

DSD 605 Software Testing and Security- Assessment 2

Module Code	DSD 605	Module Title		Software Testing and Security	
Level	6	Credit Value of Module		15	
Assessment Task	Create a web application, or modify an existing one, with authentication and authorization management, principles of data privacy using security industry practices.				
Assessment No	2	Total marks	60	Weighting	40%
Type of Submission					
Method of Submission					
Due Date	XXXXXX				
Learning Outcomes addressed					
Learning outcomes: LO2: Select and apply Software testing and quality assurance methods including technical documentation LO3: Design and create a secure web application to meet specified requirement including authentication and authorization management based on security best practice					
Passing Criteria					
Software Testing and Security Total Marks: 60 Total Weighting: 60%					
A+	90-100%	C+	60-64%		
A	85-89%	C	55-59%		
A-	80-84%	C-	50-54% PASS MARK		
B+	75-79%	D+	45-49%		
B	70-74%	D	40-44%		
B-	65-69%	D-	0-39%		
Resubmission or Reassessment Requirements					

DSD 605 Software Testing and Security- Assessment 2

If a student performance fails to meet any of the requirements in the marking schedule, the student may be given opportunity to resubmit or be reassessed.

Resubmission: If it is apparent (before or during marking) that evidence is incomplete, either by omission or error, the assessor may direct the student's attention to the areas of breach, in general terms, and invite him/her to resubmit the work with the identified areas corrected. Only one resubmission opportunity may be given.

Reassessment: If, after valid processing has been applied, a student is still judged not to have reached the standard required, the student may be offered a further assessment opportunity. A new date will be agreed upon, at the assessor's discretion. Only one reassessment opportunity may be given.

Assessment brief and instructions

DSD 605 Software Testing and Security- Assessment 2

Instruction for the assessment

This assessment is worth 40% of the total marks for the course.

The assessment must be delivered individually.

Your submission must include references (APA 7 format) if referring to the source material. Failure to do so will have serious consequences.

You must fill out and submit the ATC Vision College cover sheet.

Assessment Conditions

This is a resource-based assessment. This means that you may have access to any relevant resources to assist you.

This could include your learning materials, information on the Internet, and so on. However, all work must be your own with no assistance from any other person.

Assessment brief

Create a copy of the bike website you created in DSD602. We are going to modify it to include authorization and authentication.

1. Set the following Password properties to make it easier for debugging

- RequireDigit = false;
- RequireLowercase = false;
- RequireNonAlphanumeric = false;
- RequireUppercase = true;
- RequiredLength = 6;
- RequiredUniqueChars = 1;
- RequireConfirmedEmail = false;

Screenshot the settings and add to your answer sheet

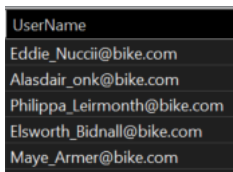
2. Modify the Register Page to automatically set confirm email to true.

Screenshot the settings and add to your answer sheet – include a line of code above and below the entry

3. Customise the Access Denied page to add an image of your own choice, something unique

Screenshot the Access Denied screen with the image and add to your answer sheet

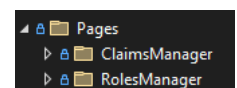
4. Add the following staff into your ASPNetUsers table via the Identity Register system(These guys are in your staff table already)



UserName
Eddie_Nucci@bike.com
Alasdair_onk@bike.com
Philippa_Leirmonth@bike.com
Elsworth_Bidnall@bike.com
Maye_Armer@bike.com

5. Create the roles of Admin, Manager, Staff, using the RoleManager interface.

- Eddie: Manager
- Alasdair: Manager
- Phillipa: Admin
- Elsworth: Staff
- Maye: Staff



DSD 605 Software Testing and Security- Assessment 2

Screenshot the Roles screen showing name and role and add to your answer sheet

Create the claims below, using the ClaimsManager interface

6. Create the following staff User Claims

User Claims
New

Eddie_Nuccii@bike.com

Type	Value
DateOfBirth	2000-1-1
Joining Date	2020-1-12
Permission	View Roles
Permission	View Claims
Permission	Delete Stock
Permission	Edit Stock

Alasdair_onk@bike.com

Type	Value
DateOfBirth	2000-1-1
Joining Date	2020-1-12
Permission	Edit Stock

Philippa_Leirmonth@bike.com

Type	Value
DateOfBirth	1970-1-1
Joining Date	2020-1-12
Permission	View Roles
Permission	View Claims
Permission	Delete Stock

Elsworth_Bidnall@bike.com

Type	Value
DateOfBirth	2000-1-1
Joining Date	2023-1-18
Permission	View Roles

This Person Less than 6 months from your current date

Maye_Armer@bike.com

Type	Value
DateOfBirth	2010-1-1
Joining Date	2020-1-12

This Person Less than 18 years old from your current date

Screenshot the Claims screen and add to your answer sheet

7. Create the following policies and apply them.

Only....

