ALGOMA UNIVERSITY

Library Management

Submitted to: Anthony Pagnotta

Ancy Tom.

Student ID: 199693500

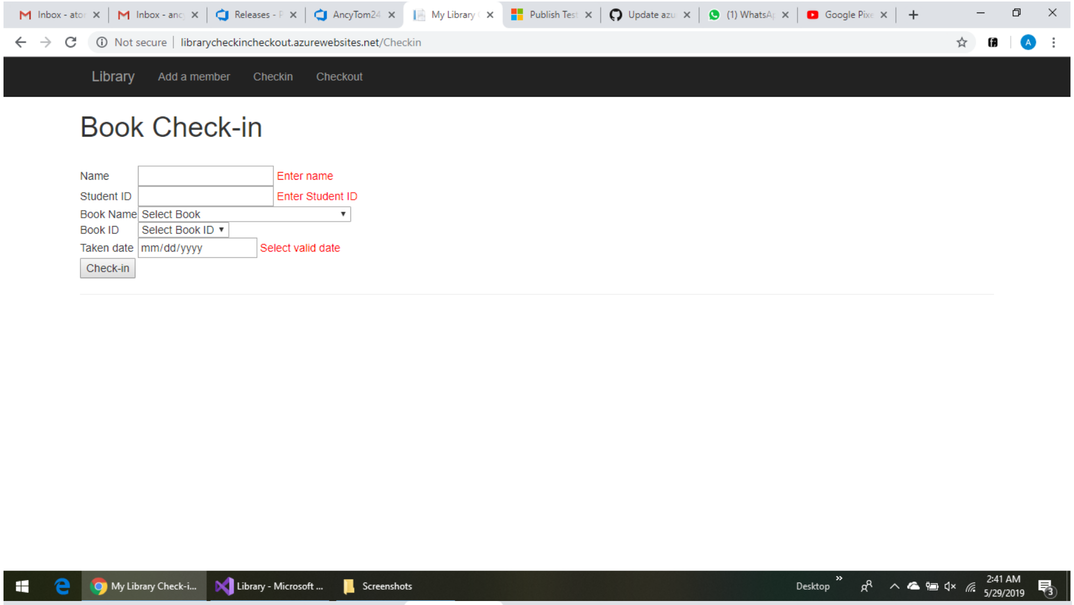
5/29/2019

Library Check-in/Check-out System

About the application (high level)

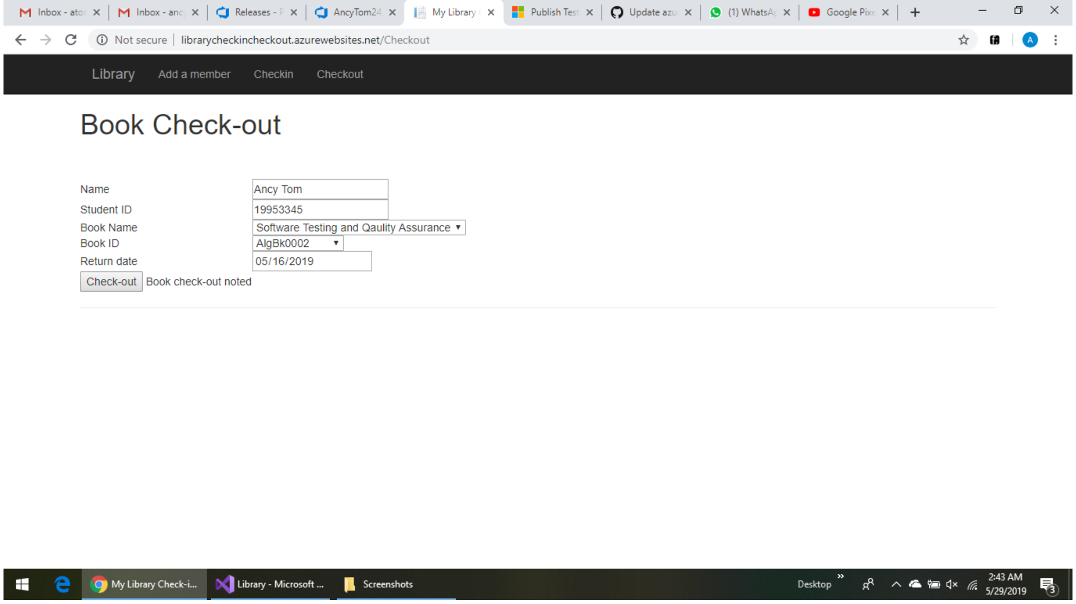
Library check-in/check-out is a simple web application that allows the user to add a member to library, check-in and check-out a book. There are 3 menus on the web application for these. Each has a form asking for various details and a button to perform the operations.

