

Lab#2 – AWS DynamoDB

Due Date: Midnight of Oct 8 (Friday)

Purpose: The purpose of this lab is to help you:

- Have comprehensive understanding of AWS DynamoDB
- Become familiar with IAM
- Be able to use AWS DynamoDB service programmatically

Instructions: Be sure to read the following general instructions carefully:

1. This lab should be completed individually by all students. Submit your solution **through the dropbox**, name your submission as **studentID(yourlastname)_Labnumber.zip**. e.g., 300123456(smith)_Lab#2.zip
2. Demonstrate your solution @Teams

Rubric

<u>Functionality</u>	<u>Marks</u>
User Login (do it in C# code) <ol style="list-style-type: none"> 1. Create a DynamoDB table to store user's credentials (user name & password) (2 marks) 2. Insert at least 3 login credentials (1 mark) 3. Implement login business logic (3 marks) 	6
DynomoDB table <ol style="list-style-type: none"> 1. Create DynamoDB table Bookshelf via AWS Management Console, and insert sample data to the table. The sample data must include at least three different users, and each user has at least 2 books on his/her bookshelf (3 mark) 2. Modeling the table Bookshelf in C# code (3 mark) 	6
List all books on user's bookshelf <ol style="list-style-type: none"> 1. View (2 marks) 2. Code (3 marks) 	5
Update DynamoDB table to reflect user's reading activities <ol style="list-style-type: none"> 1. View book content (2 marks) 2. Update the snapshot of reading activity when user clicks BookMark button <ol style="list-style-type: none"> a) Get the current page number (1 mark) b) Write bookmark time and bookmarked page# to DynamoDB table (2 marks) 3. Update the snapshot of reading activity when user closes the window (2 marks) 	7
Overall (code readability, app usability, etc.)	1

Question 1[25 marks]

Nowadays, there are many online eBook readers (<https://www.epubor.com/online-ebook-readers.html>) available. Basic function is to allow logged in user to read book on his/her bookshelf. When the user finishes reading, the snapshot of his/her reading activity will be recorded so that he/she can continue reading from where he/she left last time.

Normally I read eBook at bookshelf.vitalsource.com, currently I have 4 books on my bookshelf. I read the book **Building Microservices with .NET core** (shown in Figure 1) last time before I logged out. Today,

after I logged in and clicked “Continue Reading” button, the page (Shown in figure 2) which I finished last time was displayed, and I can continue from where I left.

