3: Registration

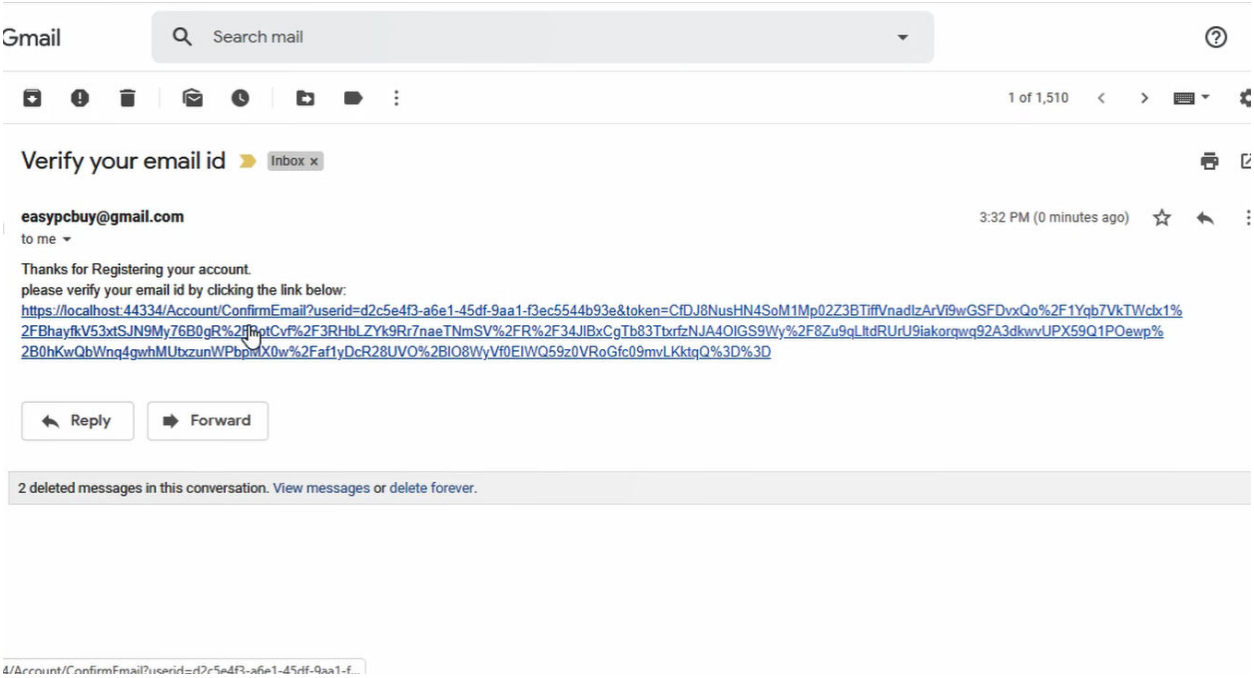


Figure 4.1.4: Registration Confirmation e-mail in mail

After verify the email address it shows a confirmation massage in website too that is shown in figure 4.1.5

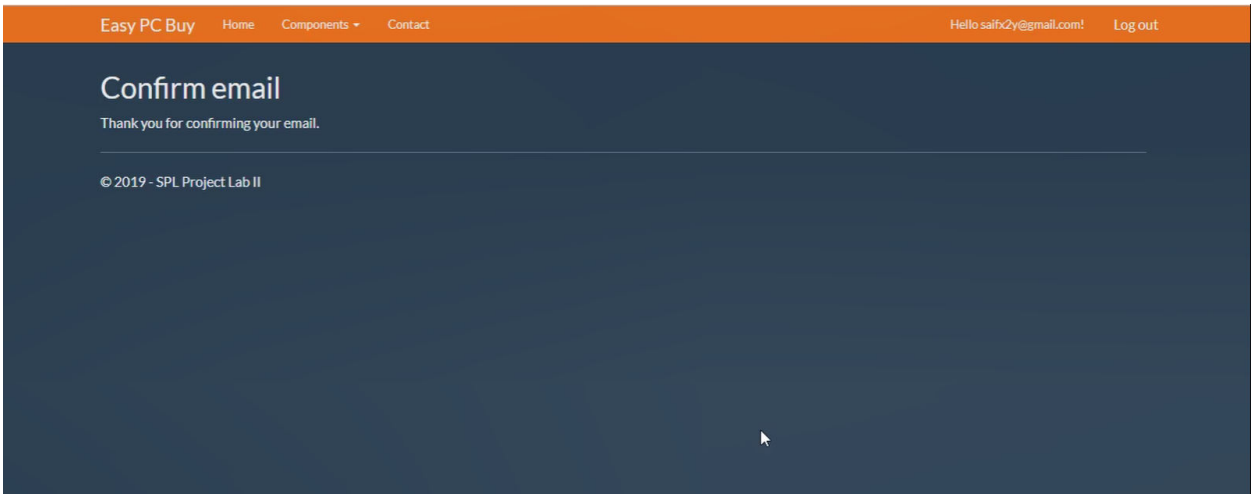


Figure 4.1.5: Confirmation e-mail in website

To enter to the personal account, members need to authenticate. The authentication module is shown in figure 4.1.6. This needs the email address and password of the member to confirm.