Application screenshots below for reference:



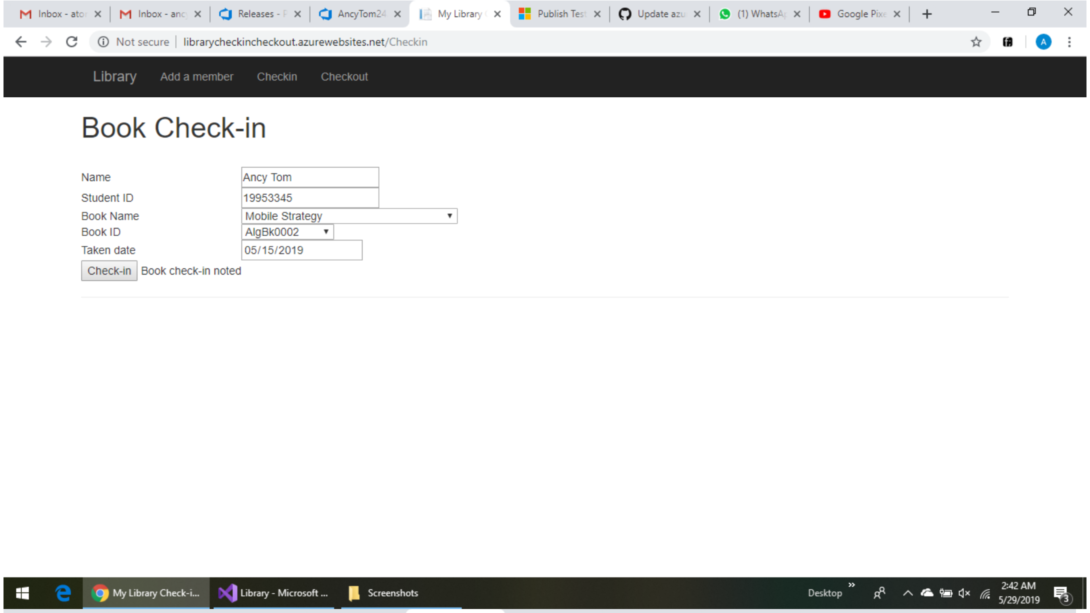
Book Check-in

Name [ ] Enter name
Student ID [ ] Enter Student ID
Book Name Select Book ▾
Book ID Select Book ID ▾
Taken date mm/dd/yyyy Select valid date
Check-in



Book Check-out

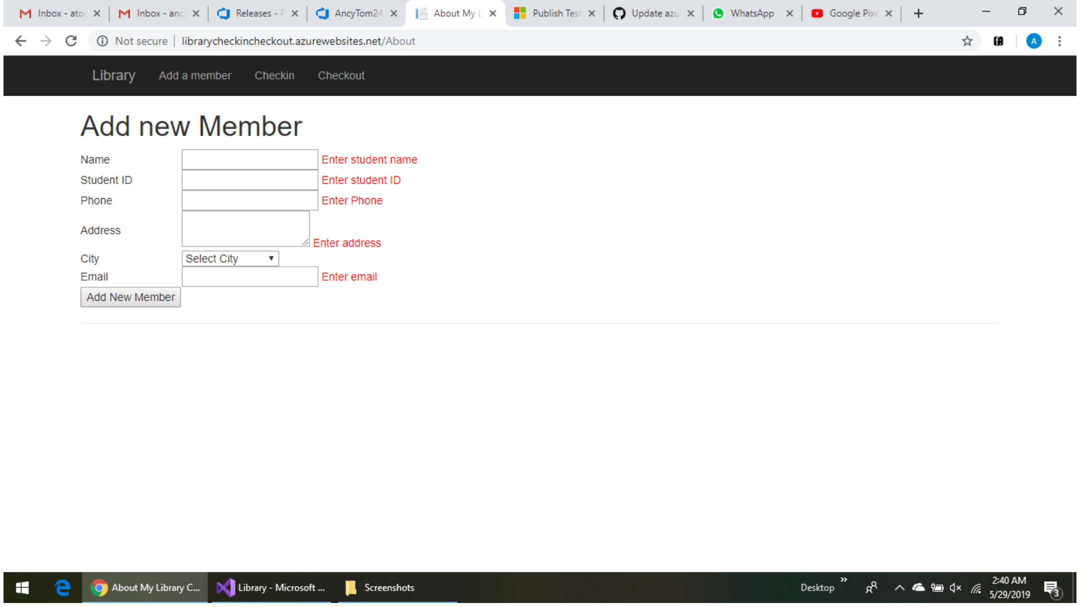
Name Ancy Tom
Student ID 19953345
Book Name Software Testing and Qaulity Assurance ▾
Book ID AlgBk0002 ▾
Return date 05/16/2019
Check-out Book check-out noted



