

2025

FORMATIVE ASSESSMENT 1

ADVANCED DESIGN PATTERN

ADP631

2025



CTU
training solutions

Creative &
Technology
Universitas

TABLE OF CONTENTS

DECLARATION OF AUTHENTICITY	3
QUALIFICATION DETAILS	4
Learning Outcomes	5
Assessment Criteria	5
Material Requirements	5
PROJECT BRIEF.....	6
Required Models & Properties:	6
Required Controllers:	7
Phases	8
EVIDENCE REQUIREMENTS.....	10
ASSESSMENT RUBRIC	12
RUBRIC MARK ALLOCATION	16


DECLARATION OF AUTHENTICITY

I, Jaydon Ellick
(FULL NAME)
hereby declare that the contents of this assessment ADP631_FA1_2025

are entirely my own work, with the exception of the following items listed below.

- **Internet: All assets, including inspiration, and references sourced from the internet and used in the final project, must be declared in this Declaration of Authenticity document.**
- List the elements of work in this project **that were not self-generated**, as well as the originator of the element.

Source	Artwork/Document Title
https://assetlifecycle.trimble.com/en/products/software/cityworks	TrimbleCityWorks
https://www.accela.com	Accela
https://www.municipalsoftware.com/cityview-suite/	CityView

Signature: 

Date: 2025/04/01

QUALIFICATION DETAILS

Programme Title	BI
Module Title	Advanced Design Pattern
Module Code	ADP 631
NQF Level	6
Credits	20
Assessment Type	Formative Assessment, Practical
Duration	2 Weeks
Total Marks	100
Hand Out Date and Time	20 March 2025 - 08:00
Submission Date and Time	04 April 2025 – 23:30

Student Name and Surname	
Student Number	
Percentage	
Status	

Learning Outcomes

- Getting Started with ASP.NET Core
- Understanding ASP.NET Core
- Your First Application
- ASP.NET Core Empty(MVC)

Assessment Criteria

- Preliminary Work
- Website Design and Database design
- Website Development
- Model, Controller and View
- CRUD operations
- Testing and Debugging
- Final Presentation

Material Requirements

- Text Editor/Coding Platform.
- Access to books, articles, and online resources for inspiration and information.
- Paper, pencils, markers, and erasers for initial wireframe and concept sketches.

PROJECT BRIEF

You have been given the task to use ASP.NET Core MVC to create a Municipality Management System as a prospective student of software development. By offering features for staff administration, reporting, service requests, and citizen registration, this system will aid in streamlining municipal services. You worked with the ASP.NET Core MVC template last year; this year, you have to build the project from scratch using an empty ASP.NET Core Web application, which calls for a better comprehension of project implementation and structuring.

From database integration and architectural design to testing and deployment, this project will walk you through every step of the development process. You will guarantee a well-organized and useful web application that is suited to municipal requirements by putting your understanding of MVC principles to use. The project is broken up into several stages to allow for methodical development.

Please note: As this project focuses on technical skill and execution using ASP.NET Core in c#, learners are allowed to source images and media for use in the website from online sources, excluding website icons or designs of any kind.

Required Models & Properties:

1. Citizen Management Model

- **CitizenID** (int, primary key, auto-increment)
- **FullName** (string, required)
- **Address** (string, required)
- **PhoneNumber** (string, required)
- **Email** (string, unique)
- **DateOfBirth** (DateTime, optional)
- **RegistrationDate** (DateTime, default: current date)

2. Service Request Model

- **RequestID** (int, primary key, auto-increment)
- **CitizenID** (int, foreign key referencing Citizens)
- **ServiceType** (string, required)
- **RequestDate** (DateTime, default: current date)
- **Status** (string, default: Pending)

3. Staff Management Model

- **StaffID** (int, primary key, auto-increment)
- **FullName** (string, required)
- **Position** (string, required)
- **Department** (string, required)
- **Email** (string, unique, required)
- **PhoneNumber** (string, required)
- **HireDate** (DateTime, required)

4. Reports Model

- **ReportID** (int, primary key, auto-increment)
- **CitizenID** (int, foreign key referencing Citizens)
- **ReportType** (string, required)
- **Details** (string, required)
- **SubmissionDate** (DateTime, default: current date)
- **Status** (string, default: Under Review)

Required Controllers:

1. CitizenController

- Manage CRUD operations for citizens.
- Actions: Index(), Create(), Edit(id), Delete(id), Details(id).

2. ServiceRequestController

- Manage citizen service requests.
- Actions: Index(), Create(), UpdateStatus(id), Delete(id), Details(id).

3. StaffController

- Manage municipal staff members.
- Actions: Index(), Create(), Edit(id), Delete(id), Details(id).

4. ReportsController

- Handle citizen-submitted reports.
- Actions: Index(), Create(), ReviewReport(id), Delete(id), Details(id).