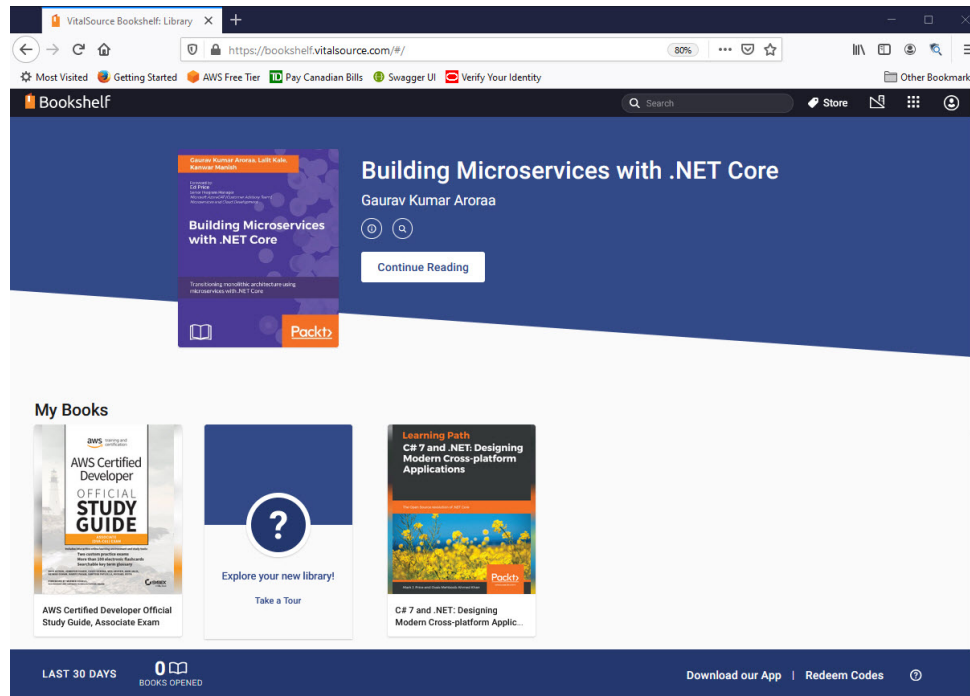


Figure 1

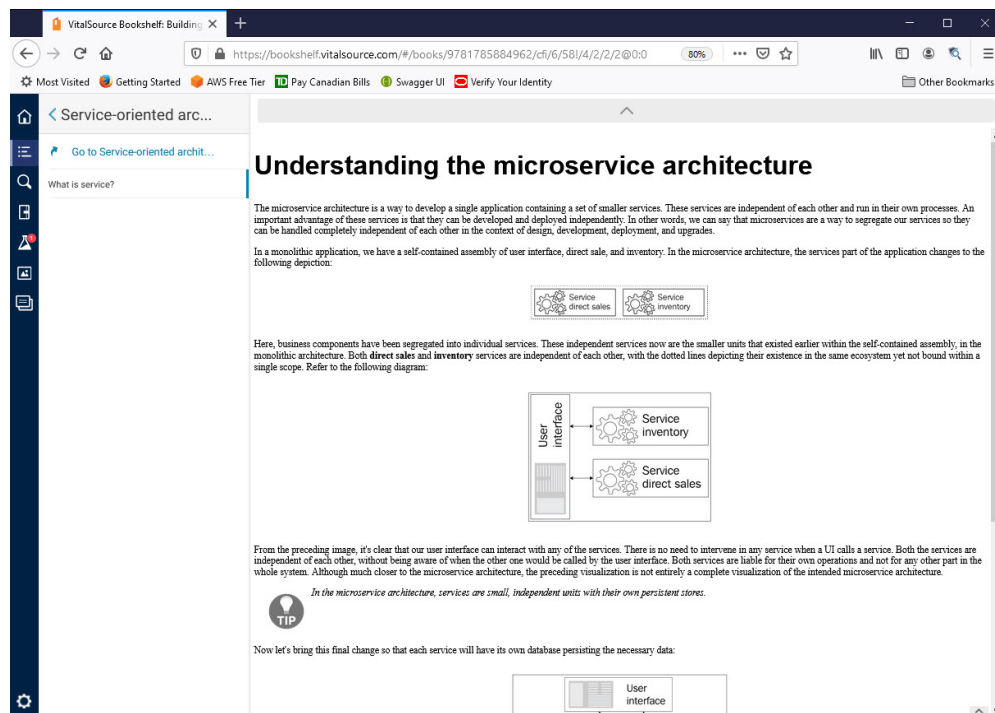