Book Check-in

Name	Ancy Tom
Student ID	19953345
Book Name	Mobile Strategy ▾
Book ID	AlgBk0002 ▾
Taken date	05/15/2019

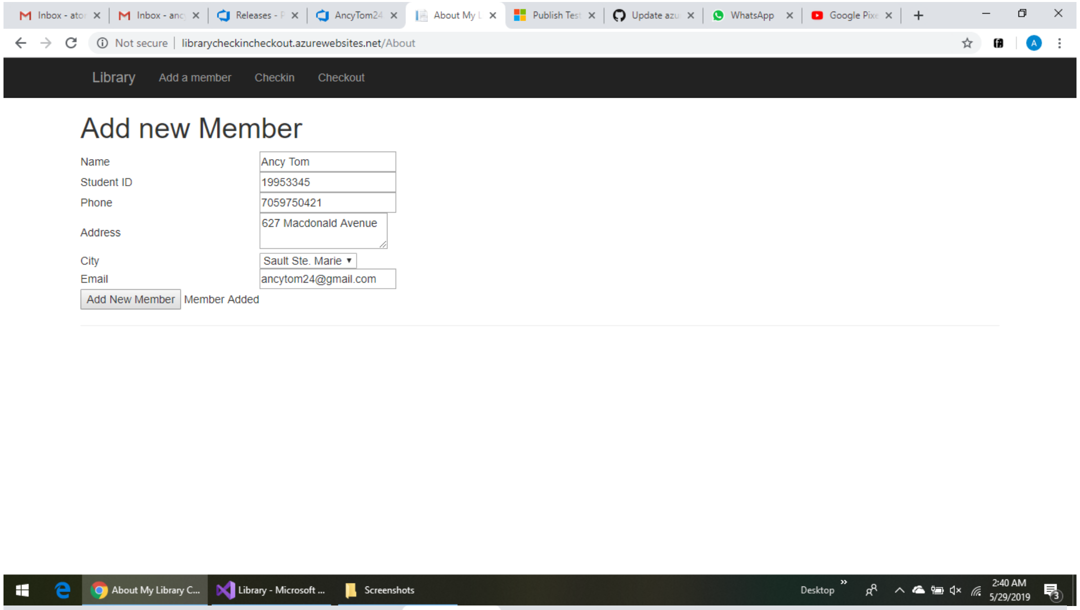
Check-in Book check-in noted



Add new Member

Name		Enter student name
Student ID		Enter student ID
Phone		Enter Phone
Address		Enter address
City	Select City ▾	
Email		Enter email

