|  |  |
| --- | --- |
| **Course code and name:** | F21MP – Masters Project and Dissertation |
| **Type of assessment:** | **Individual** |
| **Coursework Title:** | Master’s Dissertation |
| **Student Name:** | Kawthar Mohammad Adam |
| **Student ID Number:** | H00399169 |

|  |
| --- |
| **Declaration of authorship. By signing this form:**   * **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section. * I confirm that I have read, understood and followed the University’s Regulations on plagiarism as published on the [University’s website](https://www.hw.ac.uk/uk/students/studies/examinations/plagiarism.htm), and that I am aware of the penalties that I will face should I not adhere to the University Regulations. * I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on [Academic Integrity and Plagiarism](https://heriotwatt.sharepoint.com/sites/skillshub/SitePages/Academic-Integrity-and-Plagiarism.aspx)   **Student Signature***: Kawthar Adam*  **Date**: *27/11/2024* |

|  |
| --- |
| COURSEWORK WIZARD: DEADLINES VISUALISER |
|  |
| Author: |
| Kawthar Adam |
| H00399169 |
|  |
| Supervisor: |
| Dr. Hind Zantout |
|  |
|  |
| ***A thesis submitted in fulfilment of the requirements for F21MP*** |
| ***in the*** |
|  |
| **School of Mathematical and Computer Sciences** |
|  |
| **November 2024** |
|  |
|  |

# Abstract

This dissertation addresses the challenge of managing coursework deadlines, a common issue among students that often results in stress and late submissions. Learning Management Systems (LMS) lack effective tools for task prioritisation and progress tracking. To solve this, Coursework Wizard was developed as a web-based platform combining LMS functionalities with project management techniques like Kanban boards and visual progress tracking. The system helps students manage their tasks incrementally, focusing on the process rather than just deadlines, while providing lecturers with tools to monitor submissions and track student progress. The platform was built using Next.js and tested through usability evaluations, including heuristic evaluations and System Usability Scale (SUS) surveys. The findings indicate that while the platform is intuitive and effective in managing coursework, it faces limitations in functionality and scalability. This dissertation outlines the development process, system evaluation, and identifies areas for future improvements, such as mobile compatibility, automated feedback, and advanced analytics. The proposed system aims to reduce the stress of deadline management for students while preparing them for real-world task management.

**Keywords: Coursework, Deadlines, Visualisation, Usability, Time Management, LMS.**

# Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Hind Zantout, for her invaluable expertise and guidance throughout this dissertation. She was always available to provide support and advice, playing an important role not just as a supervisor but also as a potential client for this project. Without her cooperation, this project would not have been possible to complete.

I am also deeply thankful to my family and friends for their support, encouragement, and belief in me throughout my academic journey. Their constructive criticism has been a constant source of motivation. Special thanks to my sister Sonia and my friend Samrah for helping me design the website, learn the technologies, and prioritise my tasks.

# Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

Table of Figures vii

List of Tables xii

List of Abbreviations xiii

Chapter 1. Introduction 1

1.1 Project Background 1

1.2 Aims and Objectives 2

1.3 Report Outline 2

Chapter 2. Literature Review 4

2.1 Pedagogical Background 4

2.1.1 Historical Evolution of Coursework 4

2.1.2 Definition and Significance of Coursework in Education 4

2.2 Current Trends and Challenges in Coursework Management 5

2.2.1 Trends in Educational Technology 5

2.2.2 Analysis of Coursework Submission Patterns 6

2.2.3 Challenges in Coursework Management 6

2.3 Usability and User Experience in Educational Platforms 9

2.3.1 Importance of Usability in Education 9

2.3.2 Usability Testing and Evaluation 10

2.3.3 Importance of User Experience in Education 12

2.4 Coursework Management Systems 13

2.4.1 Overview of Learning Management Systems (LMS) 13

2.4.2 Types and Examples of LMS in Education 14

2.4.3 Perceived Usability and UX of Learning Management Systems (LMS) 15

2.5 Coursework Deadline Visualisation and Management 16

2.5.1 Coursework Management Strategies for LMS 17

2.5.2 Coursework Visualisation Strategies for LMS 20

2.5.3 Self-Management Strategies to Meet Deadlines 22

2.6 Education 4.0 23

Chapter 3. Requirements Analysis and Methodology 25

3.1 System Specifications Using the MoSCoW Method 25

3.1.1 Functional Requirements and MoSCoW Prioritisation 25

3.1.2 Non-Functional Requirements 25

3.1.3 MVP Goals and Additional Features 26

3.1.4 Features Planned but Not Implemented 27

3.2 Software Development Methodology 28

Chapter 4. Design and Implementation 32

4.1 Development Tools and Technologies 32

4.2 Personas and User Requirements 35

4.3 Prototyping and UI Design 36

4.4 System Design 39

4.4.1 System Architecture 39

4.4.2 Database Design 40

4.4.3 Key Design Decisions 42

4.5 Implementation 43

4.6 Assumptions 45

Chapter 5. Testing and Evaluation 47

5.1 Introduction 47

5.2 Testing 47

5.2.1 Testing Strategy 47

5.2.2 Unit Testing 47

5.2.3 Integration Testing 50

5.2.4 Heuristic Evaluation 50

5.3 Evaluation 52

5.3.1 Usability Testing 52

5.3.2 Staff Feedback 54

5.4 Summary of Findings 57

Chapter 6. Conclusion and Future Work 59

6.1 Potential for Industry Implementation and Future Work 59

6.2 Limitations 59

6.3 Conclusions and Final Reflections on the Project 60

References 62

Bibliography 73

Appendix A: Heuristic Evaluation 75

Appendix B: SUS Survey 77

Appendix C: Professional, Legal, Ethical and Social Issues 78

Appendix D: Prototype 79

Appendix E: Final Website 98

Appendix F: Questionnaires 111

Appendix G: Testing Documents 113

Appendix H: Signed Consent Forms 118

Appendix I: Evaluation Results 130

Appendix J: JIRA Sprint Planning 149

Appendix K: JIRA Sprint Retrospective 154

Appendix L: JIRA Graphs 163

Appendix M: Personas 166

# Table of Figures

Figure 3.1 The Scrum Methodology [83] 28

Figure 3.2 JIRA Backlog 29

Figure 3.3 JIRA Kanban Board for Sprint 9 29

Figure 3.4 Sprint Flowchart 31

Figure 4.1 Running Coursework Wizard in development mode 32

Figure 4.2 Visual Studio Code 33

Figure 4.3 MySQL Workbench for designing and creating the ERD 33

Figure 4.4 MySQL Workbench for creating and managing the database 33

Figure 4.5 Postman for testing the APIs 34

Figure 4.6 Running Coursework Wizard in development mode 34

Figure 4.7 Paper Prototype 37

Figure 4.8 Coursework Wizard Logo (a) Old (b) New 38

Figure 4.9 Rejected Designs for Timeline (a) Design 1 (b) Design 2 38

Figure 4.10 Coursework Wizard Architecture 40

Figure 4.11 Next.js Architecture [87] 40

Figure 4.12 Entity-Relationship Diagram (ERD) for Coursework Wizard 41

Figure 4.13 Submission Table 42

Figure 5.1 Usability Test Plan 53

Appendix B – Figure 1 The standard SUS Survey [88] 77

Appendix D – Figure 1 Rejected Prototype in Editor 79

Appendix D – Figure 2 Rejected Prototype 79

Appendix D – Figure 3 Final Prototype in Editor 80

Appendix D – Figure 4 Home Page 80

Appendix D – Figure 5 Help Page 81

Appendix D – Figure 6 Contact Us Page 81

Appendix D – Figure 7 Contact Us Submission Confirmation 82

Appendix D – Figure 8 Student Login 82

Appendix D – Figure 9 Staff Login 1 83

Appendix D – Figure 10 Staff Login 2 83

Appendix D – Figure 11 Forget Password 1 84

Appendix D – Figure 12 Forget Password 2 84

Appendix D – Figure 13 Forget Password 3 85

Appendix D – Figure 14 Forget Password 4 85

Appendix D – Figure 15 Password Reset Confirmation 86

Appendix D – Figure 16 Student Dashboard 86

Appendix D – Figure 17 Student Courses 87

Appendix D – Figure 18 Student Course Page 87

Appendix D – Figure 19 Student Coursework View Page 88

Appendix D – Figure 20 Student Timeline 88

Appendix D – Figure 21 Subtask Modal on Student Timeline 89

Appendix D – Figure 22 Student Kanban Board 89

Appendix D – Figure 23 Student Account 90

Appendix D – Figure 24 View Student Profile 90

Appendix D – Figure 25 Student Settings 91

Appendix D – Figure 26 Student Notifications 91

Appendix D – Figure 27 Student Contact Us 92

Appendix D – Figure 28 Student Contact Us Confirmation 92

Appendix D – Figure 29 Student Help Page 93

Appendix D – Figure 30 Staff Dashboard 93

Appendix D – Figure 31 Staff Course Page 94

Appendix D – Figure 32 Staff Coursework View Page 94

Appendix D – Figure 33 Staff Add New Coursework 1 95

Appendix D – Figure 34 Staff Add New Coursework 2 95

Appendix D – Figure 35 Staff Add New Coursework 3 96

Appendix D – Figure 36 Staff Add New Coursework Confirmation 96

Appendix D – Figure 37 Students List 97

Appendix D – Figure 38 View Individual Student Submission Status 97

Appendix E – Figure 1 Home Page 98

Appendix E – Figure 2 Help Page 98

Appendix E – Figure 3 Contact Us Page 99

Appendix E – Figure 4 Contact Us Confirmation Page 99

Appendix E – Figure 5 Student Login 99

Appendix E – Figure 6 Staff Login 1 100

Appendix E – Figure 7 Staff login 2 100

Appendix E – Figure 8 Forgot Password 1 100

Appendix E – Figure 9 Forgot Password 2 101

Appendix E – Figure 10 Forgot Password 3 101

Appendix E – Figure 11 Forgot Password 4 101

Appendix E – Figure 12 Password Rest Confirmation 102

Appendix E – Figure 13 Student Dashboard 102

Appendix E – Figure 14 Student Courses 102

Appendix E – Figure 15 Student Course Page 103

Appendix E – Figure 16 Student Coursework View Page 103

Appendix E – Figure 17 Student Locked Subtaaks 103

Appendix E – Figure 18 Student Timeline 104

Appendix E – Figure 19 Student Timeline Modal Subtask View 104

Appendix E – Figure 20 Student Kanban Board 104

Appendix E – Figure 21 Student Notifications 105

Appendix E – Figure 22 Student Account 105

Appendix E – Figure 23 Student View Profile 105

Appendix E – Figure 24 Student Edit Profile 106

Appendix E – Figure 25 Student Settings 106

Appendix E – Figure 26 Student Rewards 106

Appendix E – Figure 27 Student Contact Us 107

Appendix E – Figure 28 Student Help Page 107

Appendix E – Figure 29 Staff Dashboard 107

Appendix E – Figure 30 Staff Course Page 108

Appendix E – Figure 31 Staff Coursework View Page 108

Appendix E – Figure 32 Staff Coursework Deletion Confirmation 108

Appendix E – Figure 33 Staff Add Coursework 1 109

Appendix E – Figure 34 Staff Add Coursework 2 109

Appendix E – Figure 35 Staff Add Coursework Review 109

Appendix E – Figure 36 Staff Coursework Added Confirmation 110

Appendix E – Figure 37 Staff View Subtask Timeline Modal 110

Appendix E – Figure 38 Students List 110

Appendix F – Figure 1 System Usability Scale (SUS) Survey 111

Appendix F – Figure 2 Post-Usability Questionnaire 112

Appendix G – Figure 1 Task Sheet for Students 1 113

Appendix G – Figure 2 Task Sheet for Students 2 114

Appendix G – Figure 3 Information Sheet for Students 1 115

Appendix G – Figure 4 Information Sheet for Students 2 116

Appendix G – Figure 5 Consent Form for Participants 117

Appendix H – Figure 1 Consent Form 1 118

Appendix H – Figure 2 Consent Form 2 119

Appendix H – Figure 3 Consent Form 3 120

Appendix H – Figure 4 Consent Form 4 121

Appendix H – Figure 5 Consent Form 5 122

Appendix H – Figure 6 Consent Form 6 123

Appendix H – Figure 7 Consent Form 7 124

Appendix H – Figure 8 Consent Form 8 125

Appendix H – Figure 9 Consent Form 9 126

Appendix H – Figure 10 Consent Form 10 127

Appendix H – Figure 11 Consent Form 11 128

Appendix H – Figure 12 Consent Form 12 129

Appendix J – Figure 1 Sprint 1 Planning – Figma Prototype 149

Appendix J – Figure 2 Sprint 2 Planning – Figma Prototype Cont. 149

Appendix J – Figure 3 Sprint 3 Planning – Figma Prototype 150

Appendix J – Figure 4 Sprint 4 Planning – Front-End Development 150

Appendix J – Figure 5 Sprint 5 Planning – Front-End Development Cont. 151

Appendix J – Figure 6 Sprint 6 Planning – Database 151

Appendix J – Figure 7 Sprint 7 Planning – Student Back End 152

Appendix J – Figure 8 Sprint 8 Planning – Staff Back End 152

Appendix J – Figure 9 Sprint 9 Planning – Staff Testing and Report 153

Appendix K – Figure 1 Sprint 1 Retrospective 154

Appendix K – Figure 2 Sprint 2 Retrospective 155

Appendix K – Figure 3 Sprint 3 Retrospective 156

Appendix K – Figure 4 Sprint 4 Retrospective 157

Appendix K – Figure 5 Sprint 5 Retrospective 158

Appendix K – Figure 6 Sprint 6 Retrospective 159

Appendix K – Figure 7 Sprint 7 Retrospective 160

Appendix K – Figure 8 Sprint 8 Retrospective 161

Appendix K – Figure 9 Sprint 9 Retrospective 162

Appendix L – Figure 1 JIRA Backlog 163

Appendix L – Figure 2 JIRA Kanban Board 163

Appendix L – Figure 3 JIRA Cumulative Flow Diagram for Coursework Wizard 164

Appendix L – Figure 4 JIRA Burn Up Chart for Sprint 9 164

Appendix L – Figure 5 JIRA Burn Up Chart for Sprint 8 164

Appendix L – Figure 6 JIRA Burn Down Chart for Sprint 8 165

Appendix M – Figure 1 Student Persona 1 166

Appendix M – Figure 2 Student Persona 2 166

Appendix M – Figure 3 Student Persona 3 167

Appendix M – Figure 4 Student Persona 4 167

Appendix M – Figure 5 Staff Persona 1 168

Appendix M – Figure 6 Student Persona 2 168

Appendix M – Figure 7 Student Persona 3 169

# List of Tables

Table 2.1 Submission rates of coursework [25] 9

Table 3.1 Functional requirements 25

Table 3.2 Non-functional requirements 26

Table 3.3 Comparative analysis of features planned and implemented 27

Table 3.4 Sprint breakdown 30

Table 5.1 Unit Test Cases 48

# List of Abbreviations

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| AR | Augmented Reality |
| ARIA | Accessible Rich Internet Applications |
| BBA | Bachelor of Business Administration |
| BCM | British Computer Society |
| BI | Business Intelligence |
| CPS | Cyber Physical System |
| CS | Computer Science |
| CSS | Cascading Style Sheets |
| CSV | Comma Separated Values |
| CW | Coursework |
| DB | Database |
| ERD | Entity Relationship Diagram |
| FAQ | Frequently Asked Question |
| GPA | Grade Point Average |
| GPT | Generative Pre-training Transformer |
| GUI | Graphical User Interface |
| HCI | Human Computer Interaction |
| HE | Heuristic Evaluation |
| HTML | Hyper Text Transfer Protocol |
| HW | Heriot-Watt |
| IDE | Integrated Development Environment |
| IEEE | Institute of Electrical and Electronics Engineers |
| IoT | Internet of Things |
| IR | Industrial Revolution |
| ISO | International Organisation for Standardisation |
| JSX | JavaScript XML |
| LA | Learning Analytics |
| LAD | Learning Analytics Dashboard |
| LMS | Learning Management System |
| MOOC | Massive Open Online Course |
| MVC | Model View Controller |
| MVP | Minimum Viable Product |
| OOP | Object-Oriented Programming |
| OOUX | Object-Oriented UX |
| OS | Operating System |
| PJ | Project Journal |
| SDLC | Software Development Life Cycle |
| SMS | Short Message Service |
| SQL | Structured Query Language |
| STEM | Science, Technology, Engineering and Mathematics |
| SUS | System Usability Scale |
| TBD | To Be Decided |
| TMT | Temporal Motivation Theory |
| UI | User Interface |
| UX | User Experience |
| VR | Virtual Reality |
| VS | Visual Studio |
| WAF | Web Application Firewall |
| WBS | Work Breakdown Structure |
| XR | Mixed Reality |

# Chapter 1. Introduction

## Project Background

Coursework is a form of assessment that challenges students to apply knowledge in solving real-world problems through assignments, reports, dissertations, and more [1]. This practical approach has shown increased academic performance and satisfaction among students. With technology rapidly spreading globally, the education sector has introduced digital platforms such as learning management systems (LMS) to further assist students in their academic pursuits. LMS serves as e-classrooms where students can access study materials and submit assignments from home. Despite the many advantages of LMS, students often struggle to manage time and meet coursework deadlines. Statistics reveal that 50% of students delay starting coursework, and 75% submit their work within the last 48 hours, leading to heightened stress and late submissions [2] [3].

As coursework is a critical part of the degree, failure to meet deadlines can result in poor grades and subsequent depression. A Swedish study of 6,146 participants indicated that 20% of students aged 19-33 reported suicide thoughts and 3% attempted suicide. Upon further research, a positive relation was discovered between suicide attempts and school performance after considering factors like family background, social conditions, drug addiction, and health behaviours [4]. Researchers have addressed this issue by observing student behaviour through data analysis from LMS to uncover patterns. Upon evaluation, researchers found that delayed coursework is mainly due to a lack of self-control and time mismanagement [5]. Strategies to motivate students to start coursework early have been suggested, including setting proper deadlines, gamification, dashboards, task prioritisation, work breakdown structures, and time management. These strategies have shown positive results in experiments using various technologies alongside LMS. Research claims that integrating such self-regulation strategies into learning can help manage time [6].

While several LMSs exist in the market, they lack pedagogical usability and effective features for managing coursework deadlines, leading to increased stress and last-minute submissions. To address these challenges, this project proposes Coursework Wizard, a website designed to help students manage deadlines more effectively by incorporating proven strategies from experts in computer science and psychology. The website is inspired by Canvas, the LMS used at Heriot-Watt University, which has the highest usability score among all LMSs [6]. Coursework Wizard helps students break down their coursework into small, manageable milestones and summarises deadlines in an interactive, Gantt chart-like calendar. Students can track their progress visually through gamification modules that motivate them and provide real-time feedback on their performance. This incremental approach reduces last-minute anxiety and allows students to review their work before submission, improving both the quality of their work and their grades. The site also ensures academic integrity, as the step-by-step progress allows for clear evidence of the student’s own work. Ultimately, Coursework Wizard provides a comprehensive tool for managing deadlines, staying on track, and reducing stress, ensuring students can complete coursework on time with a higher degree of ownership and confidence.

