

MVC PROJECT REPORT

Document goal

This document attempts to collect and present realizations of project task for lecture *Programming in ASP.NET MVC*. By doing so I hope to make assessment of the project easier. Some aspects like Azure setup are visible only after logging in and therefor might be hard to verify otherwise.

Azure DevOps project is publicly available at address <https://dev.azure.com/293101/CV%20Manager>.

Table of contents

Document goal	1
User interface	2
Application uses Razor to generate UI	2
Applications uses Knockout or other framework to do data bindings.....	2
Application doing Get request via AJAX (or AJAX wrapper)	2
Application doing POST via AJAX (or AJAX wrapper).....	2
Student implement form for adding new records	2
Student implement form for record editing	3
Student implemented Bootstrap into application	3
Form is validated on client site.....	3
Controller and integration with Azure and O365.....	3
Controller capture data from query string	3
Controller capturing data from form	3
Controller saving data to database	3
Controller saving data to Azure Storage	3
Controller saving data to SharePoint	4
Student use Azure Function with Service Bus trigger	4
Student use Azure function with time trigger	4
Student connected Application Insight and demonstrate how to read logs	5
Student created at least 3 Unit tests (e.g. using xUnit, NUnit or MSTest) and at least one tests exception	6
Student created at least 1 integration test (e.g. using Spec Flow)	6
Student created at least 3 UI tests (e.g. using Selenium)	6
Form is validated on server site	6
Student integrated Azure AD (any AAD or AAD B2C can be used).....	7

Database and Entity Framework and data	8
Database should contain required tables with elements like indexes, auto incrementation, primary key and foreign key. Database should be presented via SQL Management Studio to confirm that student knows what Entity Framework generates.	8
Data Base deployed on Azure	8
Entity Framework migrations creating and updating database	9
Pictures are stored via Azure Storage	9
Data for HR team saved in SharePoint list	9
DevOps documentation + CI/CD	10
At least 3 user stories. Each should have title, description and acceptance criteria fulfilled. User story should be assigned to user and to sprint.	10
At least 3 task in each user stories. Task should be assigned to user story. Title description, priority and estimation should be fulfilled. Task should be assigned to user and to sprint.	11
14 sprints with start date defined each 1 week long	12
At least 3 test cases defined. Test cases should have assigned user, title and required steps to test application according to application functionality.....	12
Code commits are readable and easy to understand	13
Build pipeline works correctly	13
Release pipeline deploy application on Azure	13
Student created PowerShell script with allow to log into Azure and modify any properties on Azure Resource.....	14
Swagger documentation created for API endpoint.....	14

User interface

Application uses Razor to generate UI

Almost every View uses razor. Apply.cshtml from Views/Application could serve as an example.

Applications uses Knockout or other framework to do data bindings

Knockout is used in Views/JobOffer/Create.cshtml to crate live preview of job offer.

Application doing Get request via AJAX (or AJAX wrapper)

GET request is used in Views/JobOffer/Create.cshtml to load list of companies as well as in Views/JobOffer/Index.cshtml to load pages of job offers without page reload.

Application doing POST via AJAX (or AJAX wrapper)

POST request is used in Views/JobOffer/Create.cshtml in order to submit form.

Student implement form for adding new records

Views/JobOffer/Create.cshtml is used to create new job offer. Submitted offer is then processed by OffersContorler.cs in `public async Task<IActionResult> Post([FromBody] JobOfferCreate offer)`

Student implement form for record editing

Two methods in JobOfferController.cs called Edit are used for edition of job offers as well as connected JobOffer/Edit.cshtml view.

Student implemented Bootstrap into application

Almost every View uses Bootstrap. It can be seen for example in JobOffer/Details.cshtml view.

Form is validated on client site

Client site form validation is done in Views/JobOffer/Create.cshtml with use of [jquery.validate.js](#) library.

