Student Check-in Kiosk for Institutions Using Asp.net

Mr. D.A. Savindu Manilka Dewpuraarachchi[1*] & Mr. Mohammad Sameer Baig[2]

[1]UG Level-5 Information Technology, [2]Project Coordinator, [1,2]Al Shabaka Institutional Academy, United Arab Emirates.

ABSTRACT

This paper displays the possibility, applicability and usage of developing a user-friendly customized student KIOSK for educational centers. This study showcases how we are able to optimize our defined resources to get a decent outcome which can be very useful and has higher chances of updating.

Keywords: Applicability; Optimize; Customized; Markup; ASP.NET; Rational; Kiosks; Cascading; Validation; Navigated.

1. AIM OF THE STUDY

The initial aim for this study was to create an attendance check in system. But I decided that instead of only adding a simple attendance record system; I am capable of implementing a student kiosk in form of a ASP.NET web form portal. My goal was to develop this Student Kiosk for Al Shabaka Technical Institute located in the heart of Al Quasis, Dubai. Kiosk's technology utilizes touch-based interfaces for easy access and interaction, to improve efficiency of university processes, communication between students and university administration. It will also simplify various day to day operation and enhance the experience of the students and university staffs alike.

2. PURPOSE OF THE STUDY

During my academic years in this institute, I have observed and felt it was something missing in our campus premises. Since ASTI Academy mainly focuses on practical based learning, I questioned myself; why are we still following old fashioned documentations? Shortly I gathered up my ideas and aimed to develop a student kiosk. Building a KIOSK for Al Shabaka Technical Institute can be advantageous for both students and staff. Providing correct information and having a fast moving, smooth environment is a key to success for any organization. As we are now in 21st century every one of us are exposed to Information Technology. In this digital era information can sometimes be best provided in a digitized and interactive way.

3. STUDY SCOPE

With proper fundings and frame work design this study can be moved as a high tech portal where more options can come along with the options which are already present. If it gets the opportunity to be developed as university portal; I believe it will be with high scope and demand. If so we wont be needing to use any out source platforms to manage the academic progress as these KIOSKs will be able to handle those out as well.

In this digital era technology of KIOSKs are believed to be a straight forward ideology that it can be modified/ customized easily according to the client's convenience and requirements. My study outlines and provides the students with registration tools, academic information access, attendance markings. It can be implemented within educational training centers, Institutes, classrooms and maybe even in schools or universities.

4. LITERATURE REVIEW

Usage of proper touched based KIOSK's which are connected to a backend server or a block of code, helps the user to carry out numerous tasks by themselves without getting help from people who are physically present at those premises.

Touch based KIOSKs are being used in many industries or businesses. Such as in Bank, Restaurants, Hospitals, shops, malls, Kiosk systems for blind people, Education and in many other lines of works. As mentioned, popularity of tablet computers and touch enabled laptops are almost similar to KIOSKS are used as proof of this point.

GUI based interactive KIOSK provides the easy access to various services. Kiosks may provide Internet access for web, but KIOSKS do not provide all tasks like a Personal Computer. They perform only a few designed tasks and optimized for remote management in most of the scenarios.

5. USAGE OF TOOLS AND PLATFORMS

Hyper-Text Markup Language (HTML)

The Hyper Text Markup Language or HTML is the standard markup language for documents intended to be shown in a web program. It can be altered using internal or external innovation such as CSS and Java Script. Web browsers get HTML documents from a web server or from neighborhood capacity and render the documents into media web pages. HTML can implant programs written in an employed language for example for my study I have used C#. This influences the way of behaving and content of web pages.HTML was employed in the front-end design of the KIOSK. Several tags were used alongside the ones provided by ASP.NET. Cascading style sheets were also employed.

ASP.NET

ASP.NET web forms have been used to make up the KIOSK. It was engaged in the back-end design of this study. It helped to supply server controls which includes the HTML server controls as well as SQL server controls which I have explained below.

I have also implemented ASP.net web server controls as well as the Validation Server Controls (for input validation). Most of the controls in ASP.NET were employed in this study. I have considered the security of my web form in a ordinary manner. I was able to provide its security by using Login control (which provides login functionality), LoginStatusControl (to control the login status), the LoginName control (to display the current user's name), a LoginView control, to provide different views depending on login status.

C sharp Language (C#)

C# was employed as the programming language for back-end coding file. C# has been used for button click events, Event handlers, As well as connection strings for SQL database. C# makes it easy to employ all of the web pages and KIOSK application as a whole. In addition to ASP.NET packages and SQL servers being used as backend functions C# can also be used to implement methods.