## Aims and Objectives

The aim of this dissertation is to develop an engaging and usable website to help students and lecturers effectively manage and visualise up to four coursework deadlines. The system will integrate self-regulation and time management strategies in order to motivate students to start and finish coursework earlier. The primary objectives of the project are as follows:

* Visualise student progress to monitor performance and manage time
* Incorporate gamification modules to motivate students
* Provide regular reminders to maximise student engagement
* Display student submission statistics to lecturers
* Allow staff to track student performance and progress
* Conduct tests to evaluate usability and refine the website

## Report Outline

The organisation of the subsequent chapters in this document is as follows:

* Chapter 2: Constitutes the literature review to provide background on coursework submission and reasons for delays. Moreover, it also explores learning management systems and usability evaluation techniques.
* Chapter 3: Discusses the system requirements and prioritises them using MoSCoW. It also outlines the Solo Scrum methodology used for development.
* Chapter 4: Covers the design process, system architecture, tools, prototypes, and the implementation of the website.
* Chapter 5: Describes the testing strategies, including unit and integration testing, heuristic evaluation, and feedback from users.
* Chapter 6: Summarises the project outcomes, challenges, and suggests future improvements and potential industry applications.

# Chapter 2. Literature Review

To encourage timely coursework submissions, it is crucial to understand the importance of coursework, the reasons behind late submissions, the role of learning management systems, and the significance of usability. This chapter will look into these aspects to grasp the aims and objectives of this dissertation. Every section concludes with a brief paragraph critically analysing the previous works and connecting it with the website developed.



## Pedagogical Background

### Historical Evolution of Coursework

Around 50 years ago, Michael Bassey introduced formal assignments, or coursework, to pedagogy after completing his teacher training programme. The programme used a combination of coursework and examinations for assessments, and 98% of students were satisfied with this twofold approach. Bassey favoured it for its positive impact on stress reduction and improved performance [7]. In 1977, Derek Rowntree and John Heywood introduced alternatives to unseen exams, such as open-book, pre-released, and essay exams, to better assess students' strengths, weaknesses, and interests [8] [9]. They noted that while two people may perform the same action, their experience and motivation can differ significantly [10]. In 1985, it weighed 34% of the total marks and increased to 79% in 1994. By 1996, coursework became widespread across the UK [9].

### Definition and Significance of Coursework in Education

Coursework refers to assessments completed over a longer period, either individually or collaboratively, such as assignments, dissertations, reports, and class assessments [9] [1]. It helps develop time management and teamwork skills, fostering collective responsibility and interaction with industry experts [11].

Coursework has positively impacted academic performance, with the percentage of first-class degrees in the UK rising from 39% to 68% between the 1950s and 2013 [9]. A 2024 study found that replacing exams with coursework worth 60% of the final grade contributed to grade inflation and increased the distribution of top-ranked degrees [12]. This suggests that coursework, whether used alone or with exams, leads to higher marks due to its collaborative nature, whereas exams, though resistant to AI manipulation, often encourage cramming that doesn’t reflect professional practice [13] [14]. Furthermore, coursework has long-term benefits, as students engage consistently over time, demonstrating broader abilities and developing strategic thinking. A Norwegian study found that 70% of students applied their coursework knowledge in their professional lives [15].

However, the lack of invigilation in coursework opens the door to plagiarism and contract cheating. While anti-plagiarism systems can detect some fo rms of cheating, they cannot identify AI-generated coursework. For instance, when [16] injected 100% AI-produced coursework into a UK-based university’s portal, 94% of the submissions went undetected, and 83.4% outperformed real student submissions. Unsurprisingly, unsupervised online exams have also become targets of similar malpractices [9] [14].

In summary, while coursework enhances student performance and skill development, effective management is crucial to prevent cheating and ensure academic success.

## Current Trends and Challenges in Coursework Management

### Trends in Educational Technology

As mentioned by [9], academic bodies had started to shift to online examinations for essay-type questions, backed with overwhelming favour from older students. The usage of technology was not a surprise as many researchers predicted its permanent effect on education in 1966, stating that it will be an integral part of every child’s life in the future [17].

The demand for online courses increased as technology advanced and the number of students with heterogeneous knowledge grew. As a result, universities underwent a digital transformation, introducing blended and online teaching. With online and blended learning, students can self-regulate their learning and submit coursework remotely at any time within the assigned dates. However, this degree of autonomy often leads to procrastination, with many students postponing submission until the last moment or engaging in plagiarism, which makes online courses more challenging [18] [2]. Despite the availability of free Massive Open Online Courses (MOOCs), most students fail to complete these courses and often drop out. Engagement rates, content retention, and learning outcomes are typically lower in MOOCs compared to traditional learning methods. These issues can be addressed by using Learning Analytics (LA) [19].

### Analysis of Coursework Submission Patterns

Academic deadlines motivate students and combat procrastination, but they also lead to negative outcomes due to deadline rush [20] [21]. According to [22], 62% of students experience moderate stress over deadlines, with 19% experiencing severe stress. Short deadlines require continuous engagement whereas longer deadlines require students to work independently. As deadlines approach, students engage more in surface learning to gain temporary knowledge rather than deep learning [21]. This often results in a hyperbolic curve where submissions are clustered near deadlines and are more likely to be of poor quality [23].

To understand submission patterns, researchers analyse data from platforms like Learning Management Systems (LMS) and Massive Open Online Course (MOOC) dashboards [24]. For example, [3] found that 50% of students intentionally delay assignments until the last 24 hours, resulting in lower grades. A comparative study by [2] and [25] found second- and third-year students' submissions clustered near deadlines, with experienced students managing deadlines better. Statistically, 75% of third-year students made submissions within the last 48 hours, while 74% of second-year students submitted work in the last 24 hours.

[23] noted that deadlines scheduled before weekends increase procrastination more than those after weekends. Submissions made between 11 pm and 6 am were more likely to contain errors, indicating poor time management [2]. [26] found that fixed deadlines boosted student performance (mean ≈ 89), while self-imposed deadlines were less effective (mean ≈ 86).

To summarise, while online study offers flexibility, it often leads to procrastination and last-minute submissions, which result in poor grades and increased stress. Students need proper deadline scheduling and time management to avoid last-minute stress.

### Challenges in Coursework Management

Digital ubiquity has increased students' addiction to social media, leading them to avoid work and engage in unproductive activities. A study of 758 students in Mexico and Spain found a problematic positive correlation between excessive Internet use and procrastination [25]. Lack of self-control leads to poor prioritisation, reduced academic seriousness, higher anxiety, and diminished performance. With limited time to complete assignments, students lose motivation due to fear of failure, leading to depression. They often rationalise their poor performance by blaming time management issues and a lack of interest in the coursework [5].

1. Psychological Factors

Students struggle to focus on academics due to procrastination, which constitutes 80-95% of work issues such as unfinished assignments and missed deadlines. Since COVID-19, online learning has led to a swift surge in this trend with around 70% of university students engaging in moderate procrastination, and 14% being chronic procrastinators. Procrastination is linked to poor time management, motivation, anxiety, and perfectionism, and results in negative academic outcomes, stress, fear of failure, and mental distress [5].

In [27], procrastination is defined as the voluntary delay of intended actions despite knowing the potential consequences. Similarly, [28] define it as a habit of postponing tasks until they become too difficult to complete on time. [29] differentiates procrastination from postponement by stating that in procrastination, there is no guarantee when the task will be performed, and this could result in years of negligence. Procrastination negatively impacts lifestyle, as shown in a Swedish study where 344 students with mild procrastination experienced anxiety and depression, while others with severe procrastination showed intense psychological symptoms [25]. The delay leads to feelings of guilt and restlessness, and as deadlines approach, students regret the delay, preferring earlier deadlines [29].

Procrastination can be caused by the following:

1. Temporal discounting: Temporal discounting refers to prioritising short-term rewards, like the immediate joy from social media, over delayed future rewards such as good exam marks, leading students to procrastinate [27] [30] [21].
2. Longer deadlines: Longer deadlines can improve performance but may also cause students to overlook tasks, leading to late submissions, especially if deadlines are set before the required content is taught [31] [2]. According to Parkinson’s Law, a mismatch between task length and deadlines can lead to procrastination, as extended deadlines often discourage early completion [5].
3. Time mismanagement: Time mismanagement leads to poor self-control, as students prioritise leisure activities over coursework and panic as deadlines approach, often resorting to ineffective strategies like plagiarism, collusion, or using generative AI tools [5] [23].
4. Self-regulation failure: Self-regulation involves understanding one’s behaviour to achieve goals, while emotional intelligence means controlling emotions to guide actions. When students fail to act on their intentions due to seeking short-term rewards, cognitive dissonance develops, leading to procrastination. About 47% of students procrastinate online due to their inability to self-regulate [27].
5. Lack of motivation: Low self-esteem, fear of failure, or self-distrust decreases self-efficacy, causing students to question their ability to complete tasks and eventually avoid attempting them. Motivation may decrease further in online settings due to the absence of peer pressure [5].
6. Underestimation of time: Being overly optimistic can backfire when students fail to accurately assess task complexity, leading to poor planning and wasted time [27].
7. Perfectionism: It causes students to focus on flawless outcomes, avoiding tasks out of fear of imperfection, which results in procrastination [5].

Students engage in either passive or active procrastination. All the attributes discussed earlier apply to passive procrastinators. Active procrastinators, on the other hand, intentionally delay submissions until they are pressured by deadlines for motivation. However, [27] and [29] argue that using procrastination as a coping strategy does not benefit students. Despite limited research, it has been observed that active procrastinators score higher than passive ones, but the risk of self-handicapping and failure exists [3].