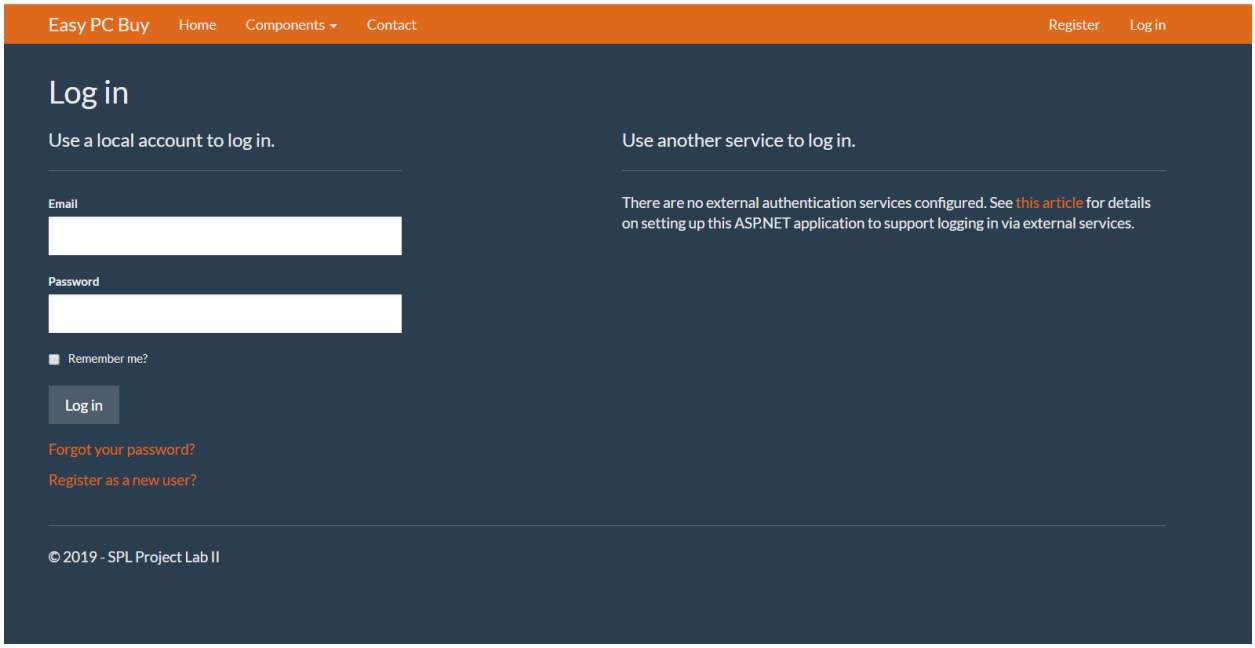


Figure 4.1.6: Authentication of members

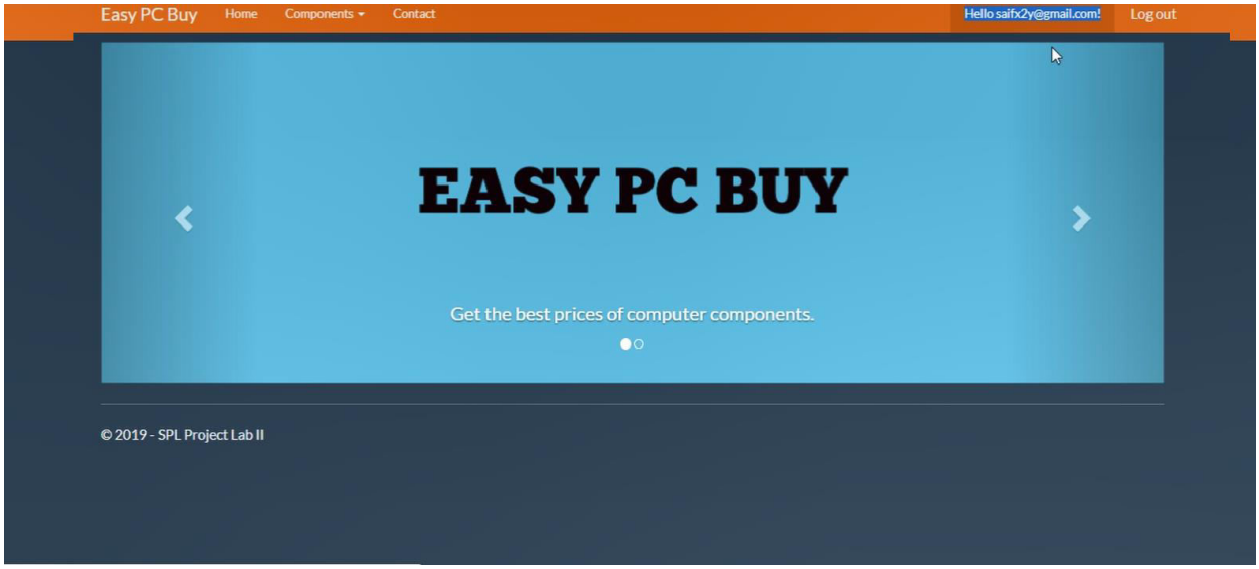


Figure 4.1.7: Log in from user email

- **Password Resetting:** If anyone forgets the password, he/she can click the forget password option to reset the password, and then they need to submit their mail for resetting new password which is shown in figure 4.1.8 and figure 4.1.9.