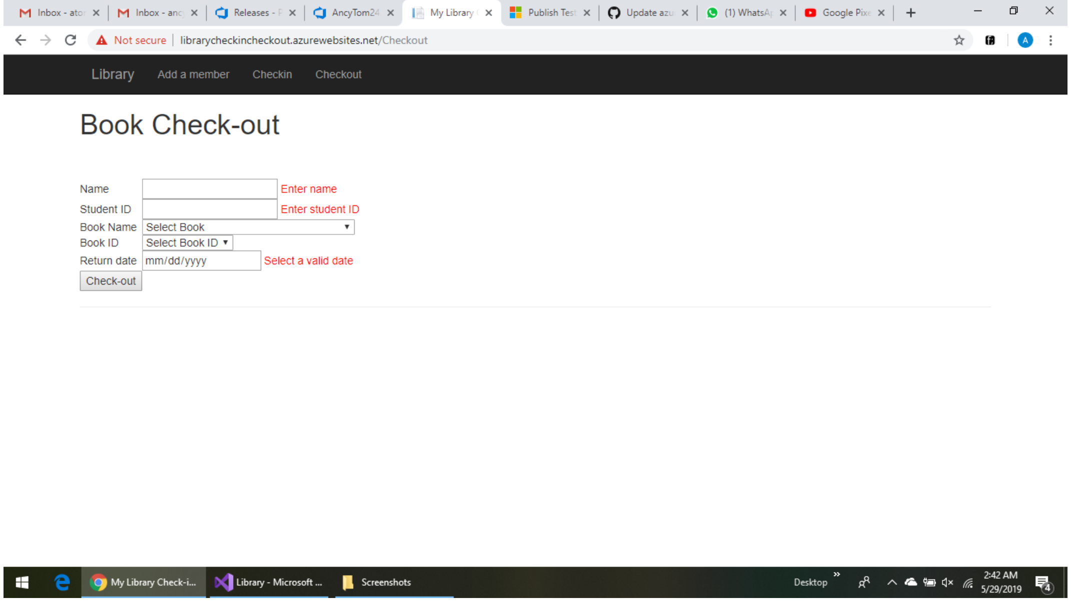
Add New Member



Add new Member

Name	Ancy Tom
Student ID	19953345
Phone	7059750421
Address	627 Macdonald Avenue
City	Sault Ste. Marie
Email	ancytom24@gmail.com

Add New Member Member Added


Not secure | librarycheckincheckout.azurewebsites.net/Checkout
Library
Add a member
Checkin
Checkout

Book Check-out

Name		Enter name
Student ID		Enter student ID
Book Name	Select Book	
Book ID	Select Book ID	
Return date	mm/dd/yyyy	Select a valid date

Check-out

About the application (low level)

Library app is created in .net framework 4.7 with c#. On click of buttons, form details are saved into a txt file. Versioning is controlled by GitHub and build and release is done in Azure. Unit test for .net framework is also used for testing.

Tools Used:

- Visual studio 2019 – IDE for developing
- GitHub – versioning
- Visual studio unit test for .net framework application – unit testing
- Azure DevOps – creating build and release pipeline, deployment.
- Windows OS

Versioning

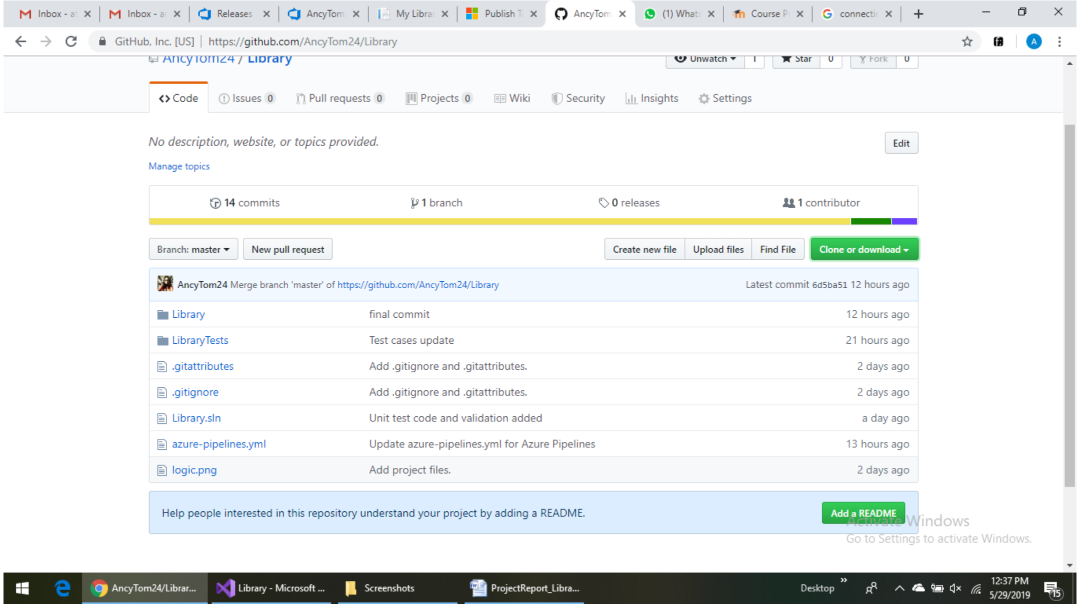
GitHub is a hosting service for version control using Git. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features [1].