1. External factors

Students may struggle to complete and submit online coursework due to a lack of resources. Common challenges include unstable internet connections and faulty devices, which can demotivate students and reduce the quality of their work. As technology becomes more pervasive, any disruption in access can negatively impact education. External factors such as work or family responsibilities, poor health, financial constraints, and emotional distress can also hinder academic progress [5]. Some students experience the "over-doer" phenomenon, where they overcommit to tasks with unrealistic timelines, leading to anxiety and missed deadlines. These delays differ from procrastination, as they are caused by external factors rather than psychological ones [5].

1. Academic Factors

[25] observed that assignment submission rates are influenced by academic factors, particularly the type of assignment, as shown in Table 2.1. They noted that students prefer working on presentations followed by projects and written assignments. Presentations are engaging and interactive, while writing tasks are often seen as daunting, leading to higher rates of non-submission, and long-term projects are delayed due to commitment challenges. The study found that difficult or unenjoyable coursework tends to be completed and submitted late. Students were more likely to procrastinate on both very easy and highly challenging tasks.

Table 2.1 Submission rates of coursework [25]

|  |  |  |  |
| --- | --- | --- | --- |
|  | On-time (%) | Delayed (%) | Not submitted (%) |
| Presentations | 24.5 | 7.4 | 1.5 |
| Written Assignments | 18.3 | 12.4 | 2.7 |
| Projects | 9.1 | 23.6 | 0.6 |

Additionally, students often delay assignments they perceive as insignificant or beyond their competence. Many online learners lack the necessary prerequisites, leading to demotivation and missed deadlines [5]. These factors can be explained by the Temporal Motivation Theory (TMT), which suggests students are more likely to engage in tasks they enjoy or find beneficial. The theory defines utility (the willingness to engage in a task) as , where E is the probability of success, V is the value of the task, D is the delay between completion and reward, and Γ is the student's sensitivity to the delay [32].

In conclusion, students face challenges in meeting coursework deadlines due to several factors. Passive procrastination and forgetfulness are common, as students seek short-term mood repair to avoid work. They may also forget submissions when deadlines are distant or when the necessary topics haven't been covered in class. To address these issues, students need a system that reminds them of deadlines and facilitates the start of their tasks.

## Usability and User Experience in Educational Platforms

Educational technology has evolved drastically in the past years, becoming a crucial part of learning. This has caused designers and developers to create software for everyday users and not just for tech savvies, leading to the concepts of user experience and usability in education. Moreover, as suggested by software psychology, educational technology needs to be evaluated using usability evaluation techniques to ensure its effectiveness [33].

### Importance of Usability in Education

The term usability (formerly known as user-friendliness) was coined in the 1980s, but researchers failed to define it because it depends on varying factors and cannot be treated as a property of one entity [34]. Usability is generally defined by objective (performance) and subjective (satisfaction) outcome measures [35]. The International Organisation for Standardisation (ISO) defines usability as the extent to which a system can satisfy specific users in a specific context by helping them achieve goals effectively and efficiently, where satisfaction refers to the positive attitude toward the system, effectiveness means achieving goals accurately, and efficiency measures the resources used [33].

Nielsen provides a more specific definition, with parameters to evaluate the ease of use of an interface:

1. Learnability: Ease of learning a system for the first time
2. Efficiency: Resources and time needed to complete tasks
3. Memorability: Ease of remembering how to use a system after a break
4. Error Rate: Reduced errors and ease of recovery
5. Satisfaction: Comfort of using the system [36]

Despite efforts to define usability, existing standards often fail to fully apply to educational technologies because they overlook pedagogical and sociocultural factors [33]. Pedagogical usability, crucial for the acceptance of educational technology, includes content, multimedia, tasks, social interaction, and personalisation. If an LMS is difficult to navigate, learners may spend more time figuring it out than engaging with content, limiting learning outcomes [37] [33]. According to [38], a learner's intention to use a platform depends on perceived ease of use, with complex systems increasing anxiety and cognitive load, thus hindering academic performance. They distinguish between technical usability and pedagogical usability, emphasising how readability and ease of use impact learning outcomes. In pedagogical usability, it is recommended that system design mimic user learning behaviour by using familiar schemas, avoiding unnecessary features, and minimising distractions and anxiety.

### Usability Testing and Evaluation

Evaluating usability is difficult but measuring difficulties incurred while using the system is easy. There is an inverse relationship between ease of use and the number of difficulties faced. Identifying and quantifying these problems can help determine the usability, and the most common method for this is usability testing [34].

In usability testing, an observer watches typical users interact with the system to collect quantitative data on usage problems and effectiveness. Users can also be asked to Think Aloud, where they verbalise their thoughts while using the system, providing qualitative data on the learning process. This is often followed by a survey to measure usability and user satisfaction, which can either be developed by the user or standardised [34].

1. Heuristic Evaluation (HE)

Usability evaluation methods aim to identify problems and maximise a system's ease of use, which is crucial for educational platforms to ensure a smooth learning experience and improve academic performance. When [38] evaluated ElectronixTutor, an Intelligent Tutoring System for electronics, students reported that poor design disrupted their learning. Issues included a mismatch between the user interface (UI) and students' mental models, and poorly visible navigation components. Students suggested the following improvements:

* Change the button terminology: The button to submit a question was labelled "Submit Your Answer," confusing students who wanted to ask a question, not submit an answer.
* Move progress bar: The progress bar was hard to find, located under a drop-down menu in the top-right corner. Students expected it on the left side with the course content.
* Improve agent graphics: Eye-tracking revealed students focused more on the tutor agent than the content. Students found the agent’s jerky graphics distracting.

These suggestions align with usability heuristics. In Heuristic Evaluation (HE), evaluators interact with the system and compare it to a list of usability principles, called heuristics. The most well-known heuristics were introduced by Nielsen and Molich in 1990, after evaluating 249 usability issues [39]. According to Nielsen, 3-5 expert evaluators can identify up to 87% of usability issues, while novice evaluators typically detect only 23%, due to a lack of understanding of the abstract heuristics. To help train novice evaluators, [40] developed a detailed version of Nielsen's heuristics (see Appendix A).

HE is commonly used because it takes around two hours, is easy to use, cost-effective, and applicable to both complete and incomplete systems. The evaluation time can be reduced further with more evaluators [39].

To ensure Coursework Wizard is highly usable and provides a smooth learning experience, avoiding the usability issues found in ElectronixTutor is essential. Coursework Wizard will undergo Heuristic Evaluation (HE) using Appendix A to identify potential usability issues. The results of this evaluation will be documented, providing insights into the system's interface and interaction flow. This will help confirm whether Coursework Wizard meets the usability standards necessary to enhance student engagement and academic performance.

1. System Usability Scale (SUS) Survey

A System Usability Scale (SUS) is a standardised psychometric tool developed by John Brooke in 1986, widely used for reliable usability evaluation across various sample sizes. Around 43% of studies use SUS to assess subjective perceptions of a system's usability [34]. In a study by [41], students were divided into five teams to evaluate software using their preferred usability tool. Three teams chose SUS because it is short, customizable, easy to calculate, and effective for comparing systems, and it encourages honest participant feedback. SUS is especially suited for evaluating educational systems, as it also focuses on learnability.

The SUS survey consists of ten statements with alternating positive and negative tones, rated on a five-point Likert scale (1 = Strongly Disagree, 5 = Strongly Agree, and 3 = neutral) (Appendix B). To calculate the final score out of 100:

1. For positive statements, subtract 1 from the scale position (x-1).
2. For negative statements, subtract the scale position from 5 (5-x).
3. Sum the ten values.
4. Multiply the total by 2.5 [34]

Interpretation of the SUS score varies. According to [34], a score above 51 is “Okay”, 72 is “Good”, and 85 is “Excellent”.

SUS is versatile and applicable to all system types. When [37] analysed research papers on educational usability, they found the usage of SUS in all of them, attaining a mean SUS score of 63.30, similar to previous research with scores of 70.09 and 68. Educational multimedia was marked most usable with a mean score of 76.43, followed by mobile applications, affective tutoring systems, LMS, and lastly, university websites. They concluded that current educational usability is good but with some issues.

The SUS is a straightforward and effective tool for evaluating usability, and it will be used to assess Coursework Wizard. The goal is to achieve a score higher than 70.09, positioning Coursework Wizard as more usable than existing platforms.

### Importance of User Experience in Education

User Experience (UX) is the user’s feelings, emotions, and preferences associated with the system before, during, and after its use. It is concerned with the levels of satisfaction driven by the user’s needs and expectations [42] [43]. UX includes subjective factors that vary from one user to another, making usability a part of UX [44]. The UX of a system can be assessed on the content, usability, aesthetics, look and feel, functionality, as well as the sensual and emotional appeal [45]. Additionally, it is influenced by the user’s demographics, perceptions, cognition, experience, financial status, and usage context [44]. Therefore, UX can also be defined as the interaction between the user, system, and context of usage [45].

Lack of student motivation and engagement poses serious challenges to in-time coursework submissions. Engagement can be defined as empowering users, making learning meaningful, and enabling interface customisation [46]. This can also be supported by gamification strategies [47].

To effectively design for UX, experts use the following tools [48] [49]:

1. Personas: Fictional characters representing the target audience, developed early in the design phase to understand user needs. Personas can be data-driven or ad-hoc.
2. Scenarios: Stories about user activities and interactions with the product, clarifying the product's purpose and features.
3. Use cases: Step-by-step processes derived from scenarios that outline task completion.
4. Object-Oriented UX (OOUX): An approach based on object-oriented programming, where objects are identified, and actions are assigned to them.

Combining personas and scenarios enhances these tools. According to [48], scenarios transform personas from passive to active by providing context, situation, and objectives.

In the context of Coursework Wizard, understanding UX principles is crucial for designing a platform that effectively supports students' time management. By developing ad-hoc personas, the design process can better align with the specific needs and expectations of students. These personas will guide the creation of a user-friendly interface that motivates students, enhances engagement, and ultimately improves academic performance.

