Dot Net Freshers' Training

Jul – 2016

Important: Clause Concerning Security

Prepared by	***Sushant Banerjee(sushantba@cybage.com)***
Reviewed by	Pushkar Kulkarni
For more information, contact ***Priyanka Shinde (Ext: 7222)***	

Training Agenda

Day	Topic	Trainer	Date	Time	Topic	Trainer	Time
1	Introductory Session	Pushkar K	18-Jul-16	11.00am-12.00pm			
2	Introduction to OOP, Data Structure and RDBMS	Sushant B	19-Jul-16	10.00am-01.00pm			
3	DDL and DML Statements	Sushant B	20-Jul-16	10.00am-01.00pm			
4	Soft Skill		21-Jul-16	11.00am-01.00pm	ALM	ALM Team	02.30pm-04.30pm
5	Soft Skill		22-Jul-16	02.30pm-04.30pm	ALM	ALM Team	02.30pm-04.30pm
6	Soft Skill		25-Jul-16	10.00am-01.00pm	ALM	ALM Team	02.30pm-04.30pm
7	SELECT Queries and Built-in-Functions	Sushant B	26-Jul-16	10.00am-01.00pm	ALM	ALM Team	02.30pm-04.30pm
8	WORKSHOP	**Mentors & Trainers**	27-Jul-16	10.00am-01.00pm	ALM	ALM Team	02.30pm-04.30pm
9	Views and Indexes	Sushant B	28-Jul-16	10.00am-01.00pm			
10	Stored Procedures and UDFs	Sushant B	29-Jul-16	10.00am-01.00pm			
11	Triggers and Transactions	Sushant B	1-Aug-16	10.00am-01.00pm			
	Feedback session by SME		2-Aug-16	02.00 pm-02.30 pm			
Test							
12	Introduction to C#	Sushant B	3-Aug-16	10.00am-01.00pm			
13	**WORKSHOP**	**Mentors & Trainers**	4-Aug-16	**02.30pm-04.30pm**			
14	OOP Concepts - Day1	Sushant B	5-Aug-16	10.00am-01.00pm			
15	OOP Concepts - Day2	Sushant B	8-Aug-16	10.00am-01.00pm			

16	**WORKSHOP**	**Mentors & Trainers**	9-Aug-16	**02.30pm-04.30pm**			
17	Exception Handling and Debugging	Sushant B	10-Aug-16	10.00am-01.00pm			
18	Collections and Generics	Sushant B	11-Aug-16	10.00am-01.00pm			
19	Delegates and Events	Sushant B	12-Aug-16	10.00am-01.00pm			
20	Threading and File Handling	Sushant B	16-Aug-16	10.00am-01.00pm			
21	Attributes and Serialization	Sushant B	17-Aug-16	10.00am-01.00pm			
22	Overview of .NET Framework	Sushant B	18-Aug-16	10.00am-01.00pm			
	Feedback session by SME		18-Aug-16	02.00 pm-02.30 pm			
Test							
23	ADO.NET - Day1	Sushant B	19-Aug-16	10.00am-01.00pm			
24	ADO.NET - Day2	Sushant B	22-Aug-16	10.00am-01.00pm			
25	**WORKSHOP**	**Mentors & Trainers**	23-Aug-16	**02.30pm-04.30pm**			
26	ADO.NET Entity Framework	Sushant B	24-Aug-16	10.00am-01.00pm			
27	LINQ to Objects	Sushant B	25-Aug-16	10.00am-01.00pm			
Test							
28	Introduction to MVC	Sushant B	26-Aug-16	10.00am-01.00pm			
29	Controllers	Sushant B	29-Aug-16	10.00am-01.00pm			
30	Views	Sushant B	30-Aug-16	10.00am-01.00pm			
31	Models with EF	Sushant B	31-Aug-16	10.00am-01.00pm			
32	Model Validations	Sushant B	1-Sep-16	10.00am-01.00pm			
33	**WORKSHOP**	**Mentors & Trainers**	2-Sep-16	**02.30pm-04.30pm**			