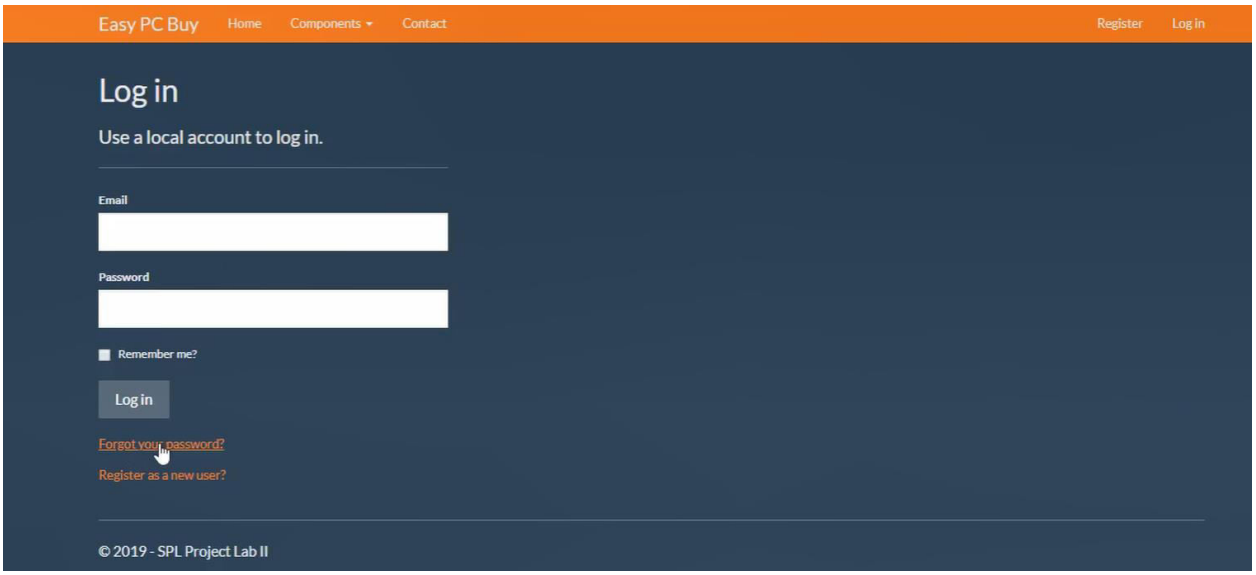
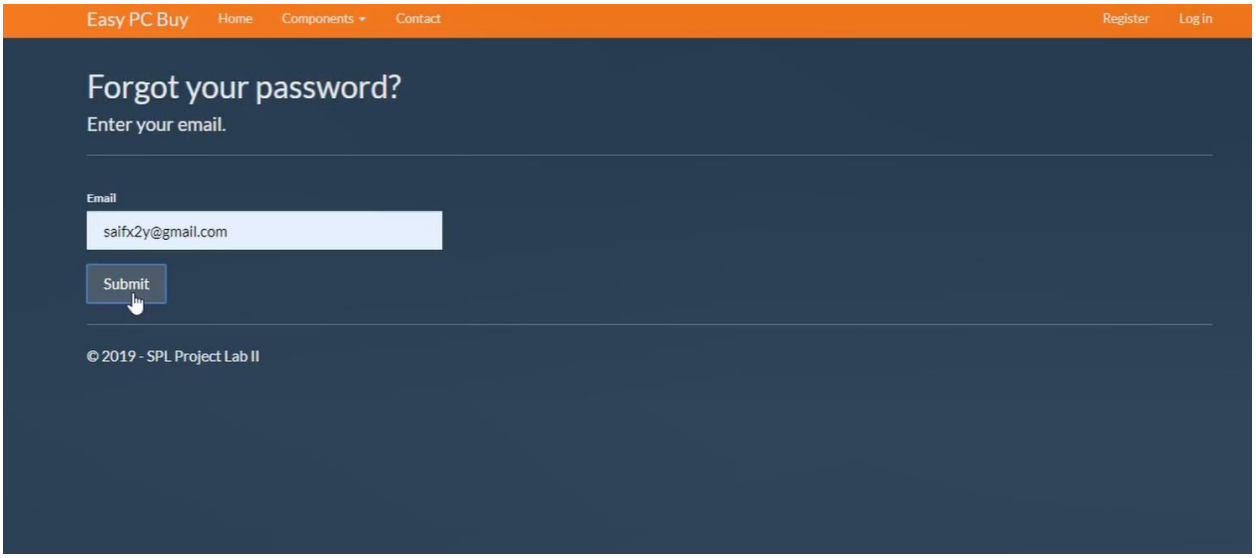


Figure 4.1.8 : Forget Password option

After that, he/she will receive an e-mail containing the password. Figure 2 and Figure 2 shows the email containing a link to reset password. This will lead to the password change option of the member.



