WEB APPLICATION DEVELOPMENT

CIS2169

lacovos lacovides 24128341

Contents

Repository URL:	2
Netlify URL:	2
1.0 Introduction	
2.0 Diagrams	
3.0 Sketches	5
5.0 Code Implementation	7
6.0 Deployment planning and practice.	16
7.0 Final Application (final output when opening the application)	21
8.0 Analysis Post-mortem	22
9.0 Conclusion	23
10.0 References	23

Table of Figures

Figure 1: Activity Diagram
Figure 2: Class Diagram
Figure 3: Use case diagram
Figure 4: Sketch number 16
Figure 5: Sketch number 2
Figure 6: Sketch number 37
Figure 7: Pushing in command line console
Figure 8: GitHub commits9
Figure 9:GitHub commits
Figure 10: GitHub commits 11
Figure 11: Developers folder in Personal Computer 12
Figure 12: Create Module coding13
Figure 13: Create degree coding 13
Figure 14: Create Assessment coding14
Figure 15: Module Slots coding14
Figure 16: Main JavaScript file coding 15
Figure 17: Netlify URL
Figure 18: Deployed Application
Figure 19: Test 1
Figure 20: Test 2
Figure 21: Test 3
Figure 22: Test 4
Figure 23: Test 5
Figure 24: Final Create Module Page
Figure 25: Final Create Degree Page
Figure 26: Final Create Assessment Page
Figure 27: Final Module Slots Page 22

Repository URL: <u>https://github.com/lacovoslacovides/CW2-1.git</u>

Netlify URL: https://iacovoscw2.netlify.app/

1.0 Introduction

An application developer has been asked to further develop and extend an already existing webbased academic management and tracking application, which at this point it tracks all academics research activities. The developer must enhance this application by improving several things that the application is lacking. For any type of work, easy and accurate planning should be followed to ensure that everything will be accurate and correct at the end. (Contributor, 2020) In this case, a correct plan includes, designing the system using appropriate UML diagrams and ensuring that the links between the builds are made possible. Understanding the way that something links to another are one of the most important aspects when it comes to developing such an application. When the diagrams are made, the implementation face will come, where the developer will use the diagram to build an application with the same links that he has designed (Bluescape, 2019). When the code is finished, various tests should be made, which will eliminate the chances of errors when the application is published. All of the steps made will be explained in this report.

2.0 Diagrams

In this section, the UML diagrams that have been designed will be explained.



Figure 1: Activity Diagram

The diagram above is an Activity Diagram. It shows the way with which the developer will begin the process of understanding the case study, all the way to the very end which is deploying the application. The very first activity that the developer had to do is to observe the existing application, see what it is lacking, and then read and try to understand what the case study is asking, along with the aims and the objectives. If the developer understood what needs to be done, he would then create the diagrams and then continue to implement the code. After the appropriate tests have been done, he would deploy the application.



Figure 2: Class Diagram

The diagram above is a Class diagram. In this diagram, some classes have been developed with the aid of the given case study. The developer had read the case study over and over again until he managed to come up with these classes. The code will be built around these classes trying to connect every bit of code. Six classes have been made and each one has a primary key and a foreign key assigned to make the relationships between the classes possible and accurate.

-	
	5
	(Frenchen)
	(THI Jul day' poge lead and
	Maulo Star 12 Cincludes?
	(create "we trane /
	() page (includes)
	Extend a linchades?
	i landiation
	The state of the s
	cincludes'
	Developer Greate an
	(add modele page
	Casto "all
	() (Note des
	() () () () () () () () () () () () () (
	(exands)
	(Add module a linutist
	1) Slas
	(use application to
	Ventues
	USIS (Hod a module) [Circludes?
	(includes)
	(Add a degree)
	and assessment

Figure 3: Use case diagram

The diagram above is the Use Case diagram that the developer has designed. The diagram shows a possible way with which the user will interact with the application's user interface.

3.0 Sketches

When a developer is asked to develop an application, one important thing to do before starting to code is to sketch some basic ways the page might look like. Thus, the developer has created some sketches about what the website's home page will look like along with the other pages. In this section, the sketches will be provided along with an explanation about each one.

	*	
0	Stotches.	
		-

Figure 4: Sketch number 1

In the sketch above the home page is presented. There will be a bar on the top with which the user will be able to do various things, like adding a new module, creating a degree, or an assessment. There will be a quick way to add a module on the left-hand side of the page where the user can enter the name of the new module, the hours that it takes, the academic, etc. At the bottom of the page, there will be a table where all the modules will be displayed.

[mm mm mm]	
Tessions Add ons	•
	-

Figure 5: Sketch number 2

In the sketch above, another way in which the home page might look is presented. There will be a top navigation bar where the user can navigate to the rest adding pages, which include adding modules, creating degrees, and creating assessments. There will be a button called lessons which will be another gateway to the adding lessons page. At the bottom of the page, there will be a table showing the modules along with the rest of the important information about each module.