Creating a new repository in GitHub

- Sign in to GitHub account
- Select Add Repo
- Enter desired name
- Click create repository button.
- My repo url : https://github.com/AncyTom24/Library.git

Connecting code from visual studio 2019 to GitHub

- Create a project in vs 2019
- View -> Team Explorer -> Home
- Select Sync menu from team explorer
- Select option to link to Git repo
- Signing to GitHub
- Add repository name
- Push code to git.
- Now, every time we have a change in code, we can push it to git.
- Before we start working on the code we should pull the latest code from git if more than one person is making changes to code.



https://github.com/AncyTom24/Library

No description, website, or topics provided.

AncyTom24 Merge branch 'master' of https://github.com/AncyTom24/Library Latest commit 6d5ba51 12 hours ago

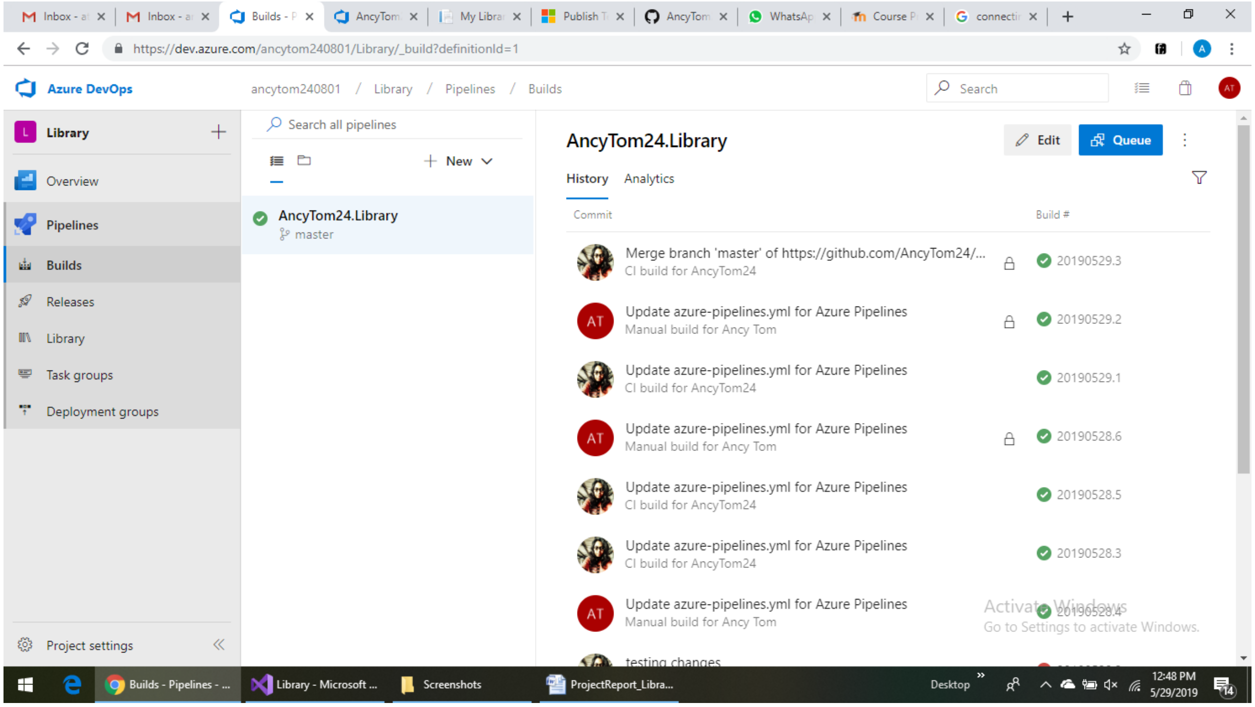
Library	final commit	12 hours ago
LibraryTests	Test cases update	21 hours ago
.gitattributes	Add .gitignore and .gitattributes.	2 days ago
.gitignore	Add .gitignore and .gitattributes.	2 days ago
Library.sln	Unit test code and validation added	a day ago
azure-pipelines.yml	Update azure-pipelines.yml for Azure Pipelines	13 hours ago
logic.png	Add project files.	2 days ago

Help people interested in this repository understand your project by adding a README.