Microsoft SQL Server 2018

This is the relational database management server that is used in the portal. It stores data in the form of tables

Basically, there are twelve unique versions of Microsoft SQL Servers to be found on internet today. Each of the versions target various markets. This depends on little single-machine applications to huge Web confronting applications with numerous simultaneous clients. In order to accomplish my backend server requirement, I have used Microsoft SQL Server Management 2018

6. WORKING PRINCIPLE

Working of student kiosk application runs through various sections of ASP.NET. The client is able to run the .sln extension file from any of the debugging operands. For this instance, I have used Google Chrome as the debugging operator. As soon as debugging starts the Home.apsx file gets executed as it my first page load option. Then the user is navigated into the Kiosk's Home page providing them with two login options.

If the client is a registered student he or she should click on "Student Login" button. Instantly it opens up "Student Login.aspx" and provides the option to type in the Student ID and Password.

If the user is a new user (student) he or she is provided with sign up option. When user clicks on "Sign up" it opens up "Student Registration.aspx" file. The client is able to view four text boxes, four labels and two buttons. When the required fields are entered data gets fetched towards the backend SQL Database. The connection string added in backend C# code allows the users to complete the registration process successfully. Users gets to view a feedback message as well. Upon successful login the students gets navigated towards the "StudentDashboard.aspx"
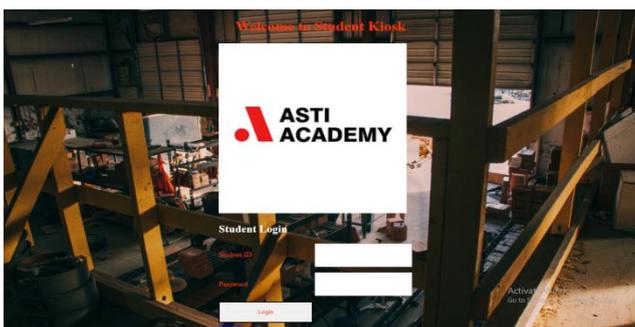
6.1. Using the KIOSK

Figure 1.1. Home Screen

Figure 1.2. Log in Screen
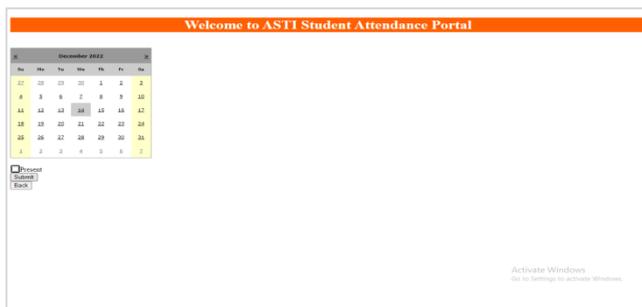
Figure 1.3. Student Dashboard

Figure 1.4. Attendance Portal

When a student logs in they are navigated towards the student dashboard (Fig. 1.3) then they are capable of selecting any of the relevant buttons in order to fulfill their requirement. Initial objective of this study was to develop an attendance marking portal for the students; as shown in Fig 1.4 now students are able to mark their attendance by just a click away. The HTML div tags and C# labels displays a welcome message and a calendar along with a checkbox property stetted to mark the attendance. When the user clicks on submit Button. "BtnSub" the relevant data gets passed on the SQL Database through the connection string pre-established.

Figure 1.5. Attendance Portal – Techers View

You may cross paths with the thought of "What if students add fake or duplicate attendance records". There is no need to worry as you can see in Fig 1.5 the attendance records which students add gets viewed and verified by admins before they gets added to the main attendance database (Fig.1.6)
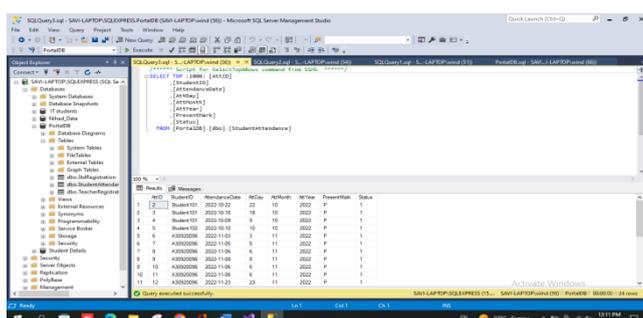

Figure 1.6. Attendance Database

www.iijsr.com

Declarations

Source of Funding

Competing Interests Statement

The authors declare no competing financial, professional, or personal interests.

Consent for publication

The authors declare that they consented to the publication of this research work.

References

[1] Zameer, Ahmed & Ambreen, Sadaf & Muhammad, Nazrul & Mohammadi, Mohammad. (2017). Developing KIOSKs-Touch Based Interactive Information System. International Journal of Computer Science and Information Security (IJCSIS), 15: 166-177.

[2] Adewale, Adeyinka. (2013). Design and Development of a University Portal for the Management of Final Year Undergraduate Studys. International Journal of Engineering and Computer Science, 2: 2911-2920.