Figure 6: Sketch number 3

In the sketch above, the third way with which the home page might look like. As in the other sketches, there is a navigation bar on the top along with a logo. Below that there are some boxes with which the user can add a module. At the bottom of the page, there is a table where the modules will be presented along with important information about each module.

5.0 Code Implementation

When the developer has finished with all of the designs and the diagrams that will help him with the development, the code needs to be written. There were some basic files already developed for the application.

The developer has used a local repository through GitHub to achieve developing this application. Every time the developer had been changing anything in the code, he would "push" those changes in the repository, along with an explanation message telling what he changed.

The latter will start writing the code and when the code is done he will do some tests, to ensure that there are no errors and then deploy the page.

CW2-1 : git — Konsole



Figure 7: Pushing in command line console.

- → The first line of code is "git add .". This code adds new or changed files in your working directory to the Git staging area.
- → The second line of code is "git status". With this code, the current state of the working directory and staging area is shown. It shows you which changes have been staged, which haven't, and which files Git isn't tracking.
- → The third line of code is "git commit -m"Edited the class diagram .xmi fie". With this code, the developer is adding an explanatory message next to the commit which will show on GitHub.

→ The final line of code is *"git push"* which pushes the commitments to the repository in GitHub.

forked from p	rofharimohanpandey/CW2	Projects 171 Wildi (1) Security Ly Insights	Settings		
	Image: Provide the second	Go	to file Add file - Code -	About	ŝ
	This branch is 18 commits ahead of profhari	ntribute 👻 🗘 Fetch upstream 👻	No description, website, or topics provided.		
	Iacovoslacovides Added another two	rows with data in the table in degree page 2e6056	1 10 minutes ago 🕥 51 commits	 □ Readme ☆ 0 stars ○ 0 supplier 	
	C .directory	Continued working on the pages.Added images	yesterday	Image: Watching Image: Watching Image: Watching	
	Iacovides_I_24128341_CW2 (1).docx	Continued the report(Added Sketches and Diagrams)	2 days ago		
	C README.md	Update README.md	3 years ago	Releases	
	activitydiagram.xmi	Designed activity diagram	3 days ago	No releases published	
	assessment.html	Edited the script files for assessment page	19 minutes ago	Create a new release	
	assessment.jpg	Continued working on the pages.Added images	yesterday		
	🗅 books.jpg	Continued working on the pages.Added images	yesterday	Packages	
	Classdiagram.xmi	Designed class diagrams	3 days ago	No packages published Publish your first package	
	🗅 degree.html	Added another two rows with data in the table in degree	page 10 minutes ago		
	🗅 degree.jpg	Continued working on the pages.Added images	yesterday	Languages	
	🗅 general.jpg	Continued working on the pages.Added images	yesterday	JavaScript 48.1% • HTML 47.1%	
	🗅 ggg.xmi	Created the pages of the application	2 days ago	• CSS 4.8%	
	home.html	Created Time Slot Page and Degree Page	4 hours ago		

Figure 8: GitHub commits.

As you can see there have been 51 commits to this application which means that the developers who worked for this application have committed their changes 51 times.

covosla	acovides / CW2-1 (Public)	
Code	imonanpanoeyic.wz	
ooue		
ۍ ۲	* master -	
-0-	Commits on Apr 21, 2022	
	Added another two rows with data in the table in degree page	C 2e60561 <>
	Added the table for the times slot page	□ e43dccb <>
	Edited the text boxes in the time slot page	□ 1856953 <>
	Edited the script files for assessment page	C 6730a8e <>
	Added table in assessments page the format of the table in assessment of the table of the table of the table of table o	₽ 806344 €
•	Edited assessment file	
	Created Time Slot Page and Degree Page	□ 1d16968 <>

Figure 9:GitHub commits.

