

INTERNSHIP REPORT

A report submitted in partial fulfillment of the requirements for the Award of Degree of

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING**

**by
K.SIREESHA
Regd. No.: 14A51A0565**

**Under Supervision of
Mr. Srinivas, HR
NanoMindz Technologies pvt.Ltd,
Vishakapatnam.
(Duration: 8th May, 2017 to 7th June, 2017)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(An Autonomous Institution)

Approved by AICTE, Permanently affiliated to JNTU, Kakinada

TEKKALI, ANDHRA PRADESH 2013

2014 – 2018

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(An Autonomous Institution)
TEKKALI



CERTIFICATE

This is to certify that the “**Internship report**” submitted by **K.SIREESHA(Regd. No.: 14A51A0565)** is work done by her and submitted during 2017 – 2018 academic year, in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING**, at **NanoMindz Technologies pvt.Ltd, Vishakapatnam.**

College Internship Coordinator
Dr. B. Rajesh

Department Internship Coordinator
Sri L.V. Satyanarayana, M.Tech.
Assistant professor, CSE

Dr.G.S.N.Murthy
M. Tech.,Ph.D.
Head of the Department
Department of CSE



NANOMINDZ TECHNOLOGIES PVT. LTD.
(An ISO 9001:2008 Certified Company)



CERTIFICATE OF INTERNSHIP

This is to certify that Ms. Kapala Sireesha, student of ADITYA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, Tekkali, having ID No. 14A51A0565 has successfully completed the internship programme from 08th May 2017 to 07th June 2017 in our organization Nanomindz Technologies Pvt. Ltd. at Visakhapatnam.

Signature of Authorized



ACKNOWLEDGEMENT

First I would like to thank Mr.Srinivas, HR, Head, of **NANOMINDZ, Vishakapatnam** for giving me the opportunity to do an internship within the organization.

I also would like all the people that worked along with me **NANOMINDZ, Vishakapatnam** with their patience and openness they created an enjoyable working environment.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I am highly indebted to Director **Prof.V.V. NageswaraRao** and Principal **Dr. K. B. MadhuSahu**, for the facilities provided to accomplish this internship.

I would like to thank my Head of the Department **Dr. G.S.N.Murthy** for his constructive criticism throughout my internship.

I would like to thank **Dr. B. Rajesh**, College internship coordinator **Sri.L.V Satyanarayana** internship coordinator Department of CSE for their support and advices to get and complete internship in above said organization.

I am extremely great full to my department staff members and friends who helped me in successful completion of this internship.

K.SIREESHA
(14A51A0565)

ABSTRACT

Business intelligence (BI) systems depend on efficient integration of disparate and often heterogeneous data. The integration of data is governed by data-intensive flows and is driven by a set of information requirements. Designing such flows is in general a complex process, which due to the complexity of business environments is hard to be done manually. In this paper, we deal with the challenge of efficient design and maintenance of data-intensive flows and propose an incremental approach, namely Co AI, for semi-automatically consolidating data-intensive flows satisfying a given set of information requirements. Co AI works at the logical level and consolidates data flows from either high-level information requirements or platform-specific programs. As Co AI integrates a new data flow, it opts for maximal reuse of existing flows and applies a customizable cost model tuned for minimizing the overall cost of a unified solution. We demonstrate the efficiency and effectiveness of our approach through an experimental evaluation using our implemented prototype.

Organisation Information:

NANOMINDZ is a professionally managed company with years of industry experience in developing and delivering Enterprise specific Software and Web development solutions using latest technologies. Quality is the buzz word in today's world without which no organization can survive. Along with quality we at NANOMINDZ. "Think Beyond" to take one step ahead and focus on Delivery of the solutions. We design processes that focus not just only on quality but also on delivery which increases the value to our global clients. Apart from training our employees on latest technologies, we also empower them to deliver exciting solutions to our clients. At the core NANOMINDZ operates in three specific domains namely Software Development, Website Design & Development and Geographic Information Services. We also offer our services in building E-Commerce solutions, Search Engine Optimization (SEO) and Database Administration services. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. We emphasize on building relationships with our clients by delivering projects on time and within budget.

Programs and opportunities:

This ground up approach helps us deliver not only the solution to our clients but also add value to At the core NANO MINDZ operates in three specific domains namely Software Development, Website Design& Development and Geographic Information Services. We also offer our services in building E-Commerce solutions, Search Engine Optimization (SEO) and Database Administration services. Under each division we further provide specific industry solutions on focused domains with cutting edge technologies. We emphasize on building relationships with our clients by delivering projects on time and within budget.