## Coursework Management Systems

### Overview of Learning Management Systems (LMS)

The development of Learning Management Systems (LMS) responded to the growth of online technology, providing an all-inclusive online classroom with features like course materials, gradebooks, professional training, and communication tools [50] [51]. LMS allows instructors and learners to distribute, share, store, and access learning materials over the Internet without time and location constraints, making information more accessible and reducing administrative costs [52].

The LMS market grew rapidly after the spread of the Internet and multimedia, with an average annual growth rate of 7.9% by 2013, peaking at 17% in some countries [53]. By 2018, 3500 institutions had adopted LMS [51], and usage increased significantly during COVID-19, with many institutions permanently adopting LMS [52].

LMS benefits both professors and students. Professors can use a variety of media, such as video, audio, images, and text, to support learning, track student performance, and distribute materials with ease. Students can enrol in classes, access materials, check grades, participate in discussions, take tests, and seek support from peers and professors [50].

A well-designed LMS should have centralised control for accessibility, self-service for tasks like enrolment, quick content creation and distribution, a secure environment, personalisation options, and integration with educational content [51].

When actively used, LMS helps students become more independent by providing constant feedback and additional resources like guides and assessments. However, studies show students often struggle with meeting deadlines. Researchers suggest implementing reward systems to motivate students, helping them stay on track, and maintain focus [50] [52].

### Types and Examples of LMS in Education

1. Canvas

Canvas, developed by Instructure, is used by 17.1% of American institutions and over 3000 universities worldwide. It competes strongly with Blackboard, especially after Blackboard acquired Angel LMS [53] [51]. Canvas is accessible via computers and mobile devices, allowing flexible participation in timed assessments and instant feedback for quizzes [14]. It integrates with open-source tools like Google Docs for collaborative learning. Features include drag-and-drop course creation, student progress tracking through Canvas Analytics, rubric creation, and grading with SpeedGrader [51] [6].

Canvas consistently receives high satisfaction and usability scores, with faculty praising its ease of course creation, file uploads, assignments, and grading. Students appreciate the modular organisation of content and the ability to engage in discussions via boards or groups. Study participants found that clear goals, timely feedback, and active discussions with instructors increased user satisfaction and cognitive presence [53].

1. Moodle

Moodle is an open-source, cost-free LMS used by 19.4% of institutions across 241 countries, with over 291 million users [53] [54]. It allows teachers to exchange files, conduct real-time discussions, and use a digital whiteboard. Though similar to Blackboard, Moodle’s key distinction is its free cost [44]. Despite updates, users report UX issues with speed, content organisation, search functions, and navigation. The mobile app is especially criticised for its poor communication features, leading some users to avoid it for courses like philosophy [54] [44]. While Moodle offers tools for content organisation, assignment submissions, and grading, instructors often underutilise these features [6].

1. Blackboard Learn

Founded in 1997, Blackboard ranks highest in the number of users. It has more than twelve million users and 33.5% of educational institutions use it. It gained popularity after acquiring communication tools and live tutorials. [53]. Blackboard and Moodle share many features, but teachers find Blackboard’s announcement page more effective for reaching students [6]. A usability survey comparing Blackboard and Moodle showed that Blackboard was easier to learn and more satisfying to use, although some students reported higher response times [53]

In terms of preference, Blackboard is favoured for assignments and gradebooks, Moodle for assignments, and Canvas for quizzes, polls, syllabi, and tests. Satisfaction ratings are “C” for Canvas and Blackboard, and “D” for Moodle [6].

### Perceived Usability and UX of Learning Management Systems (LMS)

Academic performance improves when LMS are usable and provide a productive UX [53]. While LMS share similar functionalities, they differ in user-friendliness, customisation, cost, requirements, and institutional needs [55]. Researchers have identified factors influencing the usability and UX of LMS. For example, [53] lists seven factors impacting user satisfaction on Blackboard: consistency, clear terminology, feature overload, informed location, simplicity, visible hyperlinks, and help sections. Additionally, response time, reliability, and accessibility equally influence LMS usage intention alongside usability.

However, [47] found that 50% of users are dissatisfied with LMS due to limited features, outdated UX, poor customer support, complexity, lack of agility, and inadequate reporting. These issues were categorised as design and managerial issues, although [45] argue that managerial issues do not impact UX.

[42] found that students attending online lectures were dissatisfied with the system, receiving poor grades due to system interruptions and connectivity issues. To improve UX, the following enhancements were suggested:

* Support service: Addressing software and hardware issues (e.g., slow transmission, microphone problems) and providing prompt customer support.
* Interactive communication: Allowing students to split screens to engage in both tasks and lectures simultaneously.
* Ease of use: Ensuring security, compatibility, and user-friendliness across devices.
* Learning resources: Including more diverse resources and activities to enhance engagement.

LMS provides a digital learning environment for users to access learning materials and submit coursework. The system makes learning independent for students, making it difficult for them to manage assignments and meet deadlines. Therefore, LMS must be usable. In comparisons, Canvas was preferred over Moodle and Blackboard, which faced usability issues. Analysing these shortcomings provides insight into improving systems like Coursework Wizard to enhance usability and UX.

## Coursework Deadline Visualisation and Management

Throughout the academic year, students engage in several different activities outside school hours that suppress the skills of managing course load. Poor time management causes students to complete less than 80% of their assignments, which results in poor grades. Students who complete more assignments tend to receive higher grades [56]. To help students manage their time, experts have introduced management and visualisation tools and techniques which include processes, frameworks, concepts, trends, visuals, and exercises to meet project requirements [57] [58]. The concept of coursework management is inspired by project management where the project manager organises, schedules, controls, and monitors the project and team to achieve the goal efficiently. It is reported that project management improves the success rate of projects [59].

### Coursework Management Strategies for LMS

1. Deadlines

Imposing deadlines for coursework promotes time management but also increases stress and pressure [28]. Therefore, instructors must ensure that deadlines do not clash with other courses, allow sufficient time for completing the work, can be managed alongside other commitments, and promote student well-being [23]. By setting correct deadlines, students have a boundary against which they must work. The problem arises when deadlines are not set correctly, leading to counterproductive activities and last-minute submissions [29]. To avoid such issues, experts suggest that midnight deadlines scheduled at the end of the week encourage students to spread their work over the week to avoid last-minute stress. It is also advisable to avoid setting deadlines on Monday mornings as students often stay up late despite having work or school the next day [23].

Moreover, the frequency and duration of coursework also influence student performance. Having small but regular deadlines improves engagement, completion of work, and performance. However, if all courses adopt this approach, the likelihood of coinciding deadlines increases. Therefore, it is important to break down longer deadlines into smaller ones according to the module and start the deadline only after the required course material is taught. In the case of longer deadlines, instructors must monitor students to track progress and prevent collusion [2].

Lastly, according to [28], imposing significant penalties for late submission further decreases performance. They recommend a deadline policy based on their research, where instructors should impose minor penalties along with a strict midterm deadline or no penalties at all.

1. Rewards and Incentives

Providing students with incentives upon assignment submission can motivate them to work harder and meet milestones on time [31]. These rewards should complement coursework and be appealing to students [29]. In an experiment by [27], students engaged in two tasks: without rewards, and with rewards. Despite exhaustion after the first task, students performed better on the rewarded task. However, [31] noted that students are motivated by immediate rewards and that delayed rewards show no improvement in performance.

To promote timely coursework submission, students can receive rewards in various forms. Certificates for completing coursework are more rewarding than end-of-program certificates because they are received immediately, and not after 3-4 years. Furthermore, they are preferred over trophies as they hold recognition and value beyond university. When combined with deadlines, certificates improve grades and prevent blank submissions [31].

[31] observed that incentives did not impact low- and high-ability students. Low-ability students were unlikely to achieve high GPAs even with maximum effort and high-ability students were motivated only by monetary rewards. However, incentives are highly effective for average students who lack self-control. Deadlines and certificates can either help or hurt average students as most studies overlook them and focus on low- and high-ability students.

In summary, digital incentives like badges, trophies, certificates, and leaderboards can encourage students to increase engagement and submit work consistently. Competition on leaderboards may motivate them to start work early and avoid last-minute submissions.

1. Reminders

Long deadlines allow students to work on coursework according to their preferences and availability. However, this often leads them to forget the coursework deadline [2]. [32] developed a tool for sending automatic periodic situational awareness emails to students. These emails compare students' current progress with the due date and their peers' progress. They began a week before the deadline and were customised based on the student's most recent submission. The tool assesses the submission and rates it on a four-point scale: 'good' for students ahead of schedule, 'neutral' for those on track, 'bad' for those behind, and 'undefined' for those with insufficient information. The email subject included the course code along with the status indicator. These emails resulted in a 23% decrease in late submissions and a 31% increase in early submissions.

LMSs can share weekly reports with students about their actual and expected progress to encourage them to complete the work early. Moreover, the status indicator used by the tool will help inform the decision of defining the status of the progress bar in Coursework Wizard.

1. Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) is a project management technique where the multi-level project is broken down into smaller manageable activities for managing complex projects. The tasks are arranged in levels where each parent level has several detailed child tasks that need to be completed to complete the parent task. The breakdown can be represented using spreadsheets, flowcharts, lists, or Gantt charts. These representations provide a roadmap for the project where team members can focus on their tasks with an understanding of where and how their tasks fit into the picture. It shows the milestones, dependencies, and deliverables of the project as well [59]. There are two kinds of WBS:

* Deliverable-based WBS: It focuses more on the deliverables produced throughout the project by breaking down the project into deliverables needed, and therefore, the main product is placed at the top of the hierarchy with sub-tasks as children.
* Phase-based WBS: It focuses more on the project phases by breaking down the project into phases of the project lifecycle, and therefore, the final phase of the project is placed at the top of the hierarchy with sub-phases as children.