lacovos lacovides

24128341

-0-					Upd Upd
	Commits on Apr 18, 2022				
	Designed class diagrams the accovoslacovides committed 3 days ago	D	6b9bc3d	\diamond	
	Designed activity diagram the	Ŀ	8ddd3ac	\Diamond	
	Designed activity diagram	Ŀ	f4c43d7	\diamond	
	astarted the report Iteorovoslacovides committed 3 days ago	Ŀ	66741c2	\diamond	
	Etest the second secon	Ŀ	04e8ded	\diamond	
-0-	Commits on Feb 28, 2019				
	Update README.md Verified Profharimohanpandey committed on 28 Feb 2019	Ŀ	e822f1b	<>	
	Update README.md Verified Sportharimohanpandey committed on 28 Feb 2019	C	5e274e5	<>	
	Update README.md Verified Verified Verified	C	cad78a4	\diamond	
	Update README.md Verified	D	3e7f4e1	<>	
	Update README.md Verified	C	d0735ee	<>	
	Update module-3.json Verified Verified	Ŀ	8c6cc53	\diamond	
	Update module-2.json Verified				
CW2-1 — Dolg	ohin Coursework2_WebApp — Mozilla C Commits - IacovosIacovides/CW2 C main.js - CW2-1 - Visual Studio Co	0 📮 (7a9e733	<> git version ☆	▲ 15:59
$CW2-1 - Dolp$ $Commits \cdot Iacove$ $COMMITS \cdot C = C$	optim Oursework2_WebApp — Mozilla Ommits · LacovosIacovides/CW2 M main.js - CW2-1 - Visual Studio Co osIacovide X Site overview iacovoscw2 X Email - IACOVOS IACOVIDE X G how to clone a repo from X Corrupt - How to recover github.com/IacovosIacovides/CW2-1/commits/master master •	0 📮 (7a9e733	c) igit version ☆	▲ 15:59
CW2-1 — Dolp	whin Oursework2_WebApp — Mozilla Commits • Lacovoslacovides/CW2 main.js - CW2-1 • Visual Studio Co oslacovide × Site overview iacovoscw2 × Email - LACOVOS LACOVIDE × G how to clone a repo from × corrupt - How to recover .c github.com/lacovoslacovides/CW2-1/commits/master master × Commits on Apr 21, 2022	10 - (1 × G	7a9e733	git versior	▲ 15:59
CW2-1 — Dolp	whin Coursework2_WebApp — Mozilla Commits • Lacovoslacovides/CW2 main.js - CW2-1 • Visual Studio Co oslacovide × Site overview iacovoscw2 × Email - LACOVOS LACOVIDE × G how to clone a repo from × corrupt - How to recover .c. github.com/lacovoslacovides/CW2-1/commits/master master • Commits on Apr 21, 2022 Created Time Slot Page and Degree Page	□ ■ (× G	1d16968	c>	▲ 15:59 × + () Updat
Commits - Lacove	oblacovide: Coursework2_WebApp — Mozilla oblacovide: Coursework2_WebApp — Mozilla oblacovide: Site overview iacovoscw2 × Imain is - CW2-1 - Visual Studio Co oblacovide: Site overview iacovoscw2 × Imain is - CW2-1 - Visual Studio Co oblacovide: Site overview iacovoscw2 × Imain is - CW2-1 - Visual Studio Co oblacovide: Site overview iacovoscw2 × Imain is - CW2-1 - Visual Studio Co oblacovide: Site overview iacovoscw2 × Imain is - CW2-1 - Visual Studio Co oblacovide: Site overview iacovoscw2 × Imain is - CW2-1 - Visual Studio Co oblacovide: Imain is - CW2-1 - Visual Studio Co Imain is - CW2-1 - Visual Stud	■ ■ (× G	Tage733	⇒ git version ★	 15:59 x + Updat
Commits - Jacovi	obla Coursework2_WebApp — Mozilla Commits • lacovoslacovides/CW2 main.js - CW2-1 • Visual Studio Co oslacovide: X Site overview iacovoscw2 X Email • LACOVOS LACOVIDE: X So how to clone a repo from < X	Q ➡ (× G	Ta9e733 Image: Ta9e733 Image: Ta9e733 Image: Ta9e733 how to update Image: Ta9e733 how to update Image: Ta9e733 1d16968 Image: Ta9e733 5af4afb Image: Ta9e733	Git version ★	15:59
Commits - Jacove	hin Coursework2_WebApp — Mozilla Commits • lacovoslacovides/CW2 main.js - CW2-1 • Visual Studio Co oslacovide: X Site overview iacovoscw2 X Email • LACOVOS LACOVIDE: X how to clone a repo from < X	■	7.496733 2 2 Image: marked state st	Comparison of the second s	 15:59 × + Updat
Commits - Jacove	hin Coursework2_WebApp — Mozilla Commits • lacovoslacovides/CW2 main.js - CW2-1 • Visual Studio Co oslacovid: X Site overview iacovoscw2 X Email • LACOVOS LACOVIDE X how to clone a repo from < X	× 6	7.396733 2 2 1016968 1d16968 5af4afb bc6727a 6da3e32	 Q Q	 15:59 × + Updat
Commits - Jacove	hin Coursework2_WebApp — Mozilla Commits • lacovoslacovides/CW2• main.js • CW2•1 • Visual Studio Co oslacovide: X Site overview iacovoscw2 X Email • LACOVOS LACOVIDE: X how to clone a repo from < X		7.396733 Image: Constraint of the second s	 Q Q	15:59