[3] A tour of C#-Overview. Microsoft Learn. https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

[4] What is Visual Studio? Incredibuild. https://www.incredibuild.com/integrations/visual-studio.

[5] What is Web Forms. (2022). Microsoft Learn.

[6] https://learn.microsoft.com/en-us/aspnet/web-forms/what-is-web-forms.

[7] Wikipedia contributors. (2022). Microsoft SQL Server. https://en.wikipedia.org/wiki/Microsoft_SQL_Server.

[8] www.youtube.com. C# Tutorial - How to Connect and Use Local Database in Visual Studio 2019 | FoxLearn. [online] Available at: https://www.youtube.com/watch?v=mgtfxtjKoaA.

Appendix - A

Source Code of Home.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Home.aspx.cs"
Inherits="Asti_Portal.Home" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
<style type="text/css">
.auto-style1 {
```

```
width: 100%;

}

</style>

</head>

<body style="background-image:url('background1.jpg'); background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;">

<form id="form1" runat="server">

<div style="margin:0px 450px 20px 480px">

<asp:Label ID="lblTitle" runat="server" ForeColor="#FF3300" font-size="Large"></asp:Label>

</div>

<div style="margin:20px 450px 20px 450px">

<asp:Image ID="Image1" runat="server" ImageUrl="~/Astii.jpg" Width="400px" />

</div>

<div style="margin:0 450px 20px 450px">

<asp:Button ID="btnStudent" runat="server" BorderStyle="Solid" ForeColor="#FF0066"

OnClick="btnStudent_Click" Text="I am Student" Width="200px" Height="50px" />

<asp:Button ID="btnTeacher" runat="server" BorderStyle="Solid" ForeColor="#FF0066"

OnClick="btnTeacher_Click" Text="I am Teacher" Width="200px" Height="50px" />

</div>

</form>

</body>

</html>
```

Appendix - B

Source Code of Student Login.aspx

```
%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StudentLogin.aspx.cs"

Inherits="Asti_Portal.StudentLogin" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<style type="text/css">

.auto-style1 {

width: 100%;
```

```
height: 242px;

}

</style>

</head>

<body style="background-image:url('background1.jpg');  background-repeat: no-repeat;

background-attachment: fixed;

background-size: cover;">

<form id="form1" runat="server">

<center><h1 style="color:#FF3300">Welcome to Student Kiosk </h1></center>

<div style="margin:20px 450px 20px 450px">

<asp:Image ID="Image1" runat="server" ImageUrl="~/Astii.jpg" Width="400px" />

<h2 style="color:#FFffff">Student Login</h2>

</div>

<div style="margin:20px 450px 20px 450px">

<asp:Label ID="Label1" runat="server" Text="Student ID" ForeColor="#FF3300" Width="200px"
Height="50px"></asp:Label>

<asp:TextBox ID="txtStudentID" runat="server" Width="200px" Height="50px"></asp:TextBox>

<asp:Label ID="Label2" runat="server" Text="Password" ForeColor="#FF3300" Width="200px"
Height="50px"></asp:Label>

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password" Width="200px"
Height="50px"></asp:TextBox>

<br />

<asp:Button ID="btnLogin" runat="server" BorderStyle="Solid" ForeColor="#FF3300" Text="Login"
Width="200px" Height="50px" OnClick="btnLogin_Click" />

<br />

<asp:LinkButton ID="btnSignUp" runat="server" OnClick="LinkButton1_Click">Sign up</asp:LinkButton>

<br />

<asp:Label ID="lblmessage" runat="server" ForeColor="Red"></asp:Label>

</div>

<asp:Panel ID="Panel1" runat="server">

</asp:Panel>

</form>

</body>
```

</html>

Appendix - C

Source Code of Teacher Login.aspx

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Web.Security;

namespace Asti_Portal

{

public partial class TeacherLogin : System.Web.UI.Page

{

protected void Page_Load(object sender, EventArgs e)

{

}

protected void btnLogin_Click(object sender, EventArgs e)

{

SqlConnection con = new

SqlConnection(ConfigurationManager.ConnectionStrings["ASTIConnectionString"].ToString());

try

{

string uid = txtteacherID.Text;

string pass = txtPassword.Text;

con.Open();

string qry = "select * from TeacherRegistration where TeacherId='" + uid + "' and TeaPassword='" + pass + "'";

SqlCommand cmd = new SqlCommand(qry, con);

SqlDataReader sdr = cmd.ExecuteReader();

if (sdr.Read())

{

Response.Redirect("~/TeacherDashboard.aspx");
```

```
}

else

{

lblmessage.Text = "User Id & Password Is not correct Try again..!!";

}

con.Close();

}

catch (Exception ex)

{

Response.Write(ex.Message);

}

}

}

}
```