Controller and integration with Azure and O365

Controller capture data from query string

Example of explicit data capture from query string can be found in PagingOffersController.cs in Get method but it's quite common in other controllers and methods as well.

```
[HttpGet("{pageNumber}", Name = "Get")]  
//[HttpGet]  
public IActionResult Get([FromRoute] int pageNumber = 1, [FromQuery(Name = "search")] string searchString = "")  
{
```

Controller capturing data from form

This can be found in ApplicationController.cs in Apply method.

```
[HttpPost]  
[ValidateAntiForgeryToken]  
public async Task<ActionResult> Apply(JobApplicationCrateView model)  
{  
    if (!ModelState.IsValid)  
    {  
        return View(model);  
    }  
}
```

Controller saving data to database

This can be found in ApplicationController.cs in Apply method at its bottom.

```
    _context.JobApplications.Add(newApplication);  
    await _context.SaveChangesAsync();  
  
    return RedirectToAction("Details", controllerName: "JobOffer", routeValues: new {id = model.OfferId});  
}
```

Controller saving data to Azure Storage

This is done in ApplicationController.cs by UploadPhotoToBlobStorageAsync and UploadCVToBlobStorageAsync methods. They both call UploadFileToBlobStorageAsync().

```

private static async Task<string> UploadFileToBlobStorageAsync(IFormFile file, string fileName, string containerName)
{
    string connectionString = @"DefaultEndpointsProtocol=https;AccountName=jobofferstoragekd;AccountKey=1ERNYEI2u/olisE5019VWia25IV

    if (CloudStorageAccount.TryParse(connectionString, out var storageAccount))
    {
        CloudBlobClient cloudBlobClient = storageAccount.CreateCloudBlobClient();

        // Get reference to the blob container by passing the name by reading the value from the configuration (appsettings.json)
        CloudBlobContainer container = cloudBlobClient.GetContainerReference(containerName);
        //await container.CreateIfNotExistsAsync();

        // Get the reference to the block blob from the container
        CloudBlockBlob blockBlob = container.GetBlockBlobReference(fileName);

        using (var memoryStream = new MemoryStream())
        {
            file.OpenReadStream().CopyTo(memoryStream); //Convert to byte[]
            var bytes = memoryStream.ToArray();

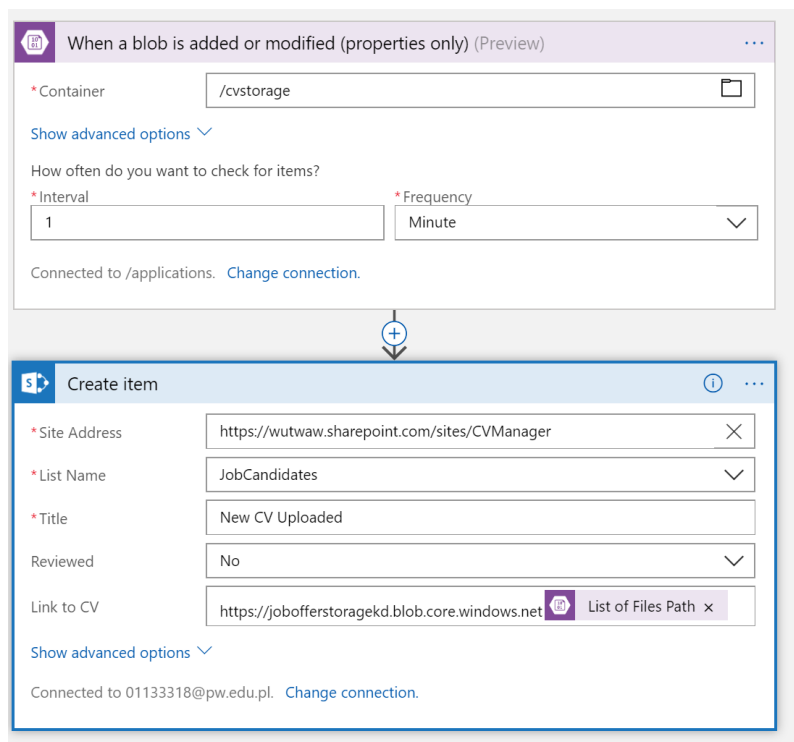
            await blockBlob.UploadFromByteArrayAsync(bytes, index: 0, count: bytes.Length); //Upload
        }

        return blockBlob.Uri.AbsoluteUri;
    }

    return null;
}
  
```

Controller saving data to SharePoint

Controller doesn't save data directly to share point. It instead saves it to blob storage and data is added by Azure LogicApp.