Commits - Jacove	hin Coursework2_WebApp — Mozilla Commits • lacovoslacovides/CW2• main.js • CW2•1 • Visual Studio Co oslacovide: X Site overview iacovoscw2 X Email • IACOVOS IACOVIDE: X how to clone a repo from < X		7396733 1016968 1d16968 5af4afb bc6727a 6da3e32 da948a9	git version \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow	15:59
Commits - Jacove	hin Coursework2_WebApp — Mozilla Commits · lacovoslacovides/CW2 CM main.js - CW2-1 - Visual Studio Co balacovide × Site overview lacovoscw2 × Email - LACOVOS LACOVIDE × G how to clone a repo from × 2 corrupt - How to recover github.com/Lacovoslacovides/CW2-1/commits/master master - Commits on Apr 21, 2022 Created Time Slot Page and Degree Page lacovoslacovides commited 2 hours ago Commits on Apr 20, 2022 Created the Assessment page. lacovoslacovides commited 22 hours ago Continued working on the pages.Added images lacovoslacovides commited 22 hours ago Continued the html pages. I have added a table that reads for text boxes lacovoslacovides commited yesterday Commits on Apr 19, 2022 Created the pages of the application lacovoslacovides commited 2 days ago Edited the index.html page (changed the heading of the page) lacovoslacovides commited 2 days ago		7.396733 7.396733 1016968 1d16968 5af4afb bc6727a 6da3e32 da948a9 ae0e1d9	git version \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow	15:59
Commits - Lacove	hin Coursework2_WebApp — Mozilla O commits - lacovoslacovides/CW2 A mainjs - CW2-1 - Visual Studio Co stacovide: X Site overview iacovoscw2 X E mail - LACOVOS LACOVID: X E how to clone a repo from X & corrupt - How to recover a github.com/Lacovoslacovides/CW2-1/commits/master master - Commits on Apr 21, 2022 Created Time Slot Page and Degree Page lacovoslacovides committed 2 hours ago Commits on Apr 20, 2022 Created the Assessment page. lacovoslacovides committed 2 hours ago Continued working on the pages.Added images lacovoslacovides committed 2 hours ago Continued the html pages. I have added a table that reads for text boxes lacovoslacovides committed 2 hours ago Continued the html pages. I have added a table that reads for text boxes lacovoslacovides committed 2 was ago Created the pages of the application lacovoslacovides committed 2 days ago Edited the index.html page (changed the heading of the page) lacovoslacovides committed 2 days ago Continued the report(Added Sketches and Diagrams) lacovoslacovides committed 2 days ago		7.396733 7.396733 1016968 1016968 5af4afb bc6727a 6da3e32 da948a9 ae0e1d9 ee5a776	git version $\hat{\mathbf{x}}$ $\hat{\mathbf{x}}$ $\hat{\mathbf{x}}$ $\hat{\mathbf{x}}$ $\hat{\mathbf{x}}$ $\hat{\mathbf{x}}$	 15:59 x + Updat
Commits - Lacove	hin Cursework2_WebApp — Mozilla Site overview iacovoscw2 × Imain js- CW2-1 - Visual Studio Co stacovide: × Site overview iacovoscw2 × Imain js- CW2-1 - Visual Studio Co stacovide: × Site overview iacovoscw2 × Imain js- CW2-1 - Visual Studio Co stacovide: × Site overview iacovoscw2 × Imain js- CW2-1 - Visual Studio Co commits on Apr 21, 2022 Created Time Slot Page and Degree Page Imain js- CW2-1 - Visual Studio Co Commits on Apr 20, 2022 Created the Assessment page. Imacovostacovides committed 22 hours ago Continued working on the pages. Added images Imacovostacovides committed 22 hours ago Continued the thmi pages. I have added a table that reads for text boxes Imacovostacovides committed 24 hours ago Commits on Apr 19, 2022 Created the hages of the application Imacovostacovides committed 2 days ago Edited the index.html page (changed the heading of the page) Imacovostacovides committed 2 days ago Continued the treport(Added Sketches and Diagrams) Imacovostacovides committed 2 days ago Commits on Apr 18, 2022		7.396733 2 2 how to update how to update 1d16968 5af4afb bc6727a 6da3e32 da948a9 ae0e1d9 ee5a776	git version \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow \Rightarrow	 15:59 x + Updat
CW2-1 — Dol	Attin Occursework2_WebApp — Mozilla (③ Commits - Jacovosalacovides/CW2 (叭 main.js - CW2-1 - Visual Studio Co) Application Site overview lacovosav2 × 1 ④ Email-JacOVOS JACOVID: × 1 ⓒ how to clone a repo from: × 1 ④ corrupt - How to recover : aster - Commits on Apr 21, 2022 Created Time Slot Page and Degree Page Image: Imag		7.396733 7.396733 1016968 1016968 5af4afb bc6727a 6da3e32 ae0e1d9 ee5a776 6b9bc3d	Image: space of the space	 15:59 × + Updat