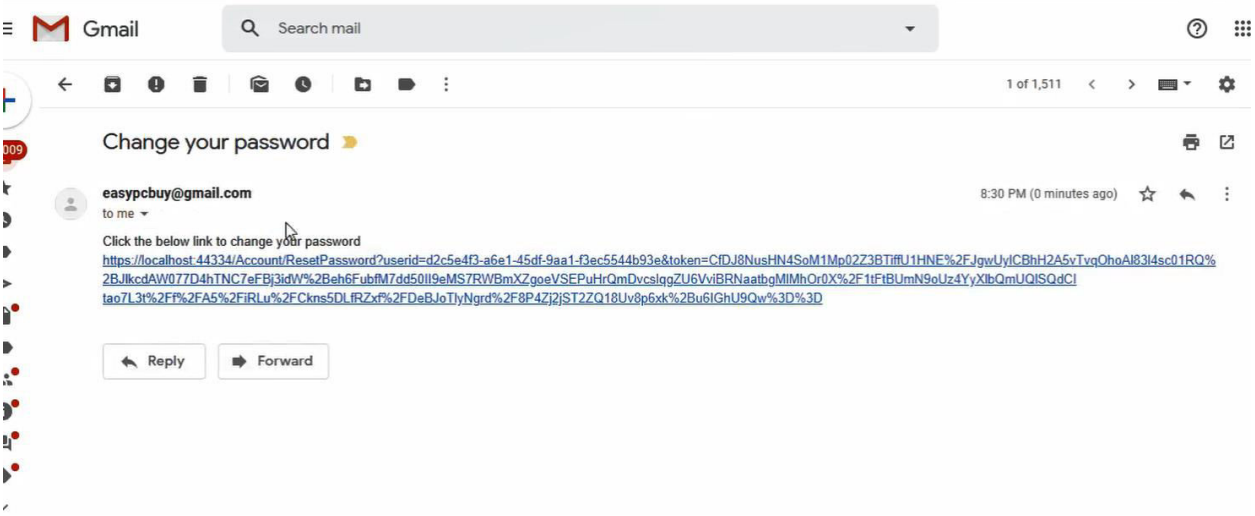


Figure 4.1.9: Password retrieval e-mail

If a user wants to change his/her password he can do so by doing the following. He has to provide his current password then his new password.