34	Security and Identity	Sushant B	5-Sep-16	10.00am-01.00pm			
35	Overview of MVC Advanced	Sushant B	6-Sep-16	10.00am-01.00pm			
36	Introduction to Web API	Sushant B	7-Sep-16	10.00am-01.00pm			
37	**WORKSHOP**	**Mentors & Trainers**	8-Sep-16	**02.30pm-04.30pm**			
38	Unit Testing	Sushant B	9-Sep-16	10.00am-01.00pm			
	Feedback session by SME		9-Sep-16	02.00 pm-02.30 pm			
Test							
39	HTTP	Sushant B	12-Sep-16	10.00am-12.00pm			
40	HTML 5	TBD	13-Sep-16	10.00am-12.00pm			
41	HTML 5	TBD	14-Sep-16	10.00am-12.00pm			
42	CSS 3	TBD	15-Sep-16	10.00am-12.00pm			
43	CSS 3	TBD	16-Sep-16	10.00am-12.00pm			
44	CSS 3	TBD	19-Sep-16	10.00am-12.00pm			
45	SASS	TBD	20-Sep-16	10.00am-12.00pm			
46	SASS	TBD	21-Sep-16	10.00am-12.00pm			
47	JavaScript	Sushant B	22-Sep-16	10.00am-12.00pm	JavaScript	Sushant B	02.30pm-04.30pm
48	JavaScript	Sushant B	23-Sep-16	10.00am-12.00pm	JavaScript	Sushant B	02.30pm-04.30pm
49	JavaScript	Sushant B	26-Sep-16	10.00am-12.00pm			
50	Angular JS	TBD	27-Sep-16	10.00am-12.00pm	Angular JS	TBD	02.30pm-04.30pm
51	Angular JS	TBD	28-Sep-16	10.00am-12.00pm	Angular JS	TBD	02.30pm-04.30pm
52	Angular JS	TBD	29-Sep-16	10.00am-12.00pm			
Test							

	Closure Meeting	TBD	TBD	TBD			

Contents

1. SQL Server 2012

Agenda

- RDBMS and SQL Server 2012
- DDL and DML Statements
- SELECT Queries and Built-in Functions
- Views and Indexes
- Stored Procedures and UDFs
- Triggers and Transactions

Assignments

DDL Statements

1. Create a database name "BookStoreDB" and create following tables in the database with the below mentioned columns using SQL Server.

    I. Author – AuthorId, AuthorName, DateOfBirth, State, City, Phone.
    II. Publisher – PublisherId, PublisherName, DateOfBirth, State, City, Phone.
    III. Category – CategoryId, CategoryName, Description.
    IV. Book – BookId, Title, Description, Price, ISBN, PublicationDate, Image.
    V. Order – OrderId, Date, Quantity, UnitPrice, ShippingAddress.

2. Modify the BookStoreDB and add all the required keys. Establish relationships among tables and apply following business rules.

    I. A Book can have multiple authors.
    II. An Author can write more than one book.
    III. A Book belongs to only one category.
    IV. A Book can be published by only one publishing house.
    V. An order can be placed for a single book but multiple quantities.

SELECT Statements

3. Write the appropriate SQL queries against BookStoreDB database to support following operations:

a. Get All the books written by specific author
b. Get all the books written by specific author and published by specific publisher belonging to “Technical” book Category
c. Get total books published by each publisher.
d. Get all the books for which the orders are placed.

Stored Procedures

4. Write the following stored procedure using SQL Server in BookStoreDB database to support following operations:

    a. Get All the books written by specific author
    b. Get all the books written by specific author and published by specific publisher belonging to “Technical” book Category
    c. Get total books published by each publisher.
    d. Insert a particular book
    e. Update a particular book by id
    f. Delete a particular book by id

Triggers

5. Let’s assume that we have a table name “Book_History” table. If a particular book is deleted from the “Book” table, an entry with same book records to “Book_History” table must take place. Automate this process using trigger.

6. The “Book” table got an attribute “Price”. Let’s assume that we have a business requirement where we must ensure that the “Price” should not be less than 1. If any insert or update statement tries to make the “Price” less than 1, the SQL Server must terminate such insert or update statements. Write an appropriate trigger to implement the business requirement.

7. Create a trigger on the table “Order” and add the following functionalities. When a new order is placed, it should check whether the required quantity is available in the “Book” table. If not, it should show appropriate message and the insert statement to “Order” table should be terminated. If the quantity in book table is sufficient, it should deduct the quantity ordered from the quantity in hand in the book table and update the quantity.

2. C# 5 and Visual Studio 2013

Agenda

- C# Language Fundamentals
- OOP Concepts
- Overview of .NET Framework
- Exception Handling and Debugging
- Using Visual Studio
- Delegates and Events
- Collections and Generics
- Threading
- File Handling
- Attributes
- Serialization

Assignments

Classes and Objects

Create a console application name "BookStoreApp" using C#. Write the following classes with properties. Use Automatic Properties and use suitable access modifiers for the class and properties. All the properties should have an appropriate data types.
Author – AuthorId, AuthorName, DateOfBirth, State, City, Phone.
Publisher – PublisherId, PublisherName, DateOfBirth, State, City, Phone.
Category – CategoryId, CategoryName, Description.
Book – BookId, Category, Title, Author, Publisher, Description, Price, ISBN, PublicationDate.