Figure 10: GitHub commits.

On the annotations above, the commits that have been pushed to the Git are provided. The oldest commits have been made by another developer three years ago until a new developer has been asked to enhance the application this year. Each commit has its identification code where someone can see the details of the commit. The details include the code that has been changed, the documents that have been changed, and many more. The repository's URL that the developer has been using is <u>https://github.com/lacovoslacovides/CW2-1.git</u>.

—					CW2-1	— Dolphin					\lor \diamond \otimes
$\langle \rangle$		्र Find 🖾 Previev	v 💽 Split 🔳	Control							
Places	> Home > Project	s > CIS2169 > cwt	wo > CW2-1								
Places Figure 1 Figure 1	> Home > Project	s > CIS2169 > cwt	wo > CW2-1	books.jpg module-1,json	classdiagram.xmi	degree.html	degree.jpg README.md	general.jpg	ggg.xmi	bome.html	Jacovides_L 24128341_CW2 (1).docx
_	21 Files (1.5 MiB)							_	0		6.3 GiB free
🗱 💻 cw:	2-1 — Dolphin								() ()	🗖 🔒 📄 🔶	▲ 15:00 🔲 ≡

The application is stored in a folder on the developer's personal computer.

Figure 11: Developers folder in Personal Computer.

The code for the application is stored in the files you can see on the screen capture above. The files with the blue icon are the ones that store the coding which gives the website its looks.

The coding itself has comments above each different section of it. This makes it simpler and tidier for someone to edit if needed. When the coding is nested the correct way and there are useful comments, then the experience is much easier.



Figure 12: Create Module coding.

The coding above is about the home page which is called "Create Module" when the application is being used. The help of comments makes the understanding of the coding much easier since it explains what is going on.



Figure 13: Create degree coding.

The coding above is about the degree addition page which is called "Create Degree" when the application is being used. The help of comments makes the understanding of the coding much easier since it explains what is going on.



Figure 14: Create Assessment coding.

The coding above is about the assessment addition page which is called "Create Assessment" when the application is being used. The help of comments makes the understanding of the coding much easier since it explains what is going on.

>			information.html - CW2-1 - Visual Studio Code	\sim \diamond \otimes
File	Edit Selection Vie	ew G	o Run Terminal Help	
. ∩	EXPLORER	JS mai	nja o degree html o assessmenthini o home html o information.html x	5 1
6				
Q				Martin Constant and a second
1	- directory		<pre><needb< td=""><td>Land and a second se</td></needb<></pre>	Land and a second se
2º	activitydiagram.xmi activitydiagram.xmi		<title>Coursework2_WebApp</title>	
Č	assessment.jpg		<pre>_ clink rel="stylesheet" href="styles.css"></pre>	
-b-				1986-
~				100 mm
RP	O degree.html		<pre>shader></pre>	
	S ceneral.iog			
	≅ ggg.xmi		vriju class="tonpay"	
			<pre>Create Hodule</pre>	
	Iacovides_1_2412834		<pre>a href="degree.html">hereite Degree </pre>	
	O information.html		a http://www.action.com/action/act	
	45 main2.is			
			<ing alt="stack of books" class="ing" src="general.jpg"></ing>	
	() module-3.json		Add Module name	
	README.md		noulue hate:	
	# styles.css # tablehome.is			
			Add Module ID Medule ID> Medule ID>	Γ
			<pre><input class="input" id="id" name="quantity" type="text"/>dr /></pre>	
			<pre><:ADD HODULE Start time> Start Time:</pre>	
			<pre><input class="input" id="start" name="price" type="time"/>db />sbr /> cl_ddd Movila and time >></pre>	
			End Time:	
			<pre><input class="" imminist="" input="" pice="" type="time nome"/><or></or><or></or><or></or><or></or><or></or><or></or><or></or><or< td=""><td></td></or<></pre>	
			How and the second se	
			<pre>sth>Hodule ID </pre>	
			 	
			td> <td< th=""><th></th></td<>	
			12:99	
0				
8				(i) There is an available update. ×
572	OUTLINE		14:00	Download Update Later Release Notes
•	> TIMELINE		(4) - (4)	
_₽ m	aster ⊖0.1: ⊗0≜0			Ln 39, Col 26 Spaces: 2 UTF-8 LF HTML 🛱 🗘
цĶ	🚞 CW2-1 — Dolph		刘 information.html - CW2-1 - Visual	🖳 💽 🔛 🔒 🖻 🚸 🔺 15:11 🗔 😑

Figure 15: Module Slots coding.

The coding above is about the slots page which is called "Module Slots" when the application is being used. The help of comments makes the understanding of the coding much easier since it explains what is going on.

On all of the screen captures above there is a line of code that says:

→ <input type="..." name="..." id="..." class="..."/>
</br>

This line of code is found throughout the application multiple times. What this does, it adds an input text box on the page where the user can add their data and then return it to the table below the input boxes.

There is also another line of code that is found in every code file that says:

→ <input type="button" value="Add ... +" onclick="AddRowX()" id="add">/>
</br>

This line of code is found in every coding file. This line is the one that produces a button below the input text boxes, where when the button is clicked, then a function in a JavaScript file adds that data in a new row in the table.

The main JavaScript file that I have used, holds all important functions that aid in adding the data from the input text boxes into the tables of each page. What this code does is that it takes the information from the text boxes, matches the data into the appropriate "id" and then returns the data to the table as a new row at the bottom of the table.