Azure DevOps

- Sign-in to azure devOps.
- Add a build pipeline.
- Select code source from GitHub.

- Select repository.
- Review pipeline YML, make necessary changes and run. – This was quite difficult step for me because I wasn’t sure what should be added to the YML.
- Once it is successfully run, a build pipeline is created.
- Now, we need to create a release pipeline.
    - Sign-in to portal.azure.com.
    - Subscribe to azure to create a resource.
    - Create resource.
    - Get free trial subscription key.
- Go to release tab in azure DevOps.
- Select new release pipeline.
- Select template – Azure app service deployment.
- Select our subscription and resource name in the form.
- Fill out other details like artifact configuration path.
- Save -> Create release.
- Select release – deploy – Success/Fail.
- If failed, check logs and correct errors and deploy again.

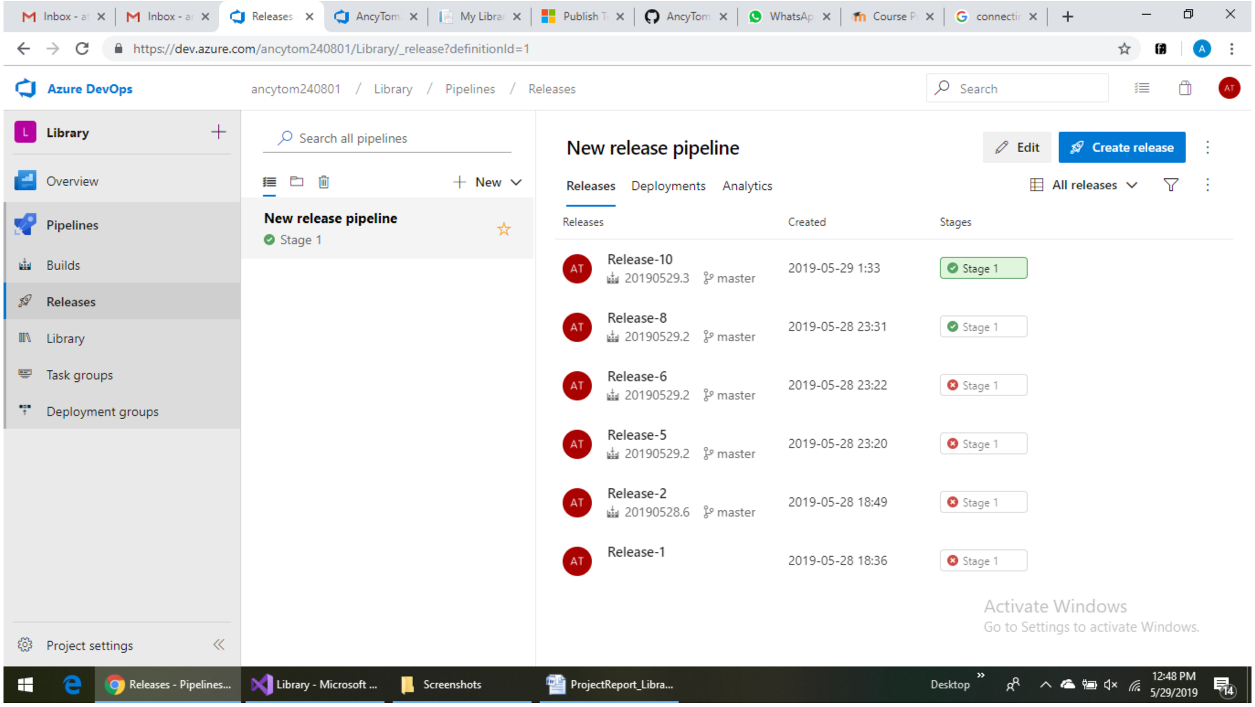


https://dev.azure.com/ancytom240801/Library/_build?definitionId=1

AncyTom24.Library
master

AncyTom24.Library

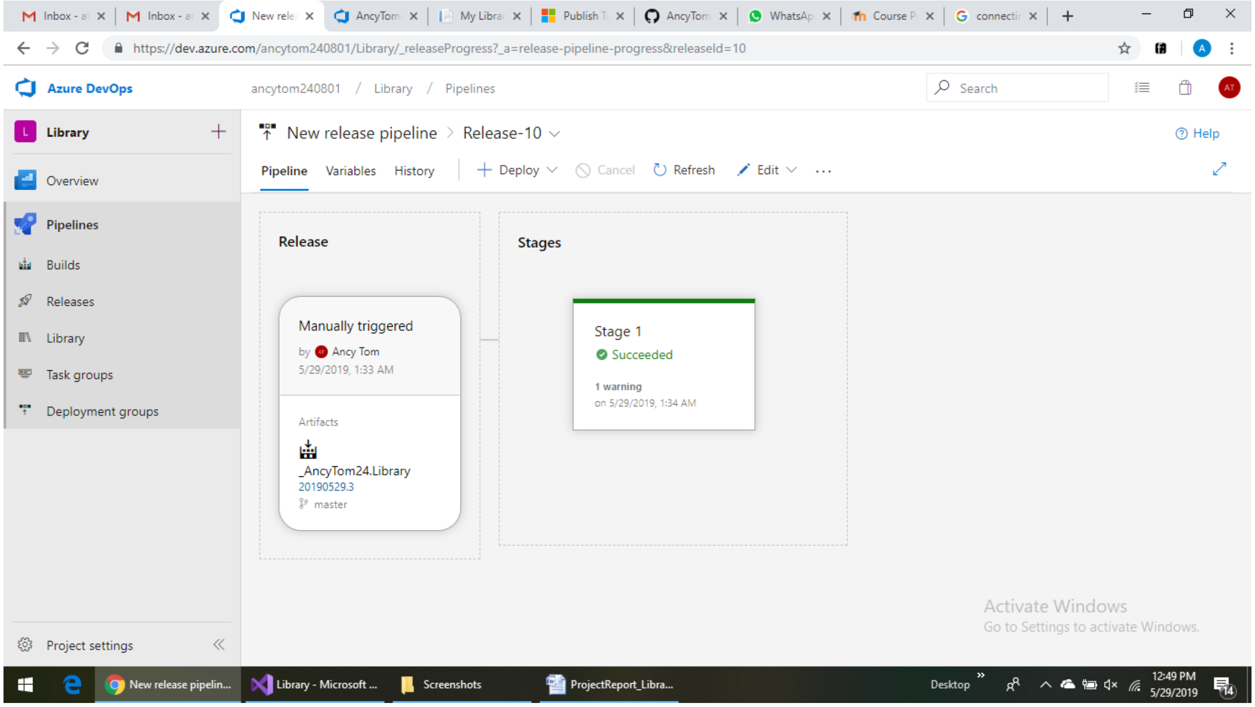
Commit	Build #