- Staff with over 6 months of service and permission "View Roles" can view RolesManager folder
- Staff with over 6 months of service and permission "View Claims" can view ClaimsManager Folder
- Staff with permission Delete Stock can Delete Stock
- Staff with permission Edit Stock can Edit Stock
- Staff over 18 can add stock
- Staff in Admin Role can view the Staff details by controlling the StaffsController. Screenshot your answer

DSD 605 Software Testing and Security- Assessment 2

Screenshot the code and add to your answer sheet

8. Generate a WebAPI Framework, add a StockApiController and install Swagger / Swatchbuckle
9. Create a ReactJs front end that consumes the Web API.
10. Outline the Purpose of CORS
11. Outline how CORS operates and the types of restrictions it offers.
12. Implement the CORS policy between them so that the front end consumes the data from the back end and displays it on your app.

Screenshot the React page with the stock on it, just a sample as its really big, and add to your answer sheet

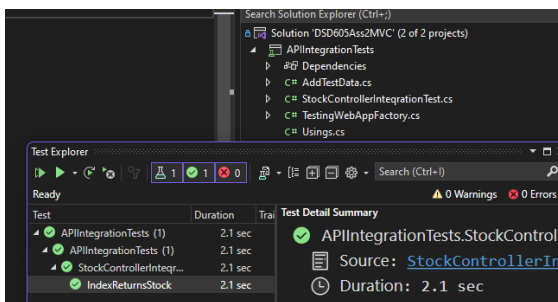
Screenshot the CORS code and add to your answer sheet

13. Create a new Integration Project, name it APIIntergrationTest

Create and integration test for the StockAPIContoller that uses 1 mock entry and checks that it returns stock.

ScreenShot the StockControllerIntegrationTest class and add to your answer sheet

Screenshot the Test Explorer result with the APIIntergrationTest Project in (see sample) and add to your answer sheet



14. Create a video of your project showing how the policies are working and the API is working.

MARKING RUBRIC

60 Marks Total							
Assessment Acceptance criteria	Grade Distribution				Student Mark	Comments	
1. Set the following Password properties	0	1	2	3	3		
	Insufficient or Not attempted	Few properties completed	Most properties completed	All properties completed			
2. Modify the Register Page	0		5		5		
	Insufficient or Not attempted		feature completed				
3. Customise the Access Denied page	0		5		5		
	Insufficient or Not attempted		feature completed				
4. Add the staff	0	1	2	6	6		
	Insufficient or Not attempted	Few staff completed	Most staff completed	All staff complete			
5. Create the roles of Admin, Manager, Staff	0	1	2	6	6		
	Not completed at all	Few roles completed	Most roles completed	All roles complete			
6. Create the following staff User Claims	0	1	2	6	6		
	Insufficient or Not attempted	Few Claims completed	Most Claims completed	All Claims complete			
7. Create the following staff Policies and apply them.	0	5	7	10	10		
	Insufficient or Not attempted	Some Policies completed	Most Policies completed	All Policies complete			
10. Outline the Purpose of CORS	0	1	2		2		
	Insufficient or Not attempted	Answer Sufficient	Answer Complete				
11. Outline how CORS operates and the types of restrictions.	0	1	2		2		
	Insufficient or Not attempted	Answer Sufficient	Answer Complete				
12. Implement the CORS policy	0		5		5		
	Insufficient or Not attempted		feature Complete				
13. Create a new Integration Project			7	10	10		
	Not completed at all	3 Policies completed	4 Policies completed	5 Policies completed		All Policies complete	
Total 40 Marks				Total Marks		60	

MARKING RUBRIC

NOTES:

***Sufficient** is defined as per the points below:

- a) The answer provided by the student is correct and is supported by relevant ideas and/or supporting details.
- b) Information is presented in a logical manner that is easily followed.
- c) There is minimal interruption to the work due to misspellings and/or grammatical errors.

***Insufficient** is defined as not meeting any of the points above (a, b and c).

***Reasonable** is defined as meeting at least point a) above.