Figure 16: Main JavaScript file coding.

The coding that the developer has done for the tables of each page is a simple table code. The code that the developer uses is:

<table id="....." border="1" style="width: 100%;":

The "" part in the code is the one that is responsible for creating the table. The way that the program knows how to end the table is by using the same piece of code but with a "/" before the word **table.** Thus, it is written like this: "". Inside these two pieces of code, all the columns and rows go. The developer assigned a "**"** bracket which is the new row and then for the row to end the developer used "**"**. Then with "**>...../th>"** the developer assigned the table header element which are the cells with the bold text inside which are the table headers. For the rest of the rows the developer used "**...../td>** for the ordinary text in the cells. Below is an example of table coding on the page "Create Degree".

```
        > ctr>
        > Degree Name 
        > Degree ID 
        > Degree ID 
        > Degree ID 
        > Degree ID 
        > Degree Academic 
        > Degree
```

6.0 Deployment planning and practice.

When the coding has been finished, the developer will start testing whether everything is working the way it should, accurately and fast. The developer will connect his GitHub account to the system called "Netlify", where he can deploy the website and test it. The link to deploy the application is https://iacovoscw2.netlify.app/.



Figure 17: Netlify URL.

On the screen capture above, the home page of **Netlify** is provided. As you can see at the top of the page the developer's name is given along with the name of the application. For the developer to deploy the application, he needs to click on the URL which is given in the first small box of the page. When that URL is clicked it will run and open the application.

9			Coursework2_W	ebApp - Go	ogle Chrome				\lor \diamond \otimes
Deploys iacovoscw2 ×	Coursework2_WebApp ×	O GitHub		× +					
\leftrightarrow \rightarrow \mathbf{C} $\hat{\mathbf{e}}$ iacovoscw2.netlit	y.app								☆ 🕕 :
Teacher's Portal						Create Module	Create Degree	Create Assessment	Module Slots
Module Name:					4				
	h								
Academic:									
Learning Outcomes:									
Add Module +									
Module Name	Degree		Module ID	Hours	Credits	Academic		Learning Outcomes	
Fundamentals of UXD	Web Design and Deve	lopment	CIS2148	45	100	David Walsh		Learn about UXD	
Fundamentals of Web Coding	Web Design and Deve	lopment	CIS2152	50	100	Mohammed Awais		Learn about Web Coding	
Data Driven Design	Web Design and Deve	lopment	CIS 2168	50	100	Hari Padney	Lear	n about Data Driven Desi	gns
Server and Client-side Scriptin	g Web Design and Deve	lopment	CIS 2167	50	100	Tom Smith	Learn abo	ut Server and Client-side	Scripting

Figure 18: Deployed Application.

🐇 🌀 Coursework2_WebApp - Google C...

On the screenshot above, the deployed application is portrayed. As you can see there is a navigation bar with all of the major features that the application offers. Additionally, you can see that on the

🧕 🔜 🕄 🖾 🔒 🖻 🐠 🔺 16:12 🔲 ≡

left-hand side of the page there are some text boxes where the user can input data and then return them in the table below.

The developer also tested where all the code works the way it should. Testing can eliminate the chances of getting errors on the final product (testing, 2022). On the screenshots below, various tests will be given, concerning the table and whether the data from the input text boxes are returned the correct way.

9		(oursework2_W	/ebApp - Google Chrome		~
Coursework2_WebApp × 🕙 History	× O	Iacovoslacovi	des/CW2-1	× 🔷 Site overview iacovoscw2	× S Coursework2_WebApp × +	
\leftrightarrow \rightarrow C $($ iacovoscw2.netlify.app/home.html						☆ 🕕 :
Teacher's Portal				C	Create Module Create Degree Create Assessment	Information
Module Name: Test Module ID: Test123 Hours: 50 Credits:						
Academic: Test Learning Outcomes:						
Test Test	h					
Add Module +						
Module Name	Module ID	Hours	Credits	Academic	Learning Outcomes	
Fundamentals of UXD	CIS2148	45	100	David Walsh	Learn about UXD	
Fundamentals of Web Coding	CIS2152	50	100	Mohammed Awais	Learn about Web Coding	
Data Driven Design	CIS 2168	50	100	Hari Padney	Learn about Data Driven Designs	
Server and Client-side Scripting	CIS 2167	50	100	Tom Smith	Learn about Server and Client-side Scriptin	8
Test	Test123	50	100	Test	Test Test	

Figure 19: Test 1.

As you can see on the screen capture above, the developer had inputted some data in the text boxes on the left hand, and when he clicked on the "Add Module +" button, the data were returned as a new row in the table below.

🗱 💼 GW2-1 — Dolphin 👘 data — Dolphin 👘 Coursework2, WebApp - Google C... 刘 home.html - CW2-1 - Visual Studio... 📮 🛈 📮 👘 🔺 17:45 🗔 🚍



Figure 20: Test 2.