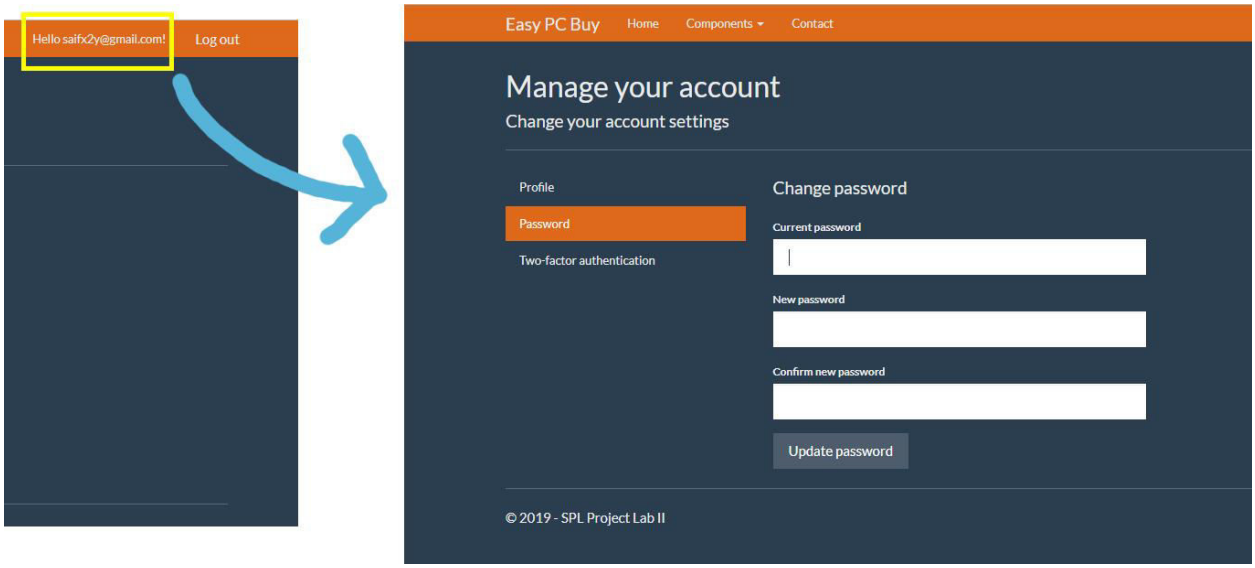


Figure 4.1.10: Changing password

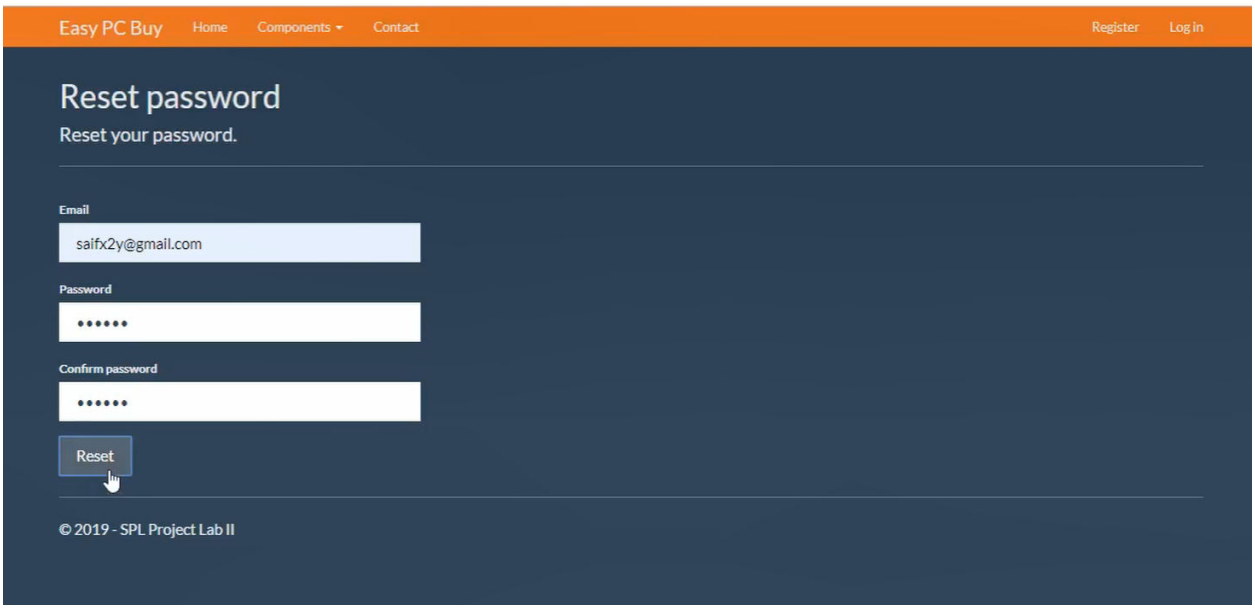


Figure 4.1.11: Password resetting

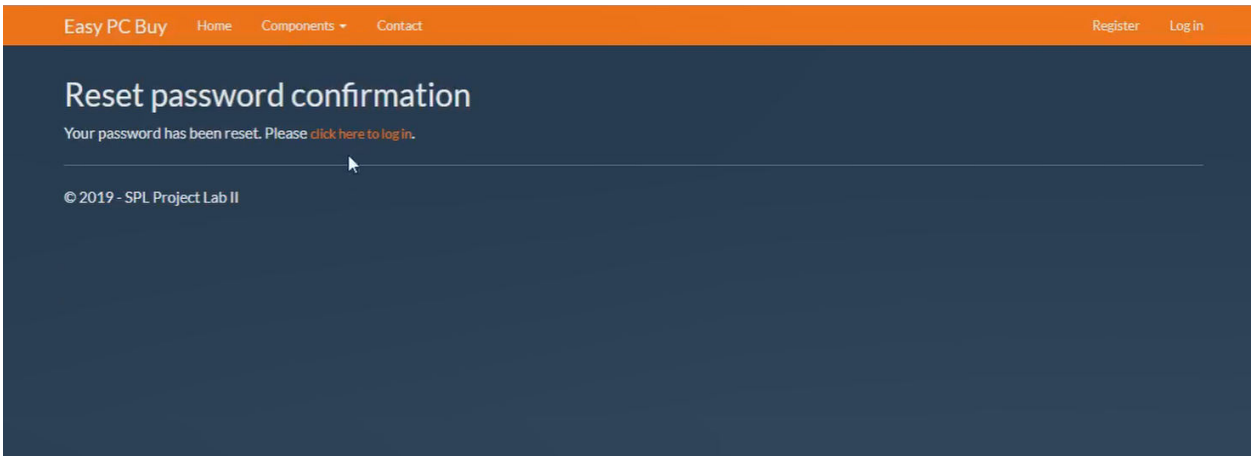


Figure 4.1.12: Reset Password confirmation page

- **Search Component:** User can view four types of component with their price and details from this website. They can select which component they want to search from this section, which is shown in figure 4.1.13.

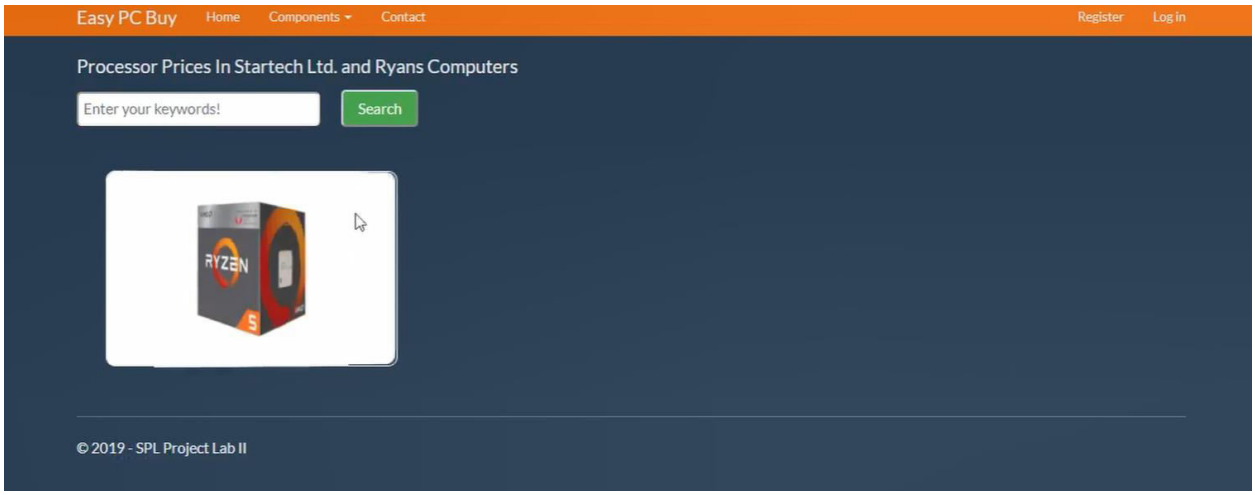


Figure 4.1.13: Searching Components

There are two categories by which users can search for components. Those are-

1. **Keyword based:** Users need to write the keyword for search the component.
2. **Price range based:** Users need to write the price for search the component. This functionality is implemented but not viewed.