Methodologies:

We follow a structured methodology for our projects which starts from designing the solution to the implementation phase. Well planned Project reduces the time to deliver the project and any additional ad-hoc costs to our clients, hence we dedicate majority of our time understanding our clients business and gather requirements. This ground up approach helps us deliver not only the solution to our clients but also add value to your investments.

Key parts of the report:

Under each division we further provide specific industry solutions on focused domains with cutting edge technologies.

Benefits of the Company/Institution through our report:

Under each division we further provide specific industry solution on focused domains with cutting edge technologies. We emphasize on building relationships with our clients by delivering projects on time and within budget.

INDEX

S.no	CONTENTS	Page no
1.	Introduction.....	1
1.1	Modules.....	2
2.	Analysis.....	3
3.	Software requirements specifications	4
4.	Technology.....	5
4.1	ASP.NET.....	5
4.2	ADP.NET	6
4.3	C#.NET.....	7
4.4	JAVA.....	7
4.5	SQL Data Base.....	8
5.	Coding.....	10
6.	Screenshots.....	11
7.	Conclusion.....	16
8.	Bibilography.....	17

Learning Objectives/Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
- Some internship is used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1st WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	8/05/17	Monday	Introduction of .NET frame work
	9/05/17	Tuesday	Features of .NET { IDE,CLR,CTS }
	10/05/17	Wednesday	Introduction of C# .NET
	11/05/17	Thursday	Introduction of console application
	12/05/17	Friday	Continuing console applications
	13/05/17	Saturday	Understanding different types of class and collection

2nd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	15/05/17	Monday	Introduction to windows application
	16/05/17	Tuesday	Understanding controls like [label, button] etc.,
	17/05/17	Wednesday	Continuing understanding controls
	18/05/17	Thursday	Introduction to windows services
	19/05/17	Friday	Continuing windows services
	20/05/17	Saturday	Introduction to class libraries

3rd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	22/05/17	Monday	Introduction .NET REMOTING
	23/05/17	Tuesday	Introduction to web programming
	24/05/17	Wednesday	Understanding what is Internet website, web request and web response
	25/05/17	Thursday	Understanding client side web technologies VS server side web technologies
	26/05/17	Friday	Introduction to HTML
	27/05/17	Saturday	HTML continued

4th WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	29/05/17	Monday	Understanding classic ASP VS ASP.NET
	30/05/17	Tuesday	Introduction to ASP.NET web server controls
	31/05/17	Wednesday	Working with different web server controls
	01/06/17	Thursday	Standard controls , validation controls
	02/06/17	Friday	Introduction to data control in depth
	03/06/17	Saturday	Project session

5th WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	05/06/17	Monday	Design& Analysis
	06/06/17	Tuesday	Coding
	07/06/17	Wednesday	Testing

1. INTRODUCTION

The complexity of business environments constantly grows, both with regard to the amount of data relevant for making strategic decisions and the complexity of included business processes. Today's dynamic and competitive markets often imply rapid (e.g., near real-time) and accurate decision making. Relevant data are stored across a variety of data repositories, possibly using different data models and formats, and potentially crossed with numerous external sources for various context aware analysis. A data integration process combines data residing on different sources and provides unified view of this data for a user [1]. For example, in a data warehousing (DW) context, data integration is implemented through extract-transform-load (ETL) that extracts, cleans, and transforms data from multiple, often heterogeneous data sources and Finally, delivers data for further analysis. There are various challenges related to data Flow design. Here we consider two: design evolution and design complexity.

A major challenge that BI decision-makers face relates to the evolution of business requirements. These changes are more frequent at the early stages of a DW design project and in part, this is due to a growing use of agile methodologies in data Flow design and BI systems in general. But changes may happen during the entire DW lifecycle. Having an up-and-running DW system satisfying an initial set of requirements is still a subject to various changes as the business evolves. The data Flows populating a DW, as other software Artefacts, do not lend themselves nicely to evolution events and in general, due to their complexity, maintaining them manually is hard. The situation is even more critical in today's BI settings, where on-the-fly decision making requires faster and more efficient adapting to changes. Changes in business needs may result in new, changed or removed information requirements. Thus having an incremental and agile solution that can automatically absorb occurred changes and produce a Flow satisfying the complete set of requirements would largely facilitate the design and maintenance of data-intensive Flows.