On the screenshot above the developer wanted to check whether the "date" feature was working the correct way and it did.



🐇 🚞 CW2-1 — Dolphin	🚞 data — Dolphin	📀 Coursework2_WebApp - G	下 CW2-1 : bash — Konsole	📢 assessment.html - CW2-1	🧕 📮 🚱 📮 🖿 🔒 🖻 🚸 🔺 18:29 🔲 ≡

Figure 21: Test 3.

On the screenshot above the developer wanted to check whether the "number" feature was working the correct way and it did.



🗱 🛅 CW2-1 — Dolphin 🗧 data — Dolphin 🥥 Coursework2_WebApp - G... 🕅 CW2-1 : bash — Konsole 🕺 assessment.html - CW2-1 - ... 🔟 🐺 🔞 💭 🕰 🔮 🐠 🔺 18:30 🔲 🚍

Figure 22: Test 4.

As you can see on the screen capture above, the developer had inputted some data in the text boxes on the left-hand and when he clicked on the "Add Assessment +" button, the data were returned as a new row at the bottom of the table.

•		Coursework2_Web	oApp — Mozilla Firefox			~
Coursework2_WebApp ×	+					
$\leftarrow \rightarrow C$	file:///home/ehustud	ent/Projects/CIS2169/Iacovides_I_24128341_C\	W2/CW2-1/information.html		☆	⊚ ≡
Teaching Portal			Create Module	Create Degree	Create Assessment	Module Slots
Module Name: TEST 2 Module ID: TEST 2 Start Time: 15:23 End Time: 15:45 Add Module +		h	G G	ENERAL		
Module N	Name	Module ID	Time Start	ing	Time Fin	ish
TEST	1	TEST 1	13:15		15:00	
TEST	2	TEST 2	15:23		15:45	

😜 Coursework2_WebApp — Mozilla ... 🏮 Coursework2_WebApp - Google C... 🍕 information.html - CW2-1 - Visual ... 🦉 🦉 🐺 🗈 🔒 🖻 🌵 🔺 14:41 🔲 😑

🗱 🚞 CW2-1 — Dolphin

On the screenshot above, the developer wanted to check whether the time features were working the way they should. He did two tests and both of them work the way they should.

As mentioned above, the URL link for viewing and using the application is https://iacovoscw2.netlify.app/.

7.0 Final Application (final output when opening the application)

Coursework2_WebApp × +								
← → C □ f	ile:///home/ehustudent/Projects/CIS2169/cwtwo	/CW2-1/home.ht	ml					⊚ ≡
Teacher's Portal				Create	Module Create I	Degree	Create Assessment	Module Slots
Module Name:								
Degree					1			
Degree:								
Module ID:								
Hours.				•		Z		
nours:				4				
a								
Credits:				A				
				2				
Academic:					1			
Learning Outcomes:								
Add Module +								
Module Name	Degree	Module ID	Hours	Credits	Academic	γ	Learning Outcom	es
Fundamentals of UXD	Web Design and Development	CIS2148	45	100	David Walsh		Learn about UXI	
Fundamentals of Web Coding	g Web Design and Development	CIS2152	50	100	Mohammed Awais		Learn about Web Co	ding
Data Driven Design	Web Design and Development	CIS 2168	50	100	Hari Padney	L	earn about Data Driven	Designs
Server and Client-side Scripti	ng Web Design and Development	CIS 2167	50	100	Tom Smith	Learn a	about Server and Client	side Scripting
			•					
🐇 🚞 CW2-1 — Dolphin	🛛 🍪 Coursework2_WebApp — Mozilla 🧔 Em	ail - IACOVOS IACO	VIDES - Ou	itl			💽 🔜 💽 🖓 🖾 🖻 🚸	▲ 14:53 🔲 =

Figure 24: Final Create Module Page.

rsework2_WebApp × +				
→ C	ome/ehustudent/Proje	ects/CIS2169/cwtwo/CW2-1/degree	.html 🏠	\odot
eaching Portal			Create Module Create Assessmen	it Module Slo
gree Name:				
en a ID				
gree ID:				
gree Academic:				
arning Outcomes:				
ards:				
ards: d Degree + Degree Name	Derree ID	Deurse Academic	Learning Outcomes	Awards
ards: d Degree + Degree Name Web Design and Development	Degree ID CIS1111	Degree Academic Dave Walsh	Learning Outcomes Front end and Back end Development	Awards
ards: Degree • Degree Name Web Design and Development Computer Science	Degree ID CIS1111 CIS2222	Degree Academic Dave Walsh Mike Mike	Learning Outcomes Front end and Back end Development Be able to read and understand programs written by others.	Awards
ards: a Degree • Degree Name Web Design and Development Computer Science Sport Science	Degree ID CIS1111 CIS2222 CIS3333	Degree Academic Dave Walsh Mike Mike Mark Mark	Learning Outcomes Front end and Back end Development Be able to read and understand programs written by others. Body Anatomy and sports knowladge	Awards

Figure 25: Final Create Degree Page.