The screenshot shows an Azure Logic App workflow. The first step is a trigger named "When a blob is added or modified (properties only) (Preview)". It is configured with the container "/cvstorage". Below the container field, there are options to "Show advanced options", "How often do you want to check for items?", and "Interval" set to "1" and "Frequency" set to "Minute". The trigger is connected to "/applications".

The second step is an action named "Create item". It is configured with the Site Address "https://wutwaw.sharepoint.com/sites/CVManager", List Name "JobCandidates", Title "New CV Uploaded", Reviewed "No", and Link to CV "https://jobofferstoragekd.blob.core.windows.net/List of Files Path". The action is connected to "01133318@pw.edu.pl".

Student use Azure Function with Service Bus

trigger

Not done

Student use Azure function with time trigger

Azure function logs number of CVs stored in Azure blob storage over time.

Home > jobofferskrzysztofda... - (disabled) TimerTrigger1

jobofferskrzysztofda... - (disabled) TimerTrigger1

Function Apps

All subscriptions

Function Apps

jobofferskrzysztofda...

Functions

(disabled) TimerTrigger1

Integrate

Manage

Monitor

Proxies

Slots (preview)

run.csx

Save

Save and run

```

1 using System;
2 using Microsoft.WindowsAzure.Storage;
3 using Microsoft.WindowsAzure.Storage.Blob;
4
5 public static void Run(TimerInfo myTimer, ILogger log)
6 {
7     log.LogInformation($"C# Timer trigger function executed at: {DateTime.Now}");
8
9     string connectionString = @"DefaultEndpointsProtocol=https;AccountName=jobofferstoragekd;
10
11     if (CloudStorageAccount.TryParse(connectionString, out var storageAccount))
12     {
13         CloudBlobClient cloudBlobClient = storageAccount.CreateCloudBlobClient();
14         CloudBlobContainer container = cloudBlobClient.GetContainerReference("cvstorage");
15         string count = cloudBlobContainer.Metadata["ItemCount"];
16
17         log.LogInformation($"Number of cvs in storate at: {DateTime.Now} is ${count}");
18     }
19 }
20
21
22

```

Student connected Application Insight and demonstrate how to read logs
Application Insight is turned on in app service settings.

[View Application Insights data](#)

Application Insights site extensions

Collect application monitoring data using Application Insights site extension

[Enable](#) [Disable](#)

Link to an Application Insights resource

Your app is connected to Application Insights resource: **CVManagerKrzysztofDabrowski**

Change your resource

Instrument your application

[.NET](#) [.NET Core](#) [Node.js](#) [Java](#)

Collection level

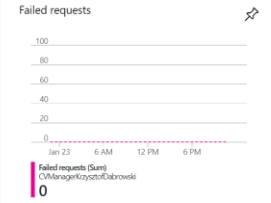
Gain full APM visibility with correlation across boundaries, improved accuracy, and rich usage analytics.

[How to get the most out of your APM data collection](#)

[Recommended](#) [Basic](#)

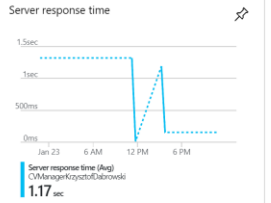
Show data for last: [30 minutes](#) [1 hour](#) [6 hours](#) [12 hours](#) [1 day](#) [3 days](#) [7 days](#) [30 days](#)

Failed requests



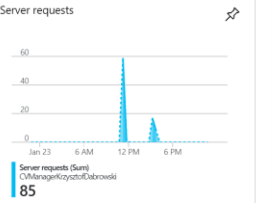
Failed requests (Sum)
CVManagerKrzysztofDabrowski
0

Server response time



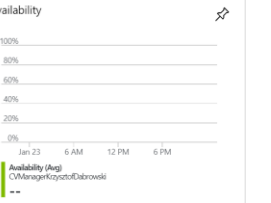
Server response time (Avg)
CVManagerKrzysztofDabrowski
1.17 sec

Server requests



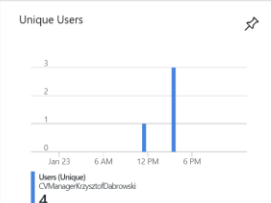
Server requests (Sum)
CVManagerKrzysztofDabrowski
85

Availability



Availability (Avg)
CVManagerKrzysztofDabrowski
100%

Unique Users



Users (Unique)
CVManagerKrzysztofDabrowski
4

Sudden jump in server response time is probably do to host machine warming up and starting as free app service plan is used.