In an enterprise environment data is usually shared among users with varying technical skills and needs, involved in different parts of a business process. Typical real-world data-intensive workloads have high temporal locality, having 80% of data reused in a range from minutes to hours. However, the cost of accessing these data, especially in distributed scenarios, is often high. At the same time, intertwined business processes may also imply overlapping of data processing. For instance, a sales department may analyze the revenue of the sales for the past year, while Finance may be interested in the overall net profit. Computing the net profit can largely benefit from the total revenue already computed for the sales department and thus, it could benefit from the sales data Flow too. The concept of reusing partial results is not new. Software and data reuse scenarios in data integration have been proposed in the past, showing that such reuse would result in substantial cost savings, especially for large, complex business environments. Data Flow re use could result in a significant reduce in design complexity, but also in intermediate Flow executions and thus, in total execution time too.

1.1 Module Description:

Shopper:

=====

- 1) User registers the site.
- 2) Products will be showed
- 3) If user selected the product and then save
- 4) User selected product is send to the Order.
- 5) If user wants to buy the product they can also buy.

Supplier

=====

- 1) send product details
- 2) send payment verification
- 3) Store buying detail
- 4) Store line items using join product and order

Order

=====

- 1) Store all the user details.
- 2) Order detail

2. SYSTEM ANALYSIS

2.1 Requirement Analysis

Existing System:

In an Existing we address these challenges and present an approach to efficient, incremental consolidation of data-intensive flows. Following common practice, our method iterates over information requirements to create the final design. we show how to efficiently accommodate a new information requirement to an existing design and also, how to update a design in lieu of an evolving information requirement. The final design satisfying all requirements comprises a multi-flow. As 'coal' is formed after the process and extreme compaction of layers of partially decomposed materials¹, Co AI processes individual data flows and incrementally consolidates them into a unified multi-flow.

Proposed System

Following the previously proposed set of flow transformations in the context of ETL processes in Co AI we extend this set considering also the associative property of n-array operations (e.g., Join) and thus rely on the following four flow transformations used for reordering the operations. Swap Applied to a pair of adjacent unary operations, it interchanges the order of these operations. Distribute/Factorize. Applied on a unary operation over an adjacent n-array operation, it respectively distributes the unary operation over the adjacent nary operation or factorizes several unary operations over the adjacent n-array operation. Merge/Split. Applied on a set of adjacent unary operations, it respectively merges several operations into a single unary operation or splits a unary operation into several unary operations. Re-associate. Applied on a pair of mutually associative n-array operations, it interchanges the order in which these operations are executed.

3. SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 System configurations

The software requirement specification can produce at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by established a complete information description, a detailed functional description, a representation of system behavior, and indication of performance and design constrain, appropriate validate criteria, and other information pertinent to requirements.

Software Requirements:

- Operating system : Windows 7 Ultimate.
- Coding Language : MVC 4 Razor
- Front-End : Visual Studio 2012 Professional.
- Data Base : SQL Server 2008.

Hardware Requirement:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 1TB.
- Ram : 4GB.

4. TECHNOLOGY

4.1 ASP.NET

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices.

ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

ASP.NET is a part of Microsoft .NET platform ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .NET framework. These codes can use the entire hierarchy of classes in .NET framework.

ASP.NET web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for tasteful processing. The server processes the output of the client actions and triggers the reactions.

Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:

- Page state
- Session state

The page state is the client state, i.e., the content of various input fields in the web form. The session state is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take an example of a shopping cart.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown on a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

The ASP.NET runtime carries the page state to and from the server across page requests while generating ASP.NET runtime codes, and incorporates the state of the server side components in hidden fields.

This way, the server becomes aware of the overall application state and operates in a two-tiered connected way.

The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:

ASP.NET is a technology, which works on the .NET framework that contains all web-related functionalities. The .Net frame work is made of an object-oriented hierarchy. An ASP.NET web applications is made of pages. When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system.

The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework. Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

4.2 ADO.NET

As you develop applications using ADO.NET, you will have different requirements for working with data. In some cases, you might simply want to display data on a form. In other cases, you might need to device a way to share information with another company.