WBS helps track the project, and if the project falls behind schedule, the team can identify the deliverables that will be impacted the most and plan ahead. Studies show that WBS simplifies project management and helps in predicting project delivery [59].

Despite being a project management tool, WBS can be implemented in academics. Instructors can divide the coursework specification into smaller tasks, linking it with the lecture content covered. In this way, students will get a well-structured specification that is easy to follow. This technique is used to inform the assumption of lecturers uploading coursework that is already divided into weekly milestones.

1. Time Mapping

Fiore suggests that when students have too much or too little time, they tend to procrastinate. To prevent this, he introduced time mapping. Here, deadlines are set for each calendar day by dividing it into 60-minute slots. First, unavailable time slots, such as school hours, are crossed out. Next, tasks are assigned to the available slots, compelling students to complete the task within the designated time period. Furthermore, each productive task is followed by a recreational activity as a reward [60].

Time mapping consists of three components: scheduling, unscheduling, and logging. In scheduling, the student reserves the available time slot for an important task, prioritising it over other activities. In unscheduling, the student adds a fun activity after the important task and highlights it using a bright colour of their choice. Bright colours boost people's moods and motivate them to work harder to achieve the reward, while colours like red signify danger, causing anxiety. Lastly, students log all their activities to analyse their time usage and identify peak times of high productivity for better planning [56].

Time mapping can also be implemented using digital schedule sheets, where students write tasks in plain text instead of using a digital calendar. However, this approach often leads to students overscheduling their time and creating unrealistic schedules [32].

The colour theory mentioned in this section will be used to inform the decision to avoid the colour red in Coursework Wizard as it represents danger and increases stress.

### Coursework Visualisation Strategies for LMS

1. Gamification

Gamification increases student motivation and engagement by integrating game design principles into non-game contexts. This concept was introduced by Nick Pelling in 2002 when he integrated gaming components like rewards into educational, fitness, and medical systems, leading to the emergence of serious games meant for serious purposes rather than entertainment. Gamification introduces fun elements into serious and boring tasks, motivating users to engage in behaviour and complete the activities [49]. In pedagogy, gamification captures the students’ attention and involves them in the learning process. What satisfies them the most is the excitement of using acquired knowledge to solve problems [61].

Gamification is a psychological strategy that involves three components [49]:

1. Motivation – “Why are we doing this?”
2. Mastery – “How are we doing this?”
3. Triggers – “When are we doing this?”

The most common and appreciated gamification element used across all systems for visualisation is the progress bar. In games, it represents the health bar, but in non-game systems, it encourages users to achieve their goals by visualising their progress. Upon completion, the brain releases endorphins, making the user feel happy and content. Failing to complete the activity may leave a sense of incompleteness, causing stress or restlessness [49]. This component is implemented in Moodle for teachers to record students’ progress and be alert to dropping engagement. However, students are not satisfied because the progress bar does not specify what it represents [61].

- progress bar image: origina picture for inspirtation in appendix

1. Learning Analytics Dashboard (LAD)

Visuals like shapes and patterns make it easier for humans to understand and analyse large volumes of data compared to textual information [62]. In 2015, Coursera had 15 million students of whom only 2.5 million completed courses. This was because students were unable to commit time, the course was poorly designed, or they had no prior knowledge. To help students manage their courses and understand their learning progress, Coursera suggested using data mining to produce visual analytics [63] [64].

LADs visualise students’ online data to represent their study habits, academic performance, and learning status. They assist students by visualising patterns and providing real-time feedback to motivate them [65]. Monitoring student data allows early detection of students with high tendencies toward procrastination and failure, with 97% accuracy [66]. These students can be easily identified, and instructors can assist them by providing extra homework, frequent feedback, and continuous monitoring [67]. Moreover, LADs track students' social networks and peer activities because upward social comparison with peers stimulates motivation and encourages students to work toward their goals [68]. In their study, [69] observed that LADs increase students' self-esteem, satisfaction, and enjoyment.

Course Signals is one of the many LADs developed for students and instructors. It collects data and presents it as a traffic light where red represents students at a higher risk of failing the course, yellow for medium risk, and green for low risk. This analysis is shared with the student through in-app notifications, emails, or SMS. Additionally, it uses bar graphs to display students' activities, line graphs for weekly trends, scatter charts for peer comparison, and sociograms for online networking [64].

1. Kanban

Kanban, meaning “sign” in Japanese, was developed by Toyota for lean management and later adapted into Agile software development by Microsoft. It is a visual technique for tracking projects by creating a board with three columns and moving cards between them:

* To do: Contains a long list of all the tasks that need to be completed
* Doing: 3-4 tasks from the “to do” are moved here when work starts on them
* Done: Completed tasks are moved to this column [70]

To ensure that time spent on tasks is efficiently utilised, the number of tasks in the “doing” column is limited. A Kanban board can be created using a physical whiteboard and sticky notes, or a specialised software [70]. Kanban improves communication within the team, problems are solved collaboratively, project completion time is reduced, and productivity is increased. Since each card is assigned to a specific member, it becomes relatively easy to follow up on tasks directly with the concerned person [71].

Kanban can be implemented in all kinds of projects and requires no training. It complements incremental development, allowing the integration of changes during the project. With Kanban, tasks are clearly defined to avoid wasting time on irrelevant tasks, thus reducing additional costs. However, Kanban supports project management but cannot be used independently [71]. Images

Gamification components notify students about their progress, LADs visualise student data by identifying patterns, and Kanban visualises work by breaking projects into discrete tasks. While Kanban is commonly used in professional projects, its implementation in academics helps students gain a clear understanding of their coursework and prevents missed deadlines. Therefore, Coursework Wizard incorporates a progress bar to visualise progress for each coursework, a rewards system for submitting subtasks, graphs to visualise statistics of students who have completed and not completed coursework, and a Kanban board to organise and prioritise subtasks.

### Self-Management Strategies to Meet Deadlines

Students can improve their academic performance and reduce procrastination through various techniques. One such method is the use of assignment logs or work journals, which help students track coursework data and monitor their progress. This practice increases self-awareness, enabling students to identify and reflect on counterproductive behaviours, leading to better time management and higher grades [56]. In 2024, Heriot-Watt University introduced project journals (PJs) for master’s students, requiring them to document their progress, challenges, and achievements in 2–3-week intervals. PJs align with Agile development practices, fostering incremental progress and helping students manage their work while preventing last-minute cramming and issues like plagiarism [72] [73].

Other techniques include the use of implementation plans, the Eisenhower Matrix, and the Pomodoro Technique. Implementation plans enhance self-regulation by bridging the gap between intentions and actions, although students often delay completing assignments until the final hours [30]. Breaking tasks into smaller, manageable subtasks is another effective approach to overcoming procrastination, as it makes tasks feel less daunting and helps students engage in goal-directed behaviour [27]. The Eisenhower Matrix prioritises tasks based on urgency and importance, helping students manage their workload and avoid last-minute stress [60]. Finally, the Pomodoro Technique involves breaking tasks into smaller subtasks and working in short, focused intervals. Students set a timer for 25 minutes, followed by a short 3–5-minute break. This cycle is repeated four times, after which students reward themselves with a longer break of 20 minutes [65] [70].

To combat procrastination and manage time, experts suggest a few techniques. Developers can integrate these techniques into learning platforms to maximise academic performance. By producing journals, students can keep track of their tasks and also provide evidence of their own work. By intending to work on coursework, students can get themselves to at least start the coursework and protect themselves from regrets of delaying work. The Eisenhower matrix will help them sort out tasks by preventing them from wasting time on unimportant and non-urgent tasks. Finally, the Pomodoro technique will motivate them to work on the task without interruption. These strategies can be integrated into LMS to limit distractions and encourage students to start working on their coursework earlier.

## Education 4.0

The revolution caused by technology was defined using the term Industrial Revolution (IR) which means the societal transition from manual work to automated work. There are 4 main IRs ranging from 1IR to 4IR, starting from the eighteenth century until today. The ongoing 4IR encompasses technologies like Artificial Intelligence (AI) and the Internet of Things (IoT), increasing human-machine interaction to boost performance and efficiency [74].

Soon, it will not be uncommon for robots to deliver lectures and solve mathematical problems using AI [75]. The job pool will face several changes as the world adopts 4IR technologies, disrupting the job market. The skills considered crucial now will be deemed unnecessary then, causing people to have several different jobs over their lifespan. According to studies, machines will advance to extreme AI levels by 2040, challenging the need to hire humans. The World Economic Forum states that 47% (75 million) of the jobs done today will be automated by 2030 and 65% (133 million) of new jobs will exist [75] [74].

To cope with such challenges, people need to be retrained so that they are equipped with the required STEM skills [76]. Therefore, the educational sectors have implemented Education 4.0 in response to align with Industry 4.0. In Education 4.0, education providers teach about technology by utilising digital technology and interconnectivity such as teleconferencing. The content is personalised to the needs of students, positively impacting the learning curve [74]. Education 4.0 is linked with smart learning environments like LMS and MOOC [77]. Studies show that three common trends will take over the educational sector in the next fifteen years: online institutions, unlimited access to online education, and MOOCs [78].

# Chapter 3. Requirements Analysis and Methodology



## System Specifications Using the MoSCoW Method

The functional and non-functional requirements in this section reflect the final implemented features, categorised by MoSCoW prioritisation. Initial requirements were refined during development to meet project constraints and goals:

* **M**ust Have: Compulsory to achieve a minimum viable product (MVP)
* **S**hould Have: Important requirements that are not necessary
* **C**ould Have: Nice to have these requirements if there is extra time
* **W**ill Not Have: Not at all important because of reasons like complexity or budget

### Functional Requirements and MoSCoW Prioritisation

Functional requirements are system features that are linked to the system’s functionality [79]. The functional requirements using the MoSCoW method are outlined in Table 3.1.