Student created at least 3 Unit tests (e.g. using xUnit, NUnit or MSTest) and at least one tests exception

Not done

Student created at least 1 integration test (e.g. using Spec Flow)

Not done

Student created at least 3 UI tests (e.g. using Selenium)

UI test are located in CVManager.UITests project.

One of the tests:

```
[Test]
public void TryToSubmitNewJobOfferWithoutDescription()
{
    _driver.Navigate().GoToUrl(@"https://cvmanagerkrzysztofzabrowski.azurewebsites.net/JobOffer"); //Go to index

    var createButton = _driver.FindElement(By.LinkText("Create job offer"));
    Assert.IsNotNull(createButton);

    createButton.Click();

    var headers2 = _driver.FindElements(By.TagName("h2"));

    //On creation form there should be such header
    Assert.IsTrue(headers2.Any(h => h.Text == "New Job Offer"));

    _driver.FindElement(By.Id("JobTitle")).SendKeys(text: "Software tester");
    _driver.FindElement(By.Id("SalaryFrom")).SendKeys(text: "4000");
    _driver.FindElement(By.Id("SalaryTo")).SendKeys(text: "6500");

    _driver.FindElement(By.Id("submitFormButton")).Click(); //Submit form and wait for redirection

    WebDriverWait wait = new WebDriverWait(_driver, new TimeSpan(days:0, hours:0, minutes:0, seconds:1));

    Assert.Throws<WebDriverTimeoutException>(code: delegate
    {
        wait.Until(condition: d => d.Url == "https://cvmanagerkrzysztofzabrowski.azurewebsites.net/JobOffer");
    });

    Assert.AreNotEqual(expected: @"https://cvmanagerkrzysztofzabrowski.azurewebsites.net/JobOffer", actual: _driver.Url);

    Assert.Pass();
}
```

Form is validated on server site

Server site validation can be seen in ApplicationController.cs in Apply method. It's using models annotations as requirements.

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Apply(JobApplicationCrateView model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }
}
```

Student integrated Azure AD (any AAD or AAD B2C can be used)

Code related to AAD B2C can be found for example in Startup.cs.

```
services.AddAuthentication(configureOptions: sharedOptions =>
{
    sharedOptions.DefaultScheme = CookieAuthenticationDefaults.AuthenticationScheme;
    sharedOptions.DefaultChallengeScheme = OpenIdConnectDefaults.AuthenticationScheme;
})
.AddAzureAdB2C(options => Configuration.Bind(key: "Authentication:AzureAdB2C", options))
.AddCookie();
```

Log in option is added to shared View.