Phases

Phase 1 - Preliminary Work (15 Marks)

- Research on existing municipality management systems.
- Define system requirements and functionalities.
 - Based on your research, list out what features your system should have.
- Create a project proposal document
 - Clearly define your system's objectives (What will it achieve?).
 - Set the scope (What will be included and what won't?).
 - List the features (What can users do in the system?).
 - Explain why your system is useful and how it improves existing solutions.
- **Deliverable:** Project proposal with defined scope and objectives.
- **Documentation Required:**
 - Overview of municipality management systems.
 - System requirements document.
 - Proposal document.

Phase 2 - Database Design (15 Marks)

- Identify key entities (e.g., Citizens, Services, Staff, Reports).
- Develop an Entity-Relationship Diagram (ERD).
- Define database schema using SQL Server.
- **Deliverable:** Database schema and ERD documentation.
- **Documentation Required:**
 - List of entities and attributes.
 - ERD diagram.

Phase 3 - Project Setup & Development (35 Marks)

- Create an empty ASP.NET Core MVC project in Visual Studio.
- Set up project structure (Controllers, Models, Views).
- Develop CRUD functionalities for municipality services.
- Implement data validation and error handling.
- **Deliverable:** Fully developed web application.
- **Documentation Required:**
 - Explanation of project structure (folders and files).
 - Code snippets of controllers and models with explanations.
 - Screenshots of working features.

Phase 4 - Testing & Debugging (15 Marks)

- Unit testing for controllers and services.
- Integration testing for database connectivity.
- Debugging and fixing identified issues.
- **Deliverable:** Test case reports and bug fixes.
- **Documentation Required:**
 - Test plan and cases.
 - Bug reports and resolutions.
 - Screenshots of test results..

Phase 5 - Documentation & Final Report (20 Marks)

- Prepare a final project report covering all development phases.
- Create a user Manual document
- Prepare a project presentation showcasing the website's features project and its functionality.

EVIDENCE REQUIREMENTS

- Completed Pre-Assessment form saved as ***yourname_ADP631_FA1_PA_2025.pdf***.
- Completed Declaration of Authenticity form saved as ***yourname_ADP631_FA1_DoA_2025.pdf***.

The final project is to be submitted in a single PDF file saved as ***yourname_ADP631_FA1_2025.pdf*** and structured as follows:

- Cover Page with the following information:
 - Name and Surname
 - Student Number
 - Module Name
 - Year
 - Formative Assessment 1
- Preliminary Work
 - Overview of municipality management systems(research).
 - System requirements document.
 - Proposal document.
- Database Design
 - List all entities ant their attributes and make sure they are in 3NF:
 - List all relationships between the entities
 - Attach the screenshot of the Entity Relationship Diagram
- Project Setup & Development
 - Step by step guide of how you created the project
 - List all nuggets packages installed
 - Code snippets of controllers, models, DBContext, AppSettings and Program.cs with explanations.
 - Screenshots of working features.
- Testing and Debugging
 - Documentation of website testing with testing plan and cases.
 - Screenshots or notes on debugging and final refinements.

- Finalization and Presentation
 - Final Website Version:
 - The completed, polished website with all required elements.
 - Presentation Slides:
 - A digital presentation explaining design choices, technical implementations, and user experience considerations.

Additional Evidence:

- Digital Presentation of website functionality:
 - A zipped project file demonstrating the website's functionality must be submitted to the respective lecturer.
 - A link to the project's GitHub repository should be included in the accompanying PDF document.

All the required evidence listed above will be saved in a single multiple-page PDF document named ***yourname_AD631_FA1*** uploaded to CampusOnline by the submission date.

- Final PDF to be compressed maintaining press quality.

Submissions not adhering to the above specifications will not be assessed.

ASSESSMENT RUBRIC

Program Title	BI
Module Title	Advanced Design Pattern
Module Code	ADP631
Total Marks	100
Assessment Due Date	04 April 2025
Student Number	
Percentage	
Status	
Marker & Signature	

Preliminary Work (15 Marks)					
Category	Poor (0-1/5)	Insufficient (2/5)	Sufficient (3/5)	Exceeds (4/5)	Exceptional (5/5)
Overview of Municipality Management Systems (Research) (5 Marks)	No research provided or lacks relevance.	Minimal research with unclear sources or explanations.	Basic research with some supporting details.	Well-researched with relevant details and proper explanations.	Comprehensive and well-documented research with examples and citations.
	/5				
System Requirements Document (5 Marks)	No system requirements document or incomplete.	Basic requirements listed but lacks detail.	Covers key requirements but lacks depth.	Well-structured requirements document with minor gaps.	Clearly defined requirements with complete details and justifications.
	/5				
Proposal Document (5 Marks)	No proposal or poorly written.	Basic proposal with unclear objectives.	Proposal covers essential aspects but lacks refinement.	Well-organized proposal with clear objectives and feasibility analysis.	Professional and detailed proposal with strong justifications and formatting.
	/5				
Total	/15				