Table 3.1 Functional requirements

|  |  |  |
| --- | --- | --- |
| ID | Details | Priority |
| R1 | Students and staff must be able to log into their respective accounts with suitable privileges. | M |
| R2 | Staff must be able to upload the coursework subtasks. | M |
| R3 | All users must be able to see the subtasks on their timeline. | M |
| R4 | Students must be able to view the coursework subtasks and visualisations. | M |
| R5 | Every coursework on the student portal must have a progress bar to track progress. | M |
| R6 | Staff must be able to track students with zero submissions. | M |
| R7 | Students should be able to organise and prioritise subtasks on Kanban Board. | S |
| R8 | Users should be able to edit their profiles. | S |
| R9 | Staff could be able to share submission statistics with students. | C |
| R10 | The website could allow users to personalise their UI. | C |
| R11 | Reminders could be provided to each student via email or website notifications. | C |
| R12 | Students who submit coursework early could be rewarded with digital incentives or be placed on a leaderboard. | C |
| R13 | Lecturers could provide one-on-one guidance to low-ability students. | C |
| R14 | The website will not support coursework grading | W |
| R15 | The system will not have a mobile version. | W |
| R16 | Advanced AI features, such as AI assistants, will not be integrated. | W |
| R17 | The website will not check for malpractices such as plagiarism. | W |

### Non-Functional Requirements

Non-functional requirements define the performance of the system [80]. They have also been prioritised using MoSCoW, as outlined in Table 3.2.

Table 3.2 Non-functional requirements

|  |  |  |
| --- | --- | --- |
| ID | Details | Priority |
| N1 | Security: The website must defend against intrusion by implementing access control, to protect student and lecturer information. | M |
| N2 | Usability: The user interface must be intuitive, satisfying, user-friendly, and meet the needs of the target audience. | M |
| N3 | Documentation: Comprehensive documentation, user guides or FAQs must be provided for all users, including developers. | M |
| N4 | Recoverability: The website should recover from system failures and resume normal processing. | S |
| N5 | Accessibility: The system could be accessible to users with disabilities. | C |
| N6 | Availability & Reliability: The website will not be available at all times with minimal downtime and perform without failures. | W |
| N7 | Scalability: When the load increases, the website will not be able to scale up. Latency and throughput shall be at acceptable levels. | W |
| N8 | Performance: The system will not be able to handle multiple concurrent users without affecting the response time. | W |

### MVP Goals and Additional Features

The MVP focuses on high-priority requirements aligned with limited time and expertise:

* Student Portal: Each student is enrolled in four courses and can visualise their progress for each coursework through a colour-coded progress bar (green for "on-time" and orange for "late") and a percentage showing completion. Students can prioritise subtasks using a Kanban board by organising unlocked subtasks, which are locked until the relevant lecture is delivered.
* Staff Portal: Staff can upload coursework subtasks and view student submission statistics. Doughnut charts, dashboard statistics, and a filterable student list allow tracking of students who have not started coursework.
* Shared Features: A Gantt-chart calendar visualises subtasks, reducing memory load and improving deadline tracking. All users can edit their profiles, receive notifications, view FAQs, submit a query, and change settings.

The website was hosted locally and run in development mode. To enforce security, separate portals for student and staff were implemented. To prevent data loss and ensure recoverability, all files were continuously pushed to GitHub with additional local backups in case of security threats such as accidental data deletion on GitHub servers. Help pages were included for all users with Frequently Asked Questions (FAQs) and a contact form for unanswered queries. During development, evaluation, and testing, the website occasionally became unresponsive but was restored by restarting the development server using the “npm run dev” prompt. Usability heuristics were used to enhance usability, and supervisor and university friends were regularly consulted as potential end users to provide feedback. Supervisor was contacted more frequently and treated as a client whose suggestions were considered strongly to achieve a high-end product. During the initial stages, accessibility was considered by including Accessible Rich Internet Applications (ARIA) in JavaScript to enable assistive technologies like screen readers. However, due to time constraints, the idea of accessibility was discontinued. Canvas was used as an inspiration, as it is considered to have the highest usability among LMS platforms.

### Features Planned but Not Implemented

The website was initially planned to send weekly reminders to students, with increased frequency to twice a week as deadlines approached because daily reminders can potentially spam and irritate students. Additionally, personalisation options were considered, allowing students to adjust font sizes or change the overall appearance with a dark theme. A reward system was also planned to motivate students on timely submissions, placing them on a leaderboard as an added incentive. These features were implemented into the GUI but were not made functional. Table 3.3 compares the features planned with features implemented.

Table 3.3 Comparative analysis of features planned and implemented

|  |  |  |
| --- | --- | --- |
| Feature | Proposal | Final Product |
| Student | | |
| Submit coursework for up to four courses | ✔ | Fully Implemented |
| Receive reminders | ✔ | Partially Implemented |
| Display colour-coded progress bar and visualisations | ✔ | Fully Implemented |
| Eisenhower matrix | ✔ | Not Implemented |
| Automated feedback on submissions | ✔ | Not Implemented |
| Mandate project journal submission | ✔ | Not Implemented |
| Kanban Board | **✗** | Fully Implemented |
| Rewards for on-time submission | **✗** | Partially Implemented |
| Locked subtasks | **✗** | Fully Implemented |
| Staff | | |
| Upload coursework subtasks | ✔ | Fully Implemented |
| Display student submission statistics | ✔ | Partially Implemented |
| Display deadline and progress visuals | ✔ | Fully Implemented |
| Provide one-on-one guidance to students | ✔ | Not Implemented |
| Track students with zero submissions | **✗** | Partially Implemented |
| Share submission statistics with students | **✗** | Partially Implemented |
| Shared | | |
| Log in | ✔ | Fully Implemented |
| UI personalisation | ✔ | Partially Implemented |
| Time mapping calendar | ✔ | Fully Implemented |
| Edit profile | ✔ | Fully Implemented |

## Software Development Methodology

Solo Scrum was used to develop Coursework Wizard. Scrum is an agile, user-centric methodology that speeds up the development process. A product backlog was prepared containing a list of features required for the final product. The backlog was divided into sprints, which are timeframes, typically ranging from one to six weeks, for completing tasks in the sprint backlog. Sprints were created successively until the product backlog was empty. At the end of each sprint, a sprint review was held to reflect on what was accomplished (Appendix K) and to plan for improvements in the next sprint (Appendix J). This iterative process ensured that each sprint becomes more efficient. Typically, teams hold daily scrum meetings to discuss daily goals and ensure the product is always workable and in a deliverable state [81] [82] [83] [84]. The entire process is summarised in Figure 3.1.

|  |
| --- |
| Figure 3.1 The Scrum Methodology [83] |

However, despite symbolising teamwork, Scrum can be adapted for solo developers, where one person performs the roles of all team members [85] [82] [83] [84]. The Scrum framework was adapted as mentioned below:

* Product Owner: Defines and prioritises the project’s scope. In this project, the product backlog was managed in JIRA (Figure 3.2) and the project's visual and functional scope were designed during Sprints 1-3.

|  |
| --- |
| Figure 3.2 JIRA Backlog |

* Scrum Master: Ensures the Scrum process runs smoothly, helps resolve problems, and keeps the project on track. As a solo developer, the progress was monitored through the Kanban board (Figure 3.3), JIRA reports (Appendix L), and project journals. Obstacles were identified and addressed by researching solutions or discussing issues with the supervisor when needed. This role was carried out throughout the entire project.

|  |
| --- |
| A screenshot of a chat  Description automatically generated  Figure 3.3 JIRA Kanban Board for Sprint 9 |

* Frontend Developer: Responsible for building the UI and ensuring design consistency. In this project, the focus was on building the user interface for the student and staff portals, ensuring alignment with the Figma prototypes. This work occurred mainly in Sprints 4-5.
* Database Designer and Backend Developer: Responsible for the database setup and backend logic. These roles were combined to design and set up the database, integrate it with the frontend, and implement the backend functionality for both the student and staff portals. This was primarily completed in Sprints 6-8.
* Tester and Documenter: Ensures usability testing is conducted, and the project is properly documented. This role was adapted in Sprint 9 where staff testing was conducted, feedback was gathered, and the results were documented for the report.

The project was developed incrementally, prioritising the MVP features first. Due to time constraints, non-essential features were moved to later sprints to focus on core functionalities, highlighting the agile flexibility of Solo Scrum. The development was completed in 9 sprints, with multiple features being implemented in each sprint. The development process could have been smoother if the sprints were more concise, rather than being vague or abstract, especially at the start of the project. Table 3.4 provides step-by-step details of each sprint's objectives, role adaptations, and results.