```
</ul>
@await Html.PartialAsync(partialViewName: "_LoginPartial")
..
```

CV Manager Home Job offers About Contact API Documentation Sign in

Azure AD

JobOfferKrzysztofDabrowski.onmicrosoft.com

JobOfferKrzysztofDabrowski

Azure AD for Office 365

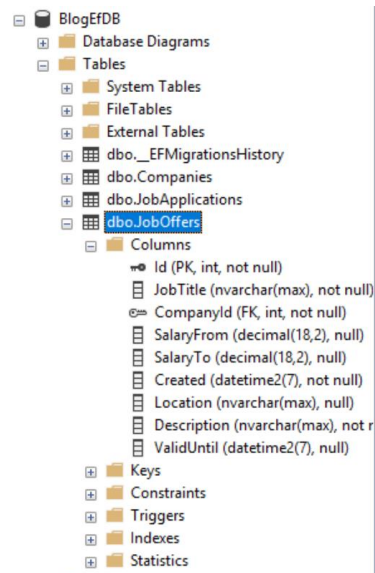
Sign-ins



Database and Entity Framework and data

Database should contain required tables with elements like indexes, auto incrementation, primary key and foreign key. Database should be presented via SQL Management Studio to confirm that student knows what Entity Framework generates.

Data base view in SSMS

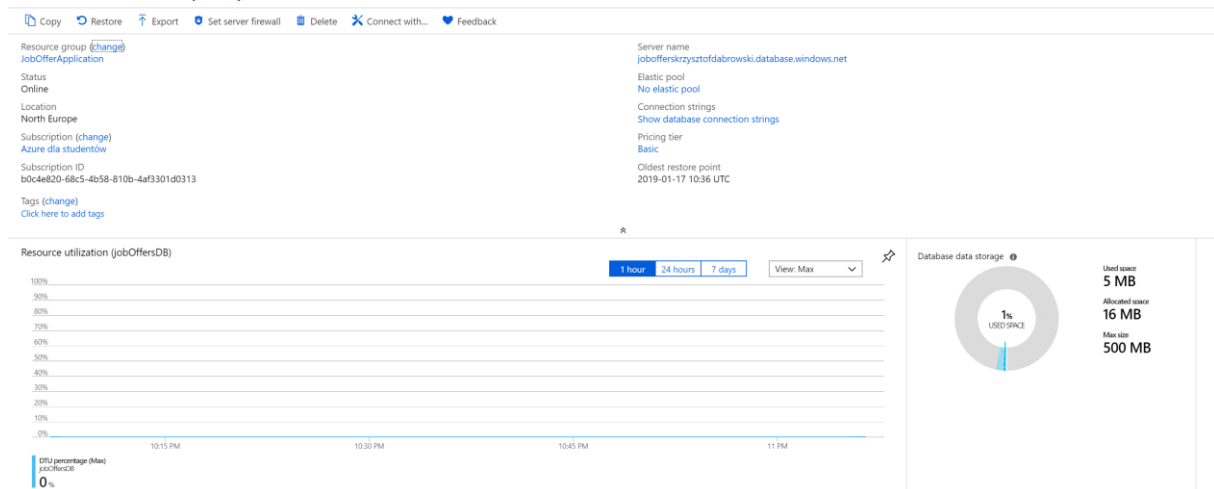


Data context used to generate migrations and set up DB with code first approach.

```

public class DataContext : DbContext
{
    public DataContext(DbContextOptions<DataContext> options) : base(options) { }
    public DbSet<JobApplication> JobApplications { get; set; }
    public DbSet<JobOffer> JobOffers { get; set; }
    public DbSet<Company> Companies { get; set; }
}
  
```

Data Base deployed on Azure



Deployed application have connection string to data base deployed on Azure.

CONNECTION STRING NAME	VALUE	TYPE
azureDB	Hidden value. Click to edit.	SQLAzure

Entity Framework migrations creating and updating database

Migrations are stored in Migrations directory

Pictures are stored via Azure Storage

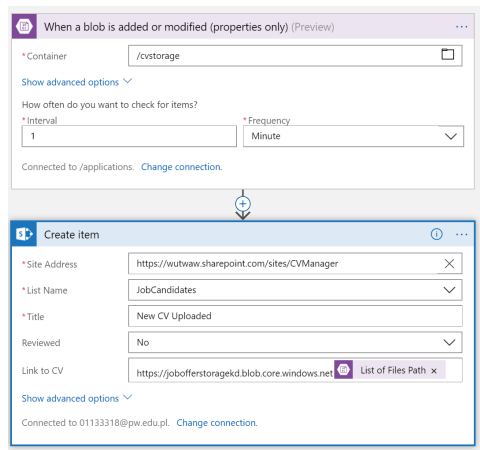
Apply() method from ApplicationController.cs saves user photo to Blob storage using UploadPhotoToBlobStorageAsync() method.

Photo is then displayed in application details view.