۵		Coursework	2_WebApp — Mozilla Firefox			~
Coursework2_WebApp × +						
← → C D file	e:///home/ehustudent/Projects/CIS	2169/cwtwo/CW2-1/asse	ssment.html		☆	5
Teaching Portal			Create Module Create	Degree Creat	te Assessment	Module Slots
Module Name:						
Assossment Title.						
issessment fille:			ACCECCA			
ssessment Number:			KODEDDIVI			
0			, icorour i			
				and the second s		
ssessment volume:				1		
ssessment weight:				. k		
dd / mm / yyyy						
Add Assessment +						
Module Name	Title	Number	Learning outcomes	Volume	Weight	Date
Data Driven Design	Task Data	1	How to collect data	20	20	02/02/2022
Data Driven Design	Task Data Filter	2	How to filter collected data	20	20	12/02/2022
Data Driven Design	Task Data Usage	3	How to use collected data	20	20	22/02/2022
Data Dalaran Daalam	Show data	4	How to Chow collected data	20	40	2/02/2022

Figure 26: Final Create Assessment Page.

🐇 🚞 CW2-1 — Dolphin

•	Coursewo	rk2_WebApp — Mozilla Firefox			~
Coursework2_WebApp ×	+				
$\leftarrow \rightarrow C$	D file:///home/ehustudent/Projects/CIS2169/cwtwo/CW2-1/inf	ormation.html		ដ	⊚ ≡
Teaching Portal Module Name: Module ID: Start Time: End Time: Add Stots +	*	Create Module	Create Degree	Create Assessment	Module Slots
	Module Name	Module ID	Time Starting	Time	Finish
	Fundamentals of UXD	CIS2168	12:00	1	4:00
Fu	ndamentals of Web Coding	CIS2152	14:00	1	6:00
	Data Driven Design	CIS 2168	11:00	1	3:00
Ser	ver and Client-side Scripting	CIS 2167	12:00	1	7:00
τ <u></u>					

🍐 Coursework2_WebApp — Mozilla ... 🌀 Email - IACOVOS IACOVIDES - Outl...

🔯 록 🔃 📮 🔒 🖻 🚸 🔺 14:53 🗔 ≡

🔟 🔜 🚱 💭 🔒 🖻 🐠 🔺 14:53 🗔 😑

Figure 27: Final Module Slots Page.

🗱 🚞 CW2-1 — Dolphin

8.0 Analysis Post-mortem

🍅 Coursework2_WebApp — Mozilla ... 🏼 🌍 Email - IACOVOS IACOVIDES - Outl...

In this section what went well during the development of the application and what went wrong will be specified. Moreover, if there was anything that the developer would do differently next time will be specified as well.

At the beginning of the code designing and implementation, the developer was very confident about what he was doing. The application was coming out the way it should. There was one major aspect that was hard for him though. When he started reading the case study, he could not understand what diagrams he should create. However, after some more reading and researching, he finally understood what to do. After designing the diagrams, the coding was easy to do except for some areas where the use of JavaScript files made it a bit confusing. With some appropriate research, he managed to finish the code resulting in a properly and accurately working system.

When the code was almost done, the repository found a corrupt file which made it impossible for the developer to push the changes into GitHub. This meant that, even though he further developed the code, he could not commit to any changed because of that file. After talking to the tutors, he then created a clone of his local repository into another folder which in the end fixed the error. This obstacle though wasted a lot of time for the developer. In the end, he managed to fix it and commit to changes again and end up pushing into GitHub and automatically into Netlify for deploying the website. Next time the developer will be more careful of what he is doing to assist in eliminating the chances of such a corrupt file emerging again and damaging the repository.

For the next time, the developer would surely do more research, since better research means that the designing of diagrams, sketches, and the code will be much more accurate and easy to do.

9.0 Conclusion

After taking into consideration everything that has been detailed above, along with the screenshots and explanations, the developer has done honest work when it came to developing this project. He has done some research that helped him to design diagrams that were needed and also, he managed to implement documents of code that led to a full, accurate working application that has no errors. As mentioned at the beginning of this report the developer should enhance an already existing application by improving several things that the application is lacking. The result is a basic fully working application that helps instructors and tutors add modules, assessments, degrees, and time slots to the modules.

10.0 References

Bluescape, 2019. *BlueScape*. [Online] Available at: <u>https://www.bluescape.com/</u> [Accessed 22 April 2022].

Contributor, C., 2020. *SmallBussines.Chron.* [Online] Available at: <u>https://smallbusiness.chron.com/</u> [Accessed 22 April 2022].

testing, S., 2022. *ToolsQA*. [Online] Available at: <u>https://www.toolsqa.com/</u> [Accessed 22 April 2022].