Collections and Generics

Use the "BookStoreApp" and create your static data store using Generic Collections. Create multiple Book objects and store using collection. Write methods to support all the operations against the collection based on the below menu. The program should

display a menu as following pattern to the user. According to user's choice the program should perform a specific operation and display appropriate output.

==========
Main Menu
==========
1. Adding new books
2. Displaying all books
3. Displaying a book by BookId
4. Updating a book by BookId
5. Deleting a book by BookId
6. Exit
Enter Your Choice (any number from 1 to 6)

Exception Handling

Modify the above BookStoreApp and implement appropriate exception handling logic. The program must handle specific exceptions using responsible handler. The program should also able to handle any exception in case specific handler is not present.

Data Validation

Modify the above BookStoreApp and use appropriate data validation to validate each and every user input. In case of invalid user input, the program should display appropriate message to help user identify his/her mistake and enter valid value.

Serialization

Modify the above BookStoreApp and write appropriate methods for the following requirements:

a. A method should create a file in the hard disk to serialize Book objects and save the objects in XML format.
b. A method should de-serialize the Book objects saved as XML data and consume in the application.

3. Database Programming in C#

Agenda

- ADO.NET
- Entity Framework
- LINQ

Assignments

ADO.NET

Use the "BookStoreApp" and the "BookStoreDB" you have already created during previous assignments. Create a Data Access Layer for the application. You should call the stored procedure wherever required that you have already created during SQL Server assignments. The DAL should expose appropriate methods for the following operations:

1. Connect to the BookStoreDB
2. Retrieve all books and display in the application
3. Retrieve a book by BookId
4. Add a new book
5. Update an existing book
6. Delete an existing book
7. Get All the books written by specific author
8. Get all the books written by specific author and published by specific publisher belonging to "Technical" book Category
9. Get total books published by each company
10. Write following queries (Using SQL Statement and Stored Procedure) and call them through the code.
    1. Retrieve all the books which published in 2013
    2. Retrieve all the books which are belonging to category "History & Politics"
    3. Retrieve all the books which are written by a given author and belonging to category "Literature & Fiction"

Note: You must use exception handling code wherever required.

Use the “BookStoreApp” and the “BookStoreDB” you have already created during previous assignments. Replace the Data Access Layer for the application using Entity Framework and LINQ to complete this assignment and the assignment is same as above.

EF Assignments

1) Create following entities and attributes :
   a) Author – AuthorId, AuthorName, DateOfBirth, State, City, Phone.
   b) Publisher – PublisherId, PublisherName, DateOfBirth, State, City, Phone.
   c) Category – CategoryId, CategoryName, Description.
   d) Book – BookId, CategoryId, Title, AuthorId, PublisherId, Description, Price, ISBN, PublicationDate, Image.
2) Connect to the BookStoreDB
3) Retrieve all books and display in the application
4) Retrieve a book by BookId
5) Add a new book
6) Update an existing book
7) Delete an existing book

LINQ Assignments

8) Write following queries (Using SQL Statement and Stored Procedure) and call them through the code.
   a) Retrieve all the books which published in 2013
   b) Retrieve all the books which are belonging to category “Technical”
   c) Retrieve all the books which are written by a given author and belonging to category “Literature & Fiction”

Note: You must use exception handling code wherever required.

4. ASP.NET MVC 5

Agenda

- Introduction to ASP.NET MVC
- Controllers
- Views
- Models
- Security and Identity
- ASP.NET MVC Advanced
- Introduction to Web API

Assignments

Develop an application name “BookStoreApp” using ASP.NET MVC and Entity Framework. The tools to develop this application should be VS 2013 and use LocalDB (SQL Server 2012) to attach your database to the application. The application should meet the following requirements. The application should support admin and user roles.

An administrator role can perform the following tasks.

9) All the functionality for the Administrator should be in separate Area.
10) The application should allow an administrator to add new books, edit and delete existing books.
11) The view used for create and edit any entity should be same.
12) The application should have following validations. The validations should be on client side as well as server side.
    a) ISBN of the book should be valid.
    b) The date of publication for the book should not be the future date. The idea is to add custom validations.
13) The application should expose Web APIs which could be used by third party applications to get book details.

Phase-II

A user role can perform the following tasks.

14) The application should allow users to browse all the books by category.
15) The application should have functionality to search the books by title, author, and publisher. This should be done using Ajax such that only the search results section gets refreshed.
16) Users can view any single book details such as book name, category name, publisher name, author name and price.
17) While viewing book details users can add the book to their shopping cart.
18) The shopping cart should allow the user to review their cart and user can remove any item from their cart.
19) The application should ask users to login as soon as the users finally want to place an order (Use Membership).
20) If the user is not a registered user, the application should allow user to register (Use Membership).
21) After the user logged in or completed the registration process, the application should ask the shipping information from the user.
22) Finally when the user submits the order, the application should display confirmation message to the user.
23) The user should be able to add review for the book. These reviews should be displayed on book details view. Submitting review should be an Ajax action.