No matter what you do with data, there are certain fundamental concepts that you should understand about the data approach in ADO.NET. You might never need to know some of the details of data handling- for example, you might never need to directly edit an XML file containing data- but it is very useful to understand the data architecture in ADO.NET, what the major data components are, and how the pieces fit together.

This introduction presents a high-level over view of these most important concepts. The topic deliberately skips over many details- for example, there is much more to data sets than what is mentioned here- in favour of simply introducing you to ideas behind the data integration in ADO.NET.

ADO.Net does not continuously live connections. In traditional client/server applications, components establish a connection to a data base and kept it open while the application is running. For a variety of reasons, this approach is impractical in many applications.

Open database connections take up valuable system resources. In most cases, databases can maintain only a small number of concurrent connections. The overhead of maintaining these connections detracts from overall application performance.

Similarly, applications that require an open database connection are extremely difficult to scale up. An application that does not scale up well might perform acceptable with four users but will likely not do so with hundreds. ASP.NET Web applications in particular need to be easily scalable, because traffic to a website can go up by orders of magnitude in a short period.

A model based on always connected data can make a difficult and impractical to exchange data across application and organizational boundaries using a connected architecture. If two components need to share the same data, both have to be connected, and a way must be devised for the components to pass data back and forth.

For all the reasons, data accessed with ADO.NET is designed around an architecture that uses connections sparingly. Applications are connected to the database only long enough to fetch or update the data. Because the database is not holding onto connections that are largely idle, it can service many more users.

4.3 Overview of C#.Net

- C# is a simple, modern, object oriented, and type –safe programming language derived from C and C++.
 - It will immediately be familiar to C and C++ programmers.
 - C# aims to combine the high productivity of visual basic and the raw power of C++
- Visual C#.NET is Microsoft’s C# development tool.
- It includes an interactive development environment, visual designers for building windows and web applications, a compiler and a debugger.
- Visual C#.NET is part of a suite of products, called Visual Studio .NET, that also includes Visual Basics .NET, Visual C++.NET and the Jscript scripting language.
- The .NET frame work defines a “Common Language Specification” (CLS), a sort of lingua franca that ensures seamless interoperability between CLS-complaint languages and class libraries.
- For C# developers this means even though C# is a new language, it has complete access to the same rich class libraries that are used by seasoned tools such as Visual Basic.NET and Visual C++.NET.

4.4 JAVA

JavaScript is an interpreter, client-side, event-based, object oriented scripting language that you can use to add dynamic interactivity to your web pages.

JavaScript scripts are written in plain text, like HTML, XML, Java, PHP and just about any other modern computer code. In this code, we will use Windows Note Pad to create and edit our JavaScript code, but there are a large number of alternatives available.

Note Pad is chosen to demonstrate JavaScript’s immediacy and simplicity.

You can use JavaScript to achieve any of the following:

- Create special effects with images that give the impression that a button is either highlighted or depressed whenever the mouse pointer is hovered over it.
- Validate information that users enter into your web forms
- Open pages in new windows, and customise the appearance of those new windows.
- Detect the capabilities of the user's browser and alter your page's content appropriately.
- Create custom pages "on the fly" without the need for a server-side language like PHP.

JavaScript is not Java, though if you come from a Java background, you will notice that both languages look similar when written. Java is a full featured and comprehensive programming language similar to C or C++, and although JavaScript can interact with Java web applications, the two should not be confused.

Different web browsers will run your JavaScript in different, sometimes incompatible ways. In order to work around this, it is often necessary to use JavaScript itself to detect the capabilities of the browser in which it finds itself, and alter its operation depending on the result.

To revisit the original definition in this chapter, note the following points:

- **Interpreted** refers to the fact that JavaScript code is executed (acted on) as it is loaded into the browser. This is a change of pace from compiled languages like Java, which check your program thoroughly before running a single line of code, and can have many implications that can catch you out if you are from a non-interpreted programming background.
- **Client-side** has been defined already in the previous chapter.
- **Event-based** refers to JavaScript's ability to run certain bits of code only when a specified event occurs. An event could be the page being loaded, a form being submitted, a link being clicked, or an image being pointed at by a mouse pointer.
- **Object-oriented signals** that JavaScript's power to exert control over an HTML page is based on manipulating objects within that page.
- If you are familiar with object-oriented programming, you will be aware of some of the power that this can bring to the coding environment.

4.5 DATABASE

About Microsoft SQL Server 2008

Microsoft SQL server is a Structured Query Language (SQL) base, client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL server.