```
@if (!string.IsNullOrEmpty(Model.PhotoFileName))
{
    
}
```


Data for HR team saved in SharePoint list

Azure LogicApp adds data to SharePoint list with candidates CVs when they are added to blob storage.



The screenshot shows an Azure Logic App workflow. The trigger is 'When a blob is added or modified (properties only) (Preview)' with the container set to '/cvstorage'. The interval is set to 1 minute. The action is 'Create item' with the following configuration:

- Site Address: <https://wutwaw.sharepoint.com/sites/CVManager>
- List Name: JobCandidates
- Title: New CV Uploaded
- Reviewed: No
- Link to CV: <https://jobofferstorageekd.blob.core.windows.net/cvstorage/0852feb3-6ea8-4438-b567-c8a1b62af7a0.pdf>



EDYTUJ ŁĄCZA

JobCandidates

+ nowy element lub edytuj tę listę

Wszystkie elementy

✓ Title	Reviewed	Link to CV
New CV Uploaded	Nie	https://jobofferstorageekd.blob.core.windows.net/cvstorage/0852feb3-6ea8-4438-b567-c8a1b62af7a0.pdf

EDYTUJ ŁĄCZA

Strona główna

Notes

Dokumenty

Strony

Ostatnie

JobCandidates

Zawartość witryny

Kosz

DevOps documentation + CI/CD

At least 3 user stories. Each should have title, description and acceptance criteria fulfilled. User story should be assigned to user and to sprint.

Example user stories:

USER STORY 106

As someone viewing job application I would like to see user's photo if there is one

Krzysztof Dąbrowski0 commentsAdd tag

StatusClosedAreaCV ManagerReasonAcceptance tests passIterationCV Manager\Sprint6WebDeployment

Updated by Krzysztof Dąbrowski just now

DescriptionWhen viewing details of job application there should be a section that displays photo that user uploaded if he did so.

Acceptance CriteriaIf job application has photo connected with it it should be visible in application details view.

DiscussionAdd a comment. Use # to link a work item, / to link a pull request, or @ to mention a person.

PlanningStory PointsPriority2Risk

ClassificationValue areaBusiness

Development+ Add linkDevelopment hasn't started on this item.Create a new branchRelated Work+ Add link - Parent93 Store user files on blob storageUpdated 19.01.2019, ClosedChild (3)108 Create action to send photo to the viewUpdated 18.01.2019, Closed107 Download photo from blob storageUpdated 18.01.2019, Closed109 Render photo in viewUpdated 18.01.2019, Closed

USER STORY 17

As a normal user I would like to be able to enter job name in search field and see all offers that have that text in their names

Krzysztof Dąbrowski0 commentsAdd tag

StatusClosedAreaCV ManagerReasonAcceptance tests passIterationCV Manager\Sprint2UI

Updated by Krzysztof Dąbrowski a minute ago

DescriptionOn the offers list page there should be input box for searching for job offers with given phrase in title.

Acceptance CriteriaThere is a functional search bar for job offers. Search should be case sensitive.

DiscussionAdd a comment. Use # to link a work item, / to link a pull request, or @ to mention a person.

PlanningStory PointsPriority2Risk

ClassificationValue areaBusiness

Development+ Add linkDevelopment hasn't started on this item.Create a new branchRelated Work+ Add link - Child (3)19 Controller action to find correct offersUpdated just now, Closed10 Search fieldUpdated 09.01.2019, Closed122 Search submit buttonUpdated just now, New

USER STORY 10

As a visitor of the site I would like to see detailed job description when i click on job offer name

Krzysztof Dąbrowski0 commentsAdd tag

StatusClosedAreaCV ManagerReasonAcceptance tests passIterationCV Manager\Sprint2UI

Updated by Krzysztof Dąbrowski just now

DescriptionWhen user click on job offer from offers list he should be redirected to detailed view of that offer.

Acceptance CriteriaJob offer details are shown when user clicks on job offer name. All public data of job offer should be displayed.

DiscussionAdd a comment. Use # to link a work item, / to link a pull request, or @ to mention a person.

PlanningStory PointsPriority2Risk

ClassificationValue areaBusiness