Those sections are shown in figure 4.1.14

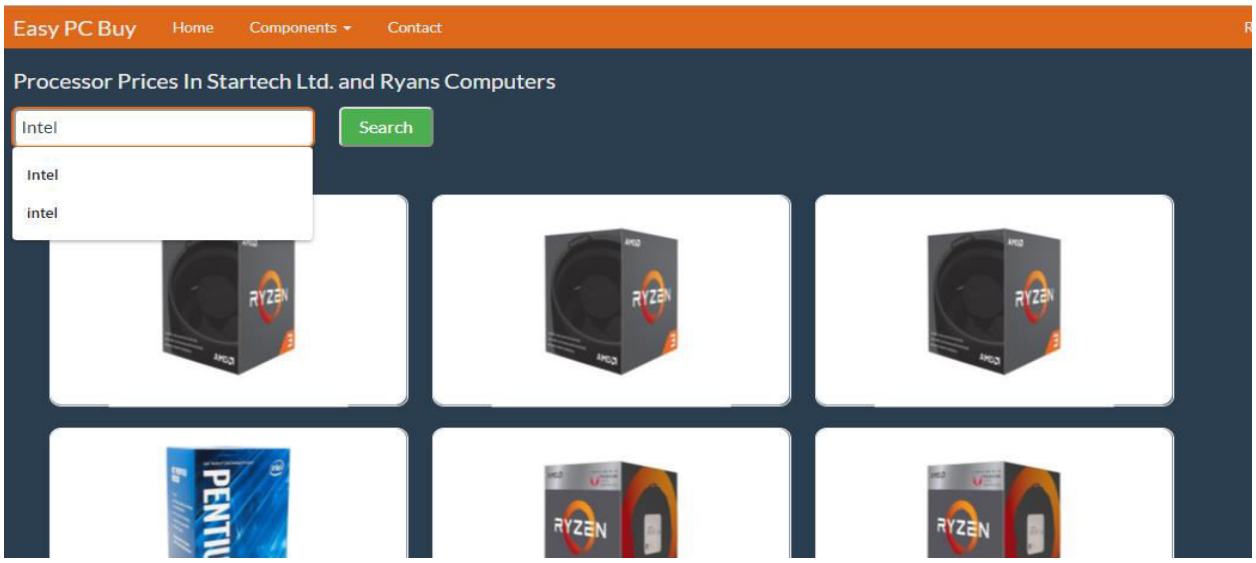


Figure 4.1.14: Component Searching- enter keywords

Then pressing the Search button or Enter key of keyboard generates the search result.

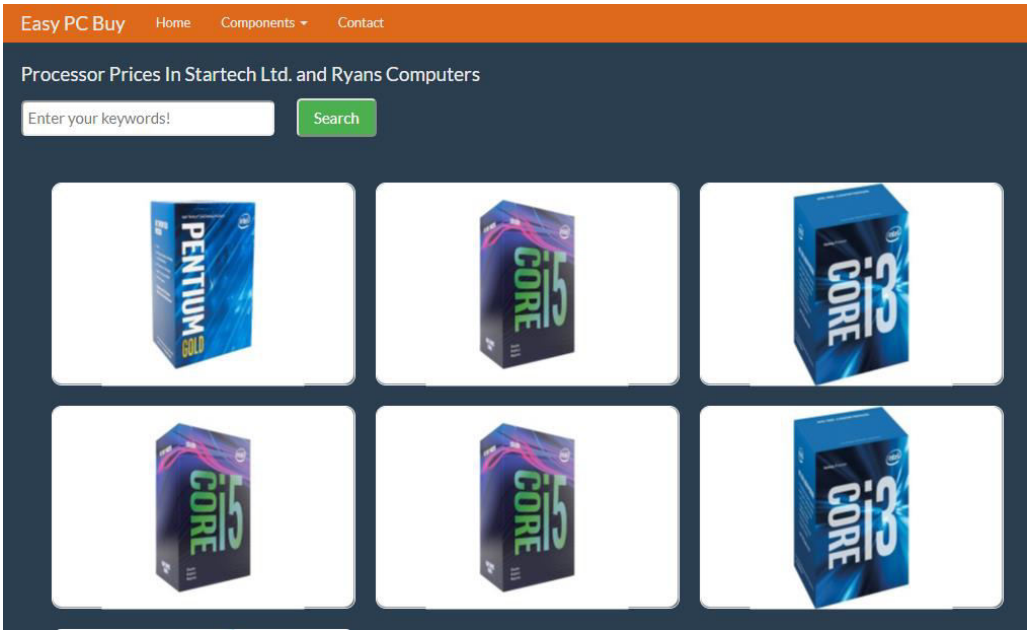


Figure 4.1.15: Search result

- **Component Details:** User can find the components detail with the image of component that is shown in figure 4.1.16.