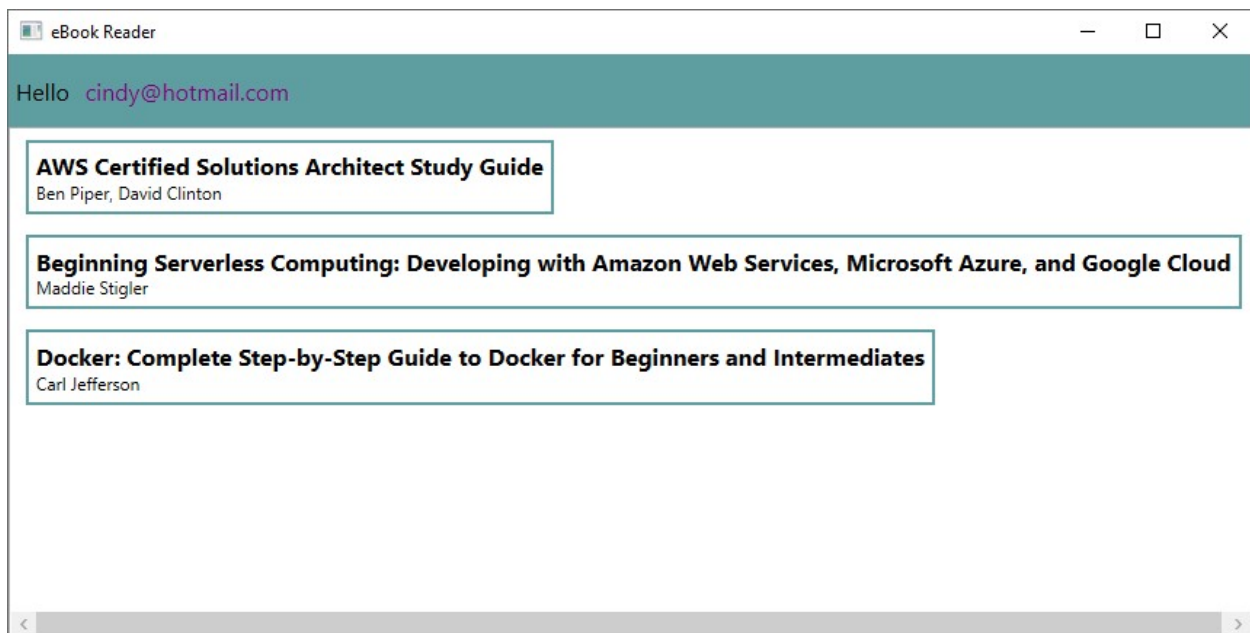
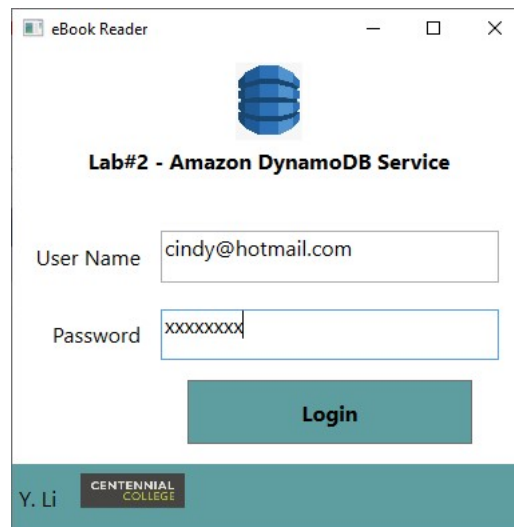