Merge branch 'master' of https://github.com/AncyTom24/... CI build for AncyTom24	20190529.3
Update azure-pipelines.yml for Azure Pipelines Manual build for Ancy Tom	20190529.2
Update azure-pipelines.yml for Azure Pipelines CI build for AncyTom24	20190529.1
Update azure-pipelines.yml for Azure Pipelines Manual build for Ancy Tom	20190528.6
Update azure-pipelines.yml for Azure Pipelines CI build for AncyTom24	20190528.5
Update azure-pipelines.yml for Azure Pipelines CI build for AncyTom24	20190528.3
Update azure-pipelines.yml for Azure Pipelines Manual build for Ancy Tom	20190528.4
testing changes	



https://dev.azure.com/ancytom240801/Library/_release?definitionId=1

New release pipeline

Releases	Created	Stages
Release-10 20190529.3 master	2019-05-29 1:33	Stage 1
Release-8 20190529.2 master	2019-05-28 23:31	Stage 1
Release-6 20190529.2 master	2019-05-28 23:22	Stage 1
Release-5 20190529.2 master	2019-05-28 23:20	Stage 1
Release-2 20190528.6 master	2019-05-28 18:49	Stage 1
Release-1	2019-05-28 18:36	Stage 1



https://dev.azure.com/ancytom240801/Library/_releaseProgress?_a=release-pipeline-progress&releaseId=10