Note: You must use exception handling code wherever required.

5. Web Technologies

Agenda

- HTTP
- HTML
- CSS
- JavaScript
- AngularJS

Assignments
The “BookStoreApp” developed using ASP.NET MVC in the previous assignments should expose all the books through a Web API. The Web API should support CRUD operation against the books. Develop an AngularJS application name “BookStoreApp” that should be working as a web client and able to call API functionalities.

Assignments (Mini Project)

Online Bus Ticket Reservation System

Develop the application using the following technologies and tools:

- ASP.NET MVC 5
- Entity Framework
- VS 2013 and SQL Server 2012 (use LocalDb to attach .mdf to App_Data folder)

The application should meet the following requirements:

1. **System Requirements**
   - I. The system should allow anonymous/authenticated users to search bus.
   - II. The system should allow search bus using categories such as All, AC/Non-AC/Sleeper/Semi-Sleeper.
   - III. The system should ask user to set source and destination while searching bus.
   - IV. The system should ask user to set date of journey while searching bus.
   - V. The system should ask user to set return date while searching bus in case user is interested in return journey.
   - VI. The system should take user to login page when user wants to book a ticket.
   - VII. The system should allow user to register and take the user to registration page in case the user is not already registered.
   - VIII. The system should redirect user to booking page/search results after successful login/register.
   - IX. The system should calculate and update available seats after each booking. The system should allow new booking against available seats only.
   - X. The system should record and display list of persons travelling for each booking.
   - XI. System should calculate fare using the below rate card for tickets.

categories	Above 16	Below 16	Below 5
AC	2 * distance	1.75 * distance	free
Non-AC	1.2 * distance	1.05 * distance	Free
Sleeper	1.5 * distance	1.25 * distance	Free
Semi-Sleeper	1.35 * distance	1.25 * distance	Free

2. **Data Validation Requirements**
    I. The system should validate data using client side and server side both.
    II. The system should validate all the required fields and display appropriate error messages.
    III. The system should validate user's date of birth at the time of registration, which should not be more than 100 years and less than 18 years.
    IV. The system should validate source and destination city and display appropriate error message in case both are same.
    V. The system should validate date of journey and display appropriate error message in case the date is in the past.
    VI. The system should validate return date of journey and display appropriate error message in case return date is before the date of journey.
    VII. System should not allow to book tickets for the same seats.
    VIII. System should not allow a single user to book more than 5 tickets.

3. The application should support two roles – User and Admin.
4. **User Activities**
    a. The user can search for the bus/seat by providing source, destination and date of journey. Time of journey and type of bus should be optional.
    b. The user can login or register with the system.
    c. The user can book/cancel tickets.
    d. Only registered user can complete the booking process.
    e. While booking tickets, user should be able to view seating arrangement in the bus.
    f. The user should be able to see list of all bookings he/she has done so far.
    g. The user should be able to view ticket as a print view.
    h. For booking tickets of round trip journey a separate view should be available to user for booking tickets.
5. **Admin Activities**
    a. The admin can manage the details of bus.
    b. The admin can manage the details of customer.
    c. Admin manages the allowed bus routes in the system.

Good to have features

1. Discrete seat booking should be allowed.
2. Cancellation
3. Storing and retrieving user details in order page.
4. Allow user to visualize sitting arrangement and selecting seats of their choice just by checking check boxes.

5. System should not allow removing buses or routes if any booking exist on that bus and the bus is using the same route.
6. If any bus is cancelled or postponed and the passengers booked tickets for that trip should be getting appropriate message as intimation in the system after they log in to the system.
7. Write a single page application (SPA) that should allow the user to search a bus and do the bookings as well.

Declaration by the Participant

My Understanding

Name:
Employee ID:
DOJ:

Sr. No	Topic	Theoretical		Practical	
		Yes	No	Yes	No
1	VS 2008 IDE, C# Introduction				
2	Class and OOPs Concepts				
3	Introduction to .Net Framework				
4	Database Programming in C#				
5	Interface, Delegation, Events				
6	Advance Programming				
7	VS 2008 IDE, ASP.NET Introduction				
8	User Interface with ASP.Net				
9	State Management				
10	Managing Data in ASP.NET				
11	Debugging and Tracing ASP.NET Applications				
12	Caching in ASP.NET				
13	Controls				
14	Handling Security in ASP.NET				
15	Web Services				

Signature of Participant

For Office Use:

Signature of Trainer **Signature of Evaluator** **Signature of Training Head**