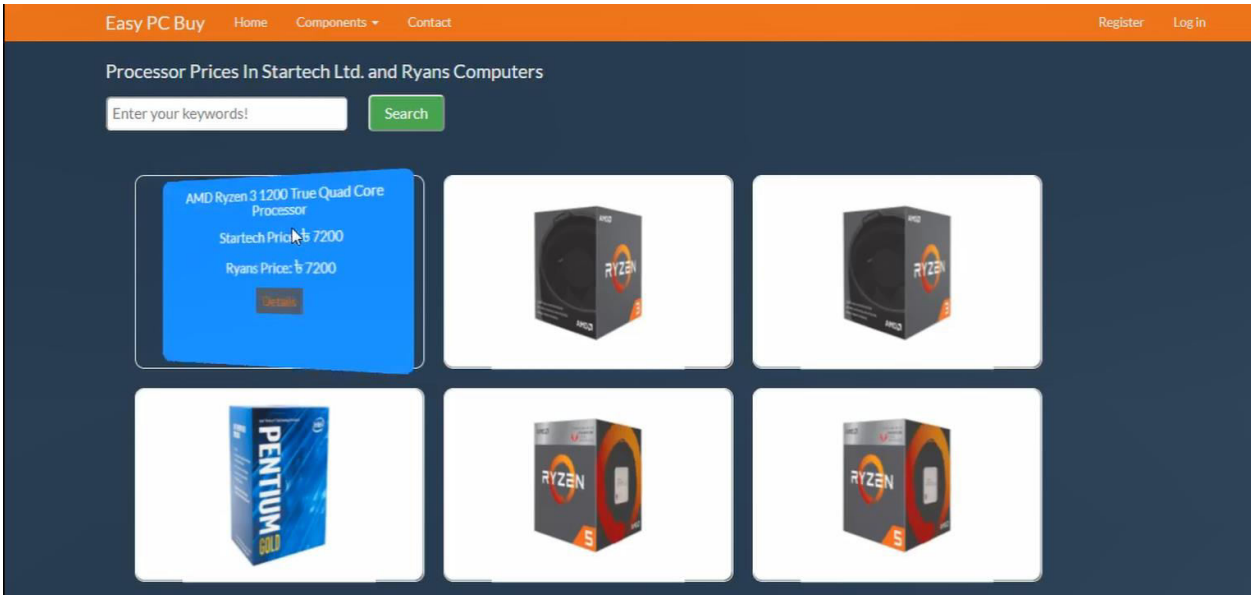


Figure 4.1.16: Components View

On clicking the details view users will be taken to the details page of the component.

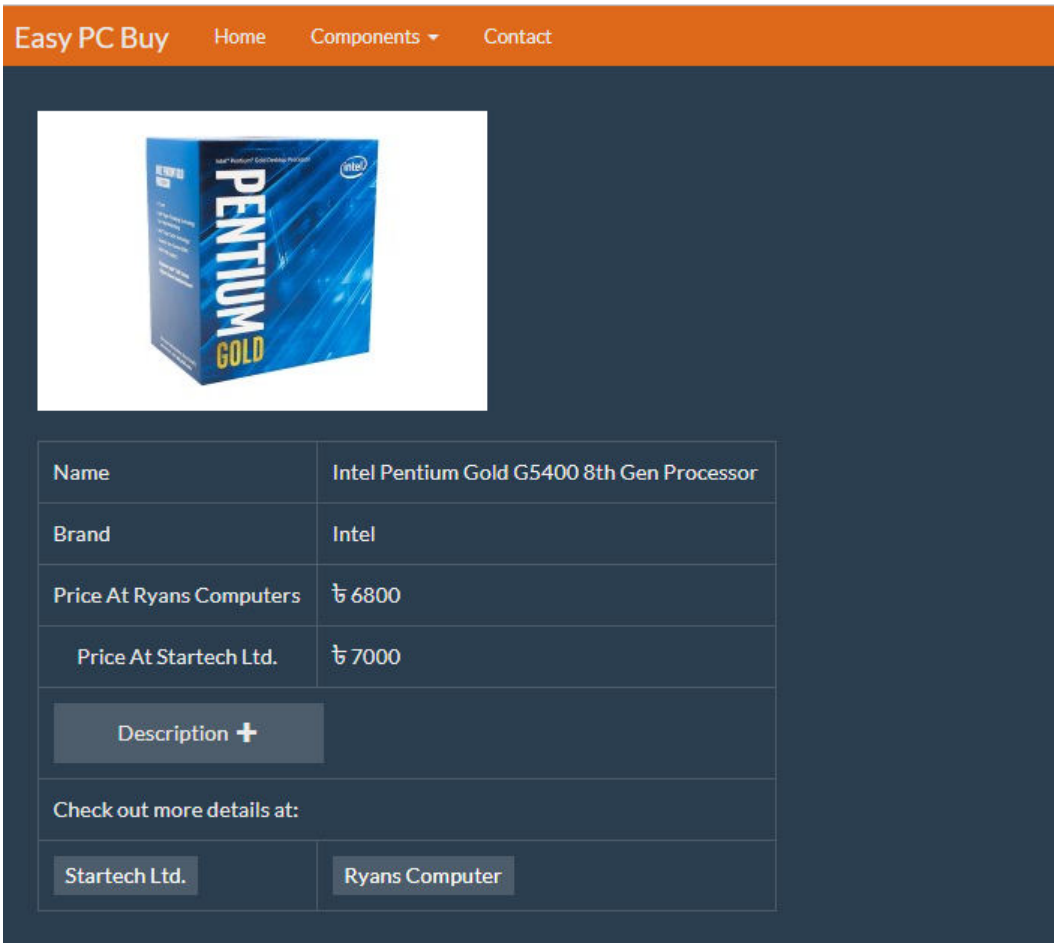


Figure 4.1.17: Details of a component

Now a user may like either the price from Startech or Ryans so he can click the website name to be redirected to that components page in the Store's website.