Development+ Add linkDevelopment hasn't started on this item.Create a new branchRelated Work+ Add link - Child (3)12 Crate view of job detailsUpdated just now, Closed11 Make controller action to display detailsUpdated 07.01.2019, Closed13 Make current job list have links to the detailsUpdated just now, Closed

At least 3 task in each user stories. Task should be assigned to user story. Title description, priority and estimation should be fulfilled. Task should be assigned to user and to spring.

Example tasks:

TASK 96
96 Add form control for file upload

Unassigned 0 comments Add tag

Save & Close Follow Refresh ...

Updated by Krzysztof Dąbrowski just now

Details (1) [icon]

Status: ● Closed Reason: ⏸ Completed Area: CV Manager Iteration: CV Manager\Sprint6WebDeployment

Description
On job application submission form there should be a button for uploading user photo.

Discussion [icon] [dropdown]

Planning
Priority: 2
Activity: Design
Effort (Hours)
Original Estimate: 0.5
Remaining: [icon]
Completed

Development
+ Add link
Development hasn't started on this item.
Create a new branch
Related Work
+ Add link
Parent
95 As a job applicant I would like to add a photo to my applica...
Updated 19.01.2019. ● Closed

TASK 98
98 Create blob storage on Azure

Unassigned 0 comments Add tag

Save & Close Follow Refresh ...

Updated by Krzysztof Dąbrowski just now

Details (1) [icon]

Status: ● Closed Reason: ⏸ Completed Area: CV Manager Iteration: CV Manager\Sprint6WebDeployment

Description
On Azure platform should be created appropriate storage for saving user photos

Discussion [icon] [dropdown]

Planning
Priority: 2
Activity: Deployment
Effort (Hours)
Original Estimate: 2
Remaining: [icon]
Completed

Development
+ Add link
Development hasn't started on this item.
Create a new branch
Related Work
+ Add link
Parent
95 As a job applicant I would like to add a photo to my applica...
Updated 19.01.2019. ● Closed

TASK 44
44 Create API Controller

Krzysztof Dąbrowski 0 comments Add tag

Save & Close Follow Refresh ...

Updated by Krzysztof Dąbrowski just now

Details (1) [icon]

Status: ● Closed Reason: ⏸ Completed Area: CV Manager Iteration: CV Manager\Sprint4WebAPI

Description
There should be a controller class that could include methods for RESTful API

Discussion [icon] [dropdown]

Planning
Priority: 2
Activity: Development
Effort (Hours)
Original Estimate: 1
Remaining: [icon]
Completed

Development
+ Add link
Development hasn't started on this item.
Create a new branch
Related Work
+ Add link
Parent
58 As a developer I would like to see API documentation
Updated 15.01.2019. ● Closed

14 sprints with start date defined each 1 week long

CV Manager		
Sprint1Init	15.10.2018	21.10.2018
Sprint2UI	22.10.2018	28.10.2018
Sprint3DB	29.10.2018	04.11.2018
Sprint4WebAPI	05.11.2018	11.11.2018
Sprint5Identity	12.11.2018	18.11.2018
Sprint6WebDeployment	19.11.2018	25.11.2018
Sprint7Sharepoint	26.11.2018	02.12.2018
Sprint8LogiApp	03.12.2018	09.12.2018
Sprint9CognitiveServices	10.12.2018	16.12.2018
Sprint10Tests	17.12.2018	23.12.2018
Sprint11ImproveHompePage	24.12.2018	28.12.2018
Sprint12AdditionalAPI	31.12.2018	04.01.2019

At least 3 test cases defined. Test cases should have assigned user, title and required steps to test application according to application functionality

Example test cases:

TEST CASE 115

115 Job offer should be added

Krzysztof Dąbrowski 0 comments Add tag

Save Follow

(Ctrl+S) ated by Krzysztof Dąbrowski just now

State: Ready Reason: Completed Area: CV Manager Iteration: CV Manager/Sprint2UI

Steps

Steps	Action	Expected result	Attachments
1.	Select Job Offers tab		
2.	Click "Create job offer" button at the bottom right of the page		
3.	Fill in all form fields (Description should be at least 50 characters long)		
4.	Click "Create" button		
5.	You should be redirected to list of job offer		
6.	Created offer should be in the job offers list		
7.			

Click or type here to add a step

Development

+ Add link
Development hasn't started on this item.
Create a new branch

Related Work

+ Add link

Link	State	Latest Update	Comment
Parent			
As a site manag...	Closed	Updated 10.01.20...	

Status

Priority: 2