A database is similar to a data file in that it is storage place for data. Like a data file, a database does not present information directly to a user, the user runs an application that accesses data from the database and presents it to the users in an untreatable format. A database typically ha two components: the files holding the physical database access data.

The DBMS is responsible for enforcing database structure, including:

- Maintaining the relationships between data in the database.
- Ensuring that data is stored correctly, and the rules that defining data relationships are not violated.
- Recovering all data to a point of known consistency in case of system failures.

Relational Database

There are different ways to organize data in a database but relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data. In a relational database is collected into tables called relations in relation theory.

When organizing data into tables, you can usually find many different ways to define tables. Relational database theory defines a process, normalization, which ensures that the set of tables you define will organize our data effectively.

Client/Server

In Client/Server system the server is a relatively large computer in a central location that manages a resource used by many people. When individuals need to use the resource, they connect over the network from their computers, or clients, to the server.

Examples of servers are: In Client/Server database architecture, the database files and DBMS software resides on a server. A communications component is provided so applications can run on separate clients and communicate to the database server over a network. The SQL server communication component also allows communication between an application running on the server and SQL server.

Server applications are usually capable of working with several clients at the same time. SQL server can work with thousands of client applications simultaneously. The server has features to prevent the logical problems that occur if a user.

While SQL server is design to work as a server in a Client/Server network, it is also capable of working as a stand-alone database directly on the client. The scalability and ease of use features of SQL server allows it to work efficiently on a client without consuming too many resource.

Structured Query Language (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational database; the most common is SQL. Both the American national standards institute (ANSI) and the International Standards Organization (ISO) has defined standards for SQL.

5. CODING

Registration Form

```
<% @ Page Title="" Language="C#" MasterPageFile="~/MasterPage.master"
AutoEventWireup="true" CodeFile="reg.aspx.cs" Inherits="reg" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
  <style type="text/css">
    .style2
    {
      width: 57%;
    }
  </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
  <table class="style2" align="center">
    <tr>
      <td colspan="2" style="text-align: center">
        <strong>User Registration Form</strong></td>
      </tr>
    <tr>
      <td>
        Name</td>
      <td>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      </td>
    </tr>
    <tr>
      <td>
        Email</td>
      <td>
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
      </td>
    </tr>
    <tr>
      <td>
        Mobile</td>
      <td>
        <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
      </td>
    </tr>
    <tr>
      <td>
        Gender</td>
      <td>
        <asp:RadioButtonList ID="RadioButtonList1" runat="server"
RepeatDirection="Horizontal">
          <asp:ListItem>Male</asp:ListItem>
        </td>
    </tr>
  </table>
</asp:Content>
```

```

        <asp:ListItem>Female</asp:ListItem>
    </asp:RadioButtonList>
</td>
</tr>
<tr>
<td>
    UserId</td>
<td>
    <asp:TextBox ID="TextBox4" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>
    Password</td>
<td>
    <asp:TextBox ID="TextBox5" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>
    <asp:Label ID="Label1" runat="server" style="font-weight: 700"></asp:Label>
</td>
<td>
    <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
        Text="Register" />
</td>
</tr>
</table><br /><br /><br />
</asp:Content>

```

MasterPage

```
<% @ Master Language="C#" AutoEventWireup="true" CodeFile="MasterPage.master.cs"
Inherits="MasterPage" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
  <asp:ContentPlaceHolder id="head" runat="server">
  </asp:ContentPlaceHolder>
  <style type="text/css">
    .style1
    {
      width: 100%;
      border: 1px solid #0000FF;
      background-color: #FFFFFF;
    }
  </style>
</head>
<body bgcolor="skyblue">
  <form id="form1" runat="server">
    <table class="style1">
      <tr>
        <td>
          <asp:Image ID="Image1" runat="server" Height="312px" Width="883px" />
        </td>
      </tr>
      <tr>
        <td>
          <asp:Menu ID="Menu1" runat="server" Orientation="Horizontal"
            style="font-weight: 700">
            <DynamicMenuStyle HorizontalPadding="60px" VerticalPadding="50px"
              Width="1000px" />
            <Items>
              <asp:MenuItem Text="Home" Value="Home"
                NavigateUrl="~/Admin/home.aspx"></asp:MenuItem>
              <asp:MenuItem Text="Add Products" Value="About Us"
                NavigateUrl="~/Admin/products.aspx"></asp:MenuItem>
              <asp:MenuItem Text="View Users" Value="Login"
                NavigateUrl="~/Admin/viewusers.aspx"></asp:MenuItem>
              <asp:MenuItem Text="View Reports" Value="Contact Us"
                NavigateUrl="~/Admin/reports.aspx"></asp:MenuItem>
              <asp:MenuItem Text="View Orders" Value="View Orders"
                NavigateUrl="~/Admin/vieworders.aspx"></asp:MenuItem>
              <asp:MenuItem NavigateUrl="~/Default.aspx" Text="Logout"
                Value="Logout">
            </asp:MenuItem>
```

```
</Items>
<StaticMenuItemStyle Width="100px" />
<StaticMenuStyle HorizontalPadding="60px" Width="800px" />
</asp:Menu>
</td>
</tr>
<tr>
<td style="margin-left: 80px">
<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</td>
</tr>
<tr>
<td style="text-align: center; margin-left: 80px">
<strong>Copy Rights reserved</strong></td>
</tr>
</table>
</form>
</body>
</html>
```