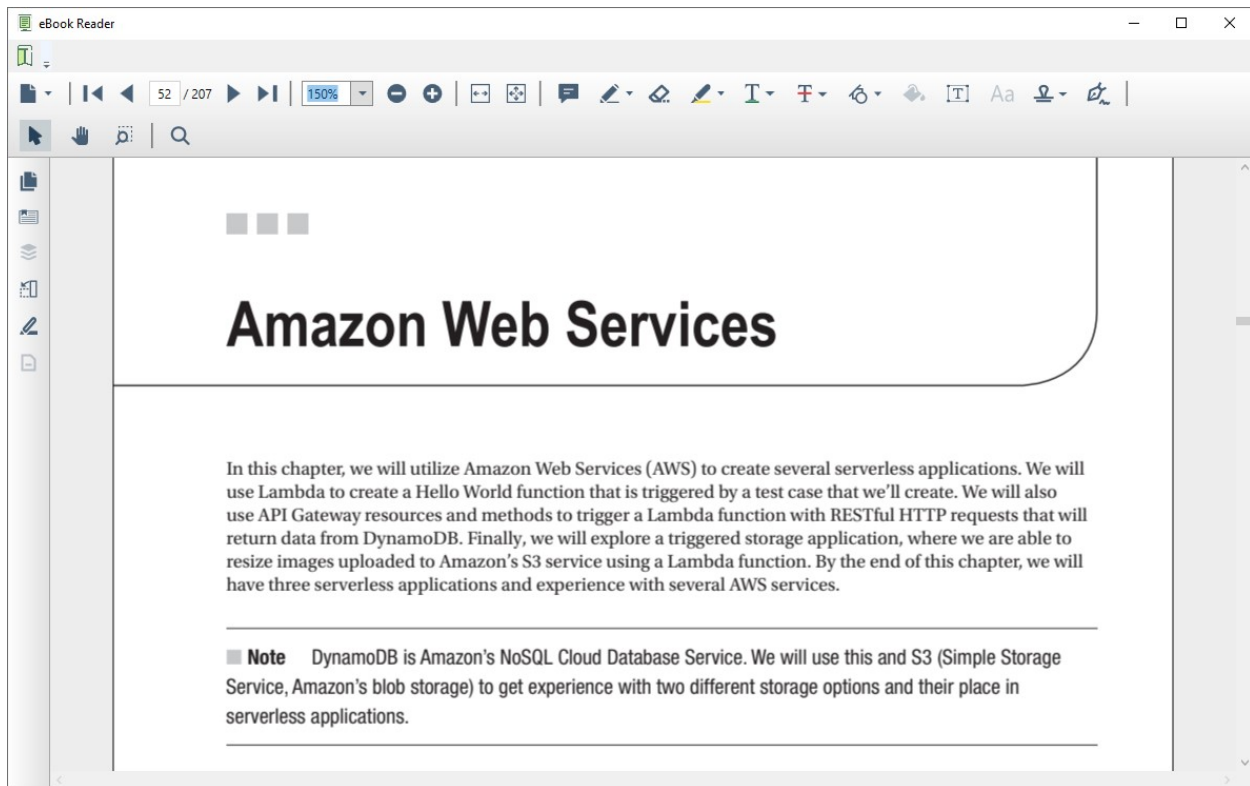
Figure 2

You are asked to implement a **WPF** app to mimic BookShelf.vitalsource.com. First, *programmatically* create DynamoDB table to store users' login credentials and insert some credentials (at least three users credentials needs to be added). Based on the existing user credentials, you can create DynamoDB table Bookshelf through **AWS Management Console**.

After a user successfully login, your app should

- 1) list all books on his/her bookshelf based on the bookmark time, and the most recently one is listed at the top
- 2) User starts to read a book by double clicking on the book
- 3) When user click on "Bookmark" button, or user closes the reading window, in both situations, the current page# should be bookmarked by updating the corresponding book item in the DynamoDB table





Hint:

Use `Syncfusion.PdfViewer` to view the pdf file.



Syncfusion.PdfViewer.WPF  by Syncfusion Inc., 72.1K downloads

Syncfusion PDF viewer for WPF is a 100 percentage managed .NET component that gives you the ability to view, review and print PDF documents from your WPF applications.

In your View (XAML file), you should include a `PdfViewerControl` element

```
<PdfViewer:PdfViewerControl x:Name="pdfViewer" ItemSource="{Binding DocumentStream}"/>
```

Please bear in mind that normally eReader does not allow user to download the book (pdf file), therefore, you need to use following code to load the book

```
using (AmazonS3Client s3Client = new AmazonS3Client(awsAccessKey, awsSecretKey, RegionEndpoint.USEast1))
{
    CurrentUserInfo.GetBucketInfo();
    GetObjectRequest request = new GetObjectRequest();
    request.BucketName = CurrentUserInfo.BucketName;
    request.Key = CurrentUserInfo.KeyName;
    GetObjectResponse response = s3Client.GetObjectAsync(request).Result;

    MemoryStream _documentStream = new MemoryStream();
    response.ResponseStream.CopyTo(_documentStream);

    Helper.selectedBookStream = _documentStream;
}
```