Automation status: Planned

TEST CASE 110

110 Job offers are displayed test

Krzysztof Dąbrowski 0 comments Add tag

Save Follow

Updated by Krzysztof Dąbrowski 19.01.2019

State: Ready Reason: Completed Area: CV Manager Iteration: CV Manager/Sprint2UI

Steps

Steps	Action	Expected result	Attachments
1.	Type website adres in the browser		
2.	When page loads click "job offers" from top menu		
3.	Wait for offers to load		
4.	Job offers should be displayed		

Click or type here to add a step

Development

+ Add link
Development hasn't started on this item.
Create a new branch

Related Work

+ Add link

Link	State	Latest Update	Comment
Parent			
As a visitor of the sl...	Closed	Updated 07.01.20...	

Status

Priority: 2

TEST CASE 111

111 Add photo button working

Krzysztof Dąbrowski 0 comments Add tag

Stat: **Ready** Reason: Completed Area: CV Manager Iteration: CV Manager\Sprint6WebDeployment Updated by Krzysztof Dąbrowski 19.01.2019

Steps

Steps	Action	Expected result	Attachments
1.	Go to Job offers tab		
2.	Wait for job offers to load		
3.	Select any offer from the list		
4.	Click "Apply now" button		
5.	Under section "Date of Birth" should be a button to upload a photo		
6.	Click button to add photo		
7.	Select any photo file from file explorer		
8.	Confirm file selection		
9.	Fill in First name and Last name		
10.	Click "Apply" button at the bottom of the form		
11.	You should be redirected to the details of selected job offer and application should be listed at the bottom		

Click or type here to add a step

Development

+ Add link
Development hasn't started on this item.
[Create a new branch](#)

Related Work

+ Add link

Link	State	Latest Update	Comment
Tests			
95 As a job applica...	Closed	Updated 19.01.20...	

Status

Priority
2

Automation status
Not Automated

Code commits are readable and easy to understand

Code commits can be read in Repos section in Azure DevOps

Build pipeline works correctly

Build pipeline is set up in Azure DevOps.

#20190121.1: Set up CI with Azure Pipelines

Manually run pon. at 19:15 by Krzysztof Dąbrowski CVManager master 2c5e314 Retained by release

Logs Summary Tests

Agent job 1 Job Started: 21.01.2019, 19:21:00

Pool: Hosted Ubuntu 1604 Agent: Hosted Agent

Step	Status	Duration
Prepare job	succeeded	< 1s
Initialize Agent	succeeded	2s
Initialize job	succeeded	2s
Checkout	succeeded	6s
Restore	succeeded	1m 9s
Build	succeeded	26s
Test	succeeded 1 warning	1s
Publish	succeeded	4s
Publish Artifact	succeeded	5s
Post-job: Checkout	succeeded	< 1s
Report build status	succeeded	< 1s

Release pipeline deploy application on Azure

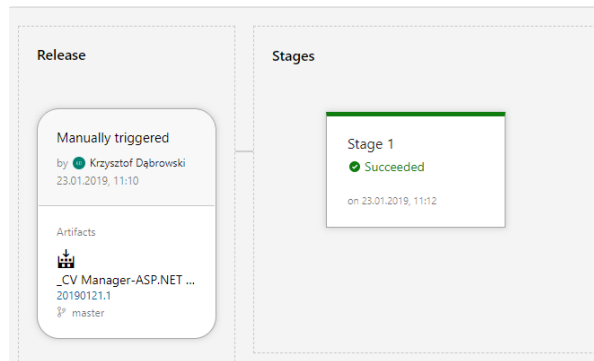
Release pipeline is set up in Azure DevOps.

New release pipeline

Releases Deployments Analytics* Edit Create a release

All releases

Releases	Created	Stages
Release-3 20190121.1 mas...	2019-01-23 11:10	Stage 1
Release-2 20190121.1 mas...	2019-01-23 11:05	Stage 1
Release-1 20190121.1 mas...	2019-01-22 11:26	Stage 1



Run on agent			Started: 23.01.2019, 11:12:13
Pool: Hosted VS2017 · Agent: Hosted Agent			... 23s
✓ Initialize Agent · succeeded			<1s
✓ Initialize job · succeeded			3s
✓ Download artifact - _CV Manager-ASP.NET Core-CI - drop · succeeded			3s
✓ Deploy Azure App Service · succeeded			15s

Student created PowerShell script with allow to log into Azure and modify any properties on Azure Resource.

Not done

Swagger documentation created for API endpoint

XML comments are written for API controller's methods. And Swagger is connected.

Companies		▼
GET	/api/Companies	Get all companies that can post job offers
Offers		▼
GET	/api/Offers	Get all job offers that have selected text inside. If searchString is empty then it returns all offers.
POST	/api/Offers	Add new offer to the data base
GET	/api/Offers/{id}	Get job offer with given id
PUT	/api/Offers/{id}	Updates specified job offer
DELETE	/api/Offers/{id}	Deletes job offer with specified id
PagingOffers		▼
GET	/api/PagingOffers/{pageNumber}	Returns page of job offers records. Can be searched.
Models		▼
JobOfferCreate >		
JobApplication >		
JobOffer >		
Company >		