Database Design (15 Marks)					
Category	Poor (0-1/5)	Insufficient (2/5)	Sufficient (3/5)	Exceeds (4/5)	Exceptional (5/5)
List all entities and attributes in 3NF (5 Marks)	Entities not listed or poorly structured with redundancy.	Basic entities listed but not normalized to 3NF.	Entities in 3NF but missing some attributes or explanations.	Well-structured entities and attributes with minor normalization issues.	Complete, well-structured entities in full 3NF with proper explanations.
	/5				
List all relationships between the entities (5 Marks)	No relationships identified or incorrect.	Some relationships listed but unclear or incorrect.	Most relationships identified but missing details.	Clearly defined relationships with only minor gaps.	Well-documented relationships with proper explanations.
	/5				
Attach a screenshot of the ER Diagram (5 Marks)	No ER diagram provided or unreadable.	ER diagram provided but lacks clarity or key elements.	ER diagram includes most required elements but is not well-organized.	Clear ER diagram with proper relationships and entity attributes.	Well-structured, professional ER diagram with complete details.
	/5				
Total	/15				

Project Setup & Development (35 Marks)					
Category	Poor (0-1/5, 0-3/10, 0-5/15)	Insufficient (2/5, 4-6/10, 4-7/15)	Sufficient (3/5, 7-8/10, 8-10/15)	Exceeds (4/5, 9/10, 11-13/15)	Exceptional (5/5, 10/10, 14-15/15)
Step-by-step guide of how you created the project (5 Marks)	No guide provided.	Minimal steps documented, lacks clarity.	Basic steps listed but missing details.	Clear steps with well-explained processes.	Comprehensive, well-structured guide with screenshots.
	/5				
List all NuGet packages installed (5 Marks)	No packages listed or incorrect.	Some packages listed but lacks justification.	Most packages listed with brief explanations.	Well-documented package list with usage explanations.	Complete and detailed package list with purpose for each.
	/5				
Code snippets of controllers, models, DbContext, AppSettings, and Program.cs with explanations (10 Marks)	No code provided or completely incorrect.	Some code snippets provided but lacks explanations or has errors.	Most required code provided but with incomplete explanations.	Well-documented code with minor missing details.	Clean, well-documented, fully functional code with clear explanations.
	/10				
Fully functional app and screenshot of output (15 Marks)	Non-functional app with major errors or missing output.	Basic functionality present but many bugs or missing features.	Mostly functional app with minor bugs. Screenshot provided.	Well-functioning app with only minor issues. Clear screenshot of output.	Fully functional, polished app with all features working. Clear and professional screenshot.
	/15				
Total	/35				

Testing & Debugging (15 Marks)					
Category	Poor (0-3/10, 0-1/5)	Insufficient (4-6/10, 2/5)	Sufficient (7-8/10, 3/5)	Exceeds (9/10, 4/5)	Exceptional (10/10, 5/5)
Documentation of website testing with test plan and cases (10 Marks)	No testing plan or test cases documented.	Minimal testing documentation with missing details.	Basic test cases documented but lacks depth.	Well-structured test cases with minor gaps.	Comprehensive testing plan with detailed test cases and results.
	/10				
Screenshots or notes on debugging and final refinements (5 Marks)	No debugging documentation or final refinements.	Basic debugging notes with missing key points.	Some debugging documentation with screenshots.	Well-documented debugging process with detailed refinements.	Fully documented debugging process with screenshots and explanations.
	/5				
Total	/15				

Finalization and Presentation (20 Marks)					
Category	Poor (0-3/15, 0-1/5)	Insufficient (4-6/15, 2/5)	Sufficient (7-10/15, 3/5)	Exceeds (11-13/15, 4/5)	Exceptional (14-15/15, 5/5)
Final Website Version (15 Marks)	Website is incomplete or broken.	Basic website functionality present but missing key features.	Functional website with some refinements needed.	Well-polished website with only minor issues.	Fully completed, visually appealing, and polished website.
	/15				
Presentation Slides (5 Marks)	No slides provided or lacks content.	Basic slides with minimal information and poor structure.	Sufficient slides with relevant details but lacks design refinement.	Well-structured slides with clear explanations and good design.	Professional, engaging, and well-designed presentation with strong explanations.
	/5				
Total(20)	/15				

RUBRIC MARK ALLOCATION

No.	Instruction Evidence	Marks
1	Preliminary Work	/15
2	Database Design	/15
3	Project Setup & Development	/35
4	Testing & Debugging	/15
5	Documentation & Final Report	/20
Grand Total:		/100
Percentage:		
Status:		
Assessment Feedback:		