6. SCREENSHOTS

Home page:

Online Shopping Portal

[Home](#) | [About Us](#) | [Contact Us](#)

Product	Price	Availabe Quantity
Nokia N8	25000.00	30
Nokia N8	25000.00	30
Nokia N8	25000.00	30
Parker Pen	200.00	100

Done | One active download (3 minutes, 58 seconds remaining) | [76% of 1 file - Downl...]

Registration page:

Online Shopping Portal

[Login](#) | [Register](#)

[Home](#)
[About Us](#)
[Contact Us](#)

Login Information

User Name:

Password:

Confirm Password:

Personal Information

First Name:

Last Name:

Email ID:

Date of Birth:

Phone:

Security Quesiton:

Security Answer:

Login Page:

The login page features four social media login buttons: Facebook (blue), Twitter (light blue), Google+ (red), and Yahoo (purple). Below these is a section for other login methods with icons for Google, LinkedIn, Facebook, and Yahoo, accompanied by the text "Secured by LoginRadius" and a privacy statement: "We will never post anything without permission". To the right, the "Login with Email" section includes input fields for "Email" and "Password", and a green "Login" button.

Online Shopping Portal

Welcome ankigupt10 | [Shopping Cart](#) | [Change Password](#) | [Update Profile](#) | [Log out](#)

- [Home](#)
- [About Us](#)
- [Contact Us](#)
- [Track Order](#)

Item	Quantity	Price	Tasks
Nokia N8	2	50000	<input type="button" value="Delete"/>

Total Amount 50000

[Continue Shopping](#)

7. CONCLUSION

We have presented Co AI, our approach to facilitate the incremental consolidation of data-intensive Flows. Co AI starts from data Flows that satisfy single information requirements. Iteratively, Co AI Identifies different possibilities for integrating new data Flows into the existing multi- Flow, focusing on the maximal data Flow reuse. Finally, Co AI suggests a unified data Flow design evaluating it with the user-specified cost model. We have developed a prototype that implements the complete functionality of Co AI. We used it to evaluate the efficiency, scalability, and the quality of the output solutions of our approach, reporting the improvement of the overall execution time as well as other benefits of integrated multi- Flows. The final goal of our overall work is to provide an end-to-end platform for self-managing the complete lifecycle of BI solutions, from information requirements to deployment and execution of data-intensive Flow

8. BIBLIOGRAPHY

The following books are referred during the analysis and execution phase of the project

1. M. Lenzerini, "Data integration: A theoretical perspective," in PODS, 2002, pp. 233–246.
2. D. Caruso, "Bringing Agility to Business Intelligence," February 2011, Information Management, <http://www.information-management.com/infodirect/2009191/business-intelligence-metadata-analytics-ETL-data-management-10019747-1.html>.
3. R. Hughes, Agile Data Warehousing: Delivering world-class business intelligence systems using Scrum and XP. IUiverse, 2008.
4. Y. Chen, S. Alspaugh, and R. Katz, "Interactive analytical processing in big data systems: A cross-industry study of map reduce workloads," Proceedings of the VLDB Endowment, vol. 5, no. 12, pp. 1802–1813, 2012.

WEBLINKS:

1. www.c#tutorial.com - covering all the most important C# concepts. This tutorial is primarily for new users.
2. www.DotnetSpider.com - what is the .NET all about? For sample projects.