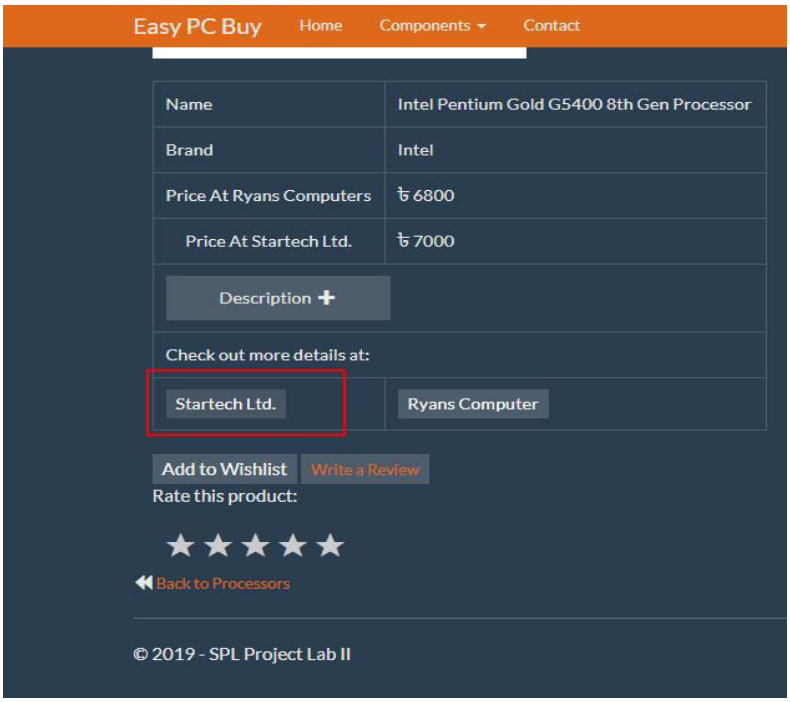


Figure 4.1.18: From the Store's website User can order the product

Chapter 5: Installation Guide of the project

5.1 Pre-requisites

1. Microsoft SQL Server 2008 or newer
2. Visual Studio 2017 or older with Support for ASP.Net Core 2.0

5.2 Guide

We have provided the backup of the database used for the project. Open up SQL server studio and then Databases>Tasks>Restore

In this tab select device and from device select the provided "final_test_database.bak" file.

Now open up the FinalProjectDemo.sln in Visual Studio and go to "appsettings.json" change the connection strings to your connection string for the database in which our database has been restored.

Now go to Tools> Nuget Package Manager>Package Manager Console

Type *dotnet restore*

This will install all the dependencies and Nuget packages required from the internet.

Now you can successfully build the project and run it.

CHAPTER 6: Conclusion and Implementation limitations

A software project means a lot of experience. This chapter summarizes the overall experience gained by the project team during development of a Easy PC Buy. This is a web based real-time web application. Here we needed to handle both the server and user side. This was a great experience for us.

6.1 Achievements

This project helped us to achieve a lot of experiences-

- We have learned a lot about how to develop a software, how to gather and analyze the requirements, the software designing process and finally the implementation phase. It was really a great challenge for us to develop the software.

- It helped us to work as a team

- With this project, we learned to implement web technologies

6.2 Obstacles

During the project development phase, we have faced some obstacles-

1. Since we had no experience in web development sector, It was a great challenge to learn and implement web technology simultaneously
2. ASP.NET core is not a very easy framework to learn, there are many ways to face problems when developing using this framework.
3. Since we had to use database first approach we were unable to implement the following functionalities fully

- rating of the products to saved
- review of the product to be updated in database properly

6.3 Future Plan

We have some future plan to expand this project-

- Give suggestion to the member about related components

- Add a wishlist of components per user

- Member can buy components from our website

We have worked really hard to complete this project. We have got a lot of help from our teachers and seniors. Hopefully, our users will be satisfied with our software.

REFERENCES

1. BeautifulSoup Library: https://www.crummy.com/software/BeautifulSoup/bs4/doc/ [last accessed: 26.05.2019; 10:00 pm]
2. Pandas Library: https://pandas.pydata.org/pandas-docs/stable/ [last accessed: 26.05.2019; 11:30 pm]
3. CSV library: https://docs.python.org/3/library/csv.html[last accessed: 26.05.2019; 11:30 pm]
4. Python 3.X: https://www.python.org/downloads/[last accessed: 25.05.2019; 10:00 am]
5. Pycharm : https://www.jetbrains.com/pycharm/[last accessed: 25.04.2019; 2.00 am]
6. Bootstrap-Hero theme: https://bootswatch.com/superhero/[last accessed: 20.04.2019; 8.00 pm]
7. ASP.Net Core 2.0: https://docs.microsoft.com/en-us/aspnet/core/getting-started/?view=aspnetcore-2.2&tabs=windows[last accessed: 05.05.2019; 7.30 pm]
8. ASP.Net Identity: https://www.pluralsight.com/guides/configuring-asp-net-identity[last accessed: 27.05.2019; 11:30 pm]
9. Bootstrap 3: https://www.w3schools.com/bootstrap/[last accessed: 20.04.2019; 9:30 pm]
10. Microsoft SQL Server: https://support.microsoft.com/en-us/help/321185/how-to-determine-the-version-edition-and-update-level-of-sql-server-an[last accessed: 28.05.2019; 4.30 am]