New release pipeline > Release-10

Release

Manually triggered
by Ancy Tom
5/29/2019, 1:33 AM

Artifacts
_AncyTom24.Library
20190529.3
master

Stages

Stage 1
Succeeded
1 warning
on 5/29/2019, 1:34 AM

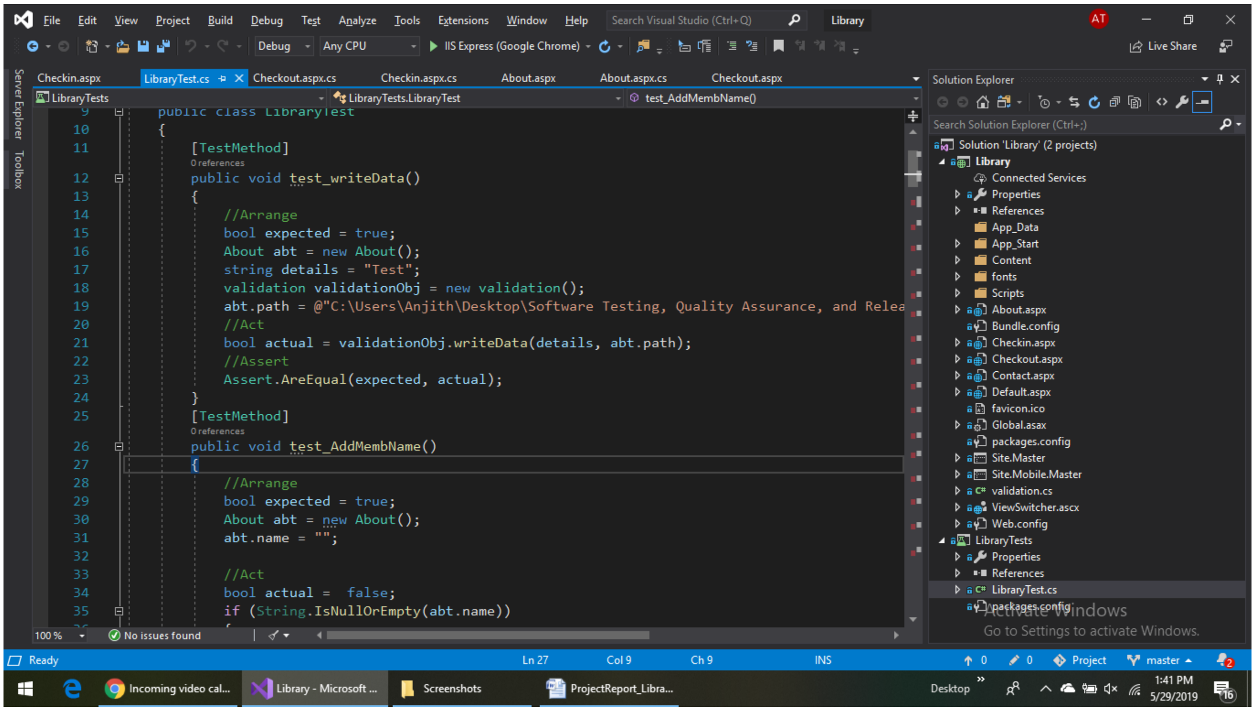
Publish to azure from vs

- Build the solution.
- Select publish from context menu.
- Add azure account to vs.
- Select configuration settings and click publish.
- Published URL: http://librarycheckincheckout.azurewebsites.net/Checkout.

Testing

A testing project for .net framework is created in the same solution folder for unit testing the app. various test cases are:

- Required filed validation and testing
- Writing data to txt file checking



```
    public class LibraryTest
    {
        [TestMethod]
        0 references
        public void test_writeData()
        {
            //Arrange
            bool expected = true;
            About abt = new About();
            string details = "Test";
            validation validationObj = new validation();
            abt.path = @"C:\Users\Anjith\Desktop\Software Testing, Quality Assurance, and Relea
            //Act
            bool actual = validationObj.writeData(details, abt.path);
            //Assert
            Assert.AreEqual(expected, actual);
        }
        [TestMethod]
        0 references
        public void test_AddMembName()
        {
            //Arrange
            bool expected = true;
            About abt = new About();
            abt.name = "";

            //Act
            bool actual =  false;
            if (String.IsNullOrEmpty(abt.name))
```

Test cases

Please refer: Test cases.xlsx

Future scope

This application can be further developed to implement other features like adding/removing books or members. Also saving the data into a proper database management system so that data can be accessed and processed as needed. I am planning to develop this application further and include all possible functionalities and the application could turn into bit complex one. I will also be planning on implementing more test cases and automating the same. This will then be uploaded into my web space. If it works well enough I will also think of outsourcing this app.

Reference

[1]Wikipedia