Table 3.4 Sprint breakdown

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint | Focused Features | Adapted Role(s) | Outcome |
| 1 | Website theme, logo design, and prototype of the landing pages | Product Owner and Designer | Selected the website’s colour theme, designed the logo, and created landing page prototypes in Figma. |
| 2 | Prototype for the login pages and student portal (R1, R4, R7) | Product Owner and Designer | Developed Figma prototypes for the login pages and partially completed the student portal prototype. |
| 3 | Prototype for the student and staff portals (R2, R3, R4, R5, R6, R8, R10, R11) | Product Owner and Designer | Finalised the full website prototype on Figma, covering both student and staff portals. |
| 4 | Set up the Next.js application and build initial frontend (R1, R3, R5, R7) | Frontend Developer | Set up the Next.js environment and implemented the GUI for pre-login pages and the majority of the student portal frontend. |
| 5 | Complete frontend for student and staff portals (R2, R3, R4, R6, R8, R11) | Frontend Developer | Finished all frontend coding for both the student and staff portals. |
| 6 | Design, create, populate, and connect database | Database Designer and Developer | Created the MySQL database, designed the schema, populated it, and connected it to the frontend. |
| 7 | Complete backend coding for the student portal (R1, R3, R5, R7) | Backend Developer and Tester | Implemented backend logic for the student portal and conducted initial usability testing. |
| 8 | Complete backend coding for the staff portal (R2, R3, R4, R6, R8, R11, R12) | Backend Developer | Completed backend coding for staff functionalities. |
| 9 | Conduct staff testing and finalise the report | Tester and Documenter | Conducted staff interviews for testing, gathered feedback, and completed the final report. |

The process followed the core principles of Solo Scrum, with adjustments made to fit the needs of a single developer. While daily scrums were skipped, tools like JIRA and the Kanban board helped maintain progress and ensure continuous delivery. The development was iterative, with each sprint becoming more focused and efficient, although clearer sprint goals at the beginning would have streamlined the process further (Figure 3.4).

|  |
| --- |
|  |
| Figure 3.4 Sprint Flowchart |

# Chapter 4. Design and Implementation

## Development Tools and Technologies

The development of the Coursework Wizard website was carried out using a combination of technologies to meet the project objectives. The following technologies were used:

* Next.js: Next.js was chosen for its full-stack capabilities, as it is built on top of React and includes both front-end and back-end functionality. This eliminated the need to learn separate technologies for the front-end and back-end, saving time and effort. The entire website was developed using Next.js, maintaining a consistent development environment throughout the project. The Next.js environment was started, refreshed and updated by prompts on the PowerShell (Figure 4.1). ESLint and Tailwind were not used for the development, the main language selected was JavaScript and not Typescript.

|  |
| --- |
| Figure 4.1 Running Coursework Wizard in development mode |

* JavaScript: JavaScript is the foundation of the project, as it is required for both React and Next.js development. Initially, React was considered for front-end development; however, by transitioning to Next.js, the development process became more efficient.
* Visual Studio Code: The editor used for coding the JSX and CSS files was Visual Studio Code (Figure 4.2). The initial idea was to use an Integrated Development Environment (IDE) like Webstorm but research suggested the potential problems it could cause. Instead, VS Code was finalised, and relevant extensions such as React.js Code Snippets, React Extension Pack, Prettier, npm Intellisense, Live Server, ESLint, ES7+ React/Redux/React-Native snippets, and HTML/CSS/JavaScript Snippetswere installed.

|  |
| --- |
| Figure 4.2 Visual Studio Code |

* MySQL: My Structured Query Language (MySQL) was selected as the database system for the project due to prior experience with relational databases. Given the time constraints, there was insufficient time to learn a non-relational database like MongoDB. MySQL’s familiar structure and commands provided an efficient solution for managing the project’s data. MySQL Workbench was used for designing the Entity Relationship Diagram (ERD) and creating the local MySQL database (Figures 4.3 and 4.4).

|  |
| --- |
| Figure 4.3 MySQL Workbench for designing and creating the ERD |
| Figure 4.4 MySQL Workbench for creating and managing the database |

* Postman: Postman was used to test and validate APIs during development, particularly for operations that involved interacting with the database (e.g., adding rows). This tool was essential for ensuring the backend APIs were functioning correctly (Figure 4.5).

|  |
| --- |
| Figure 4.5 Postman for testing the APIs |

* GitHub: Used for version control throughout the project. This was critical for preventing data loss, as well as for tracking changes made to the project over time (Figure 4.6). Using GitHub allowed to record changes and provided the ability to rollback to earlier versions, particularly during the integration of different components. However, there have been cases of GitHub crashing so local backups were also created to be on the safe side.

|  |
| --- |
| Figure 4.6 Running Coursework Wizard in development mode |

* JIRA: For tracking progress throughout the project. It was used to manage tasks, set deadlines, and monitor the overall dissertation process. JIRA helped to stay organised and ensured that each phase of development was completed within the designated timeframe.
* Figma: Figma was used to create a high-fidelity prototype of the entire website UI and depicted the actual interactions of the website, serving as a guide for front-end development. This prevented spending excessive time on design decisions or trial and error during front-end coding. The prototype also helped identify and address potential usability and UX issues early on, ensuring smooth development.
* Other tools and technologies: Apart from the primary tools and technologies, additional resources were used for completing this project. YouTube, [FreeCodeCamp](https://forum.freecodecamp.org/t/radio-buttons-in-a-group/517481/10), [Next.js](https://nextjs.org/docs/messages/next-router-not-mounted) documentation, and [NewLine](https://www.newline.co/fullstack-react/assets/media/sGEMe/MNzue/30-days-of-react-ebook-fullstackio.pdf) were used for tutorials on JavaScript, React, Next.js, and implementing features like Timeline, Doughnut Charts, and Kanban Boards. Platforms such as Google, [Reddit](https://www.reddit.com/r/neocities/comments/1bxeueu/how_to_keep_sidebars_from_overlapping_main/), ChatGPT, and [Stack Overflow](https://stackoverflow.com/questions/75734651/how-to-center-a-div-in-html:%20div%20alignment) provided assistance in troubleshooting and resolving implementation challenges. Design tools like [LunaPic](https://www5.lunapic.com/editor/) and [ThisPersonDoesNotExist](https://thispersondoesnotexist.com) were used for editing images and creating personas, while Google Fonts and Dafont helped in downloading fonts. Testing resources included [Marker](https://marker.io/blog/usability-testing-template#website-homepage-usability-template) for usability documentation, [Testfort](https://testfort.com/blog/why-your-project-needs-ui-ux-testing) for testing checklist, [Medium](https://medium.com/@userfocus/the-1-page-usability-test-plan-dbc8c3d7fb54) for usability test plan, and Microsoft Forms for post-usability surveys. Additional tools like Outlook for communication, Microsoft Word for documentation, Google Chrome for website viewing, and Snipping Tool for capturing visuals further supported project completion.

These tools were carefully chosen to support project execution, reduce the need for learning too many new technologies, and to address the challenges faced during solo development.

## Personas and User Requirements

Several personas were created during the design phase to understand the users and the context in which they will be using the website [86]. These personas helped illustrate the goals, challenges, and expectations of users and through them, the MVP and usability requirements were finalised (Appendix M). Below is a summary of 2 personas for student and staff:

* Persona 1: Alex Gibbler (BBA Student)
  + Goal: Keep track of deadlines and task prioritisation
  + Challenges: Forgets deadlines, starts working last moment, fails to prioritise tasks
  + Requirement IDs: R3, R4, R5, R7, R11
* Persona 2: Dr. Arnold Jacob (Mathematics Professor)
  + Goal: Manage and track student performance and progress
  + Challenges: Busy but organised, multiple classes and overloaded, has many students with late submissions
  + Requirement IDs: R3, R6

As mentioned in Chapter 3, the functional and non-functional requirements of the website were developed to address the challenges identified by the personas. The high-priority requirements of the MVP were selected to provide immediate value to the target users. Below is a summary of how these requirements align with the personas:

* Student (R4, R5, R7, R11, R12): The student personas require an intuitive way to track and manage their coursework progress. Features like progress bars, task prioritisation (via the Kanban board), and personalised reminders are designed to address these needs.
* Staff (R2, R6, R9, R13): Staff personas require features for managing coursework and tracking student performance. The ability to upload coursework and monitor student submissions through visuals will help meet these needs.
* Shared Features (R1, R3, R8, R10): Both student and staff personas can visualise deadlines on the Gantt chart calendar and personalise their account and profile. Usability is prioritised for all users to make the website easy to use and navigate.

## Prototyping and UI Design

A high-fidelity prototype of the entire system was developed in Figma to ease and speed up the development process (Appendix D). This phase involved critical design decisions, including the selection of the logo, colour palette, font sizes, content organisation, and website layout. The prototyping process took over a month, as it involved multiple iterations and improvements based on feedback from friends and the project supervisor. Because of unfamiliarity with Figma and too many options to decide from, this phase took longer than expected, thus causing delays in the initial project plan. A common problem encountered on Figma was that the same component was not allowed to have more than one event such as hover and on-click. For this reason, two buttons were added for design purposes, increasing the workload. Later, such animations were ignored and directly implemented in the final website.

The first prototype was created quickly but lacked usability and aesthetics (Appendix D – Figure 2). As a result, the process was restarted. Before designing any pages, the colour scheme was finalised. Inspired by the green colour palette of the Teachable website, which gave a refreshing yet academic feel, a similar colour scheme was adopted but adjusted to make it more suitable for an academic platform by changing the shades slightly. Verdana and Georgia fonts were chosen because they appeared neutral, professional, and neither overly formal nor dull. Larger font sizes were used to enhance visibility and clarity for users. The page layouts were influenced by usability principles, focusing on simplicity and reducing visual clutter. While designing the student portal, the main challenge was to organise the content in a readable fashion. To design and make changes in Figma was time-consuming and thus, a rough design on paper was created first (Figure 4.7).

|  |
| --- |
| Figure 4.7 Paper Prototype |

The following websites were referred for inspiration:

* Home page: Inspired from Dribbble which offered clean and modern web designs.
* Log in page: Inspired by OASIS (Curtin University) for its minimalism design.
* Help page: Inspired by Etihad Airways website for its interactive FAQ section, where collapsible features made the page appear clean and well-organised, as well as provide quick answers to common questions.
* Dashboards: Inspired by Power BI which included graphs and statistical displays.
* Kanban Board: Inspired by JIRA adapted to help students prioritise coursework subtasks by categorizing them as “To-Do”, “In Progress”, and “Completed”.
* General Layout: The Canvas LMS was used as an inspiration for the overall website. As mentioned in Chapter 2, Canvas ranks highest in usability compared to other LMSs, and the side navigation bar minimises space usage while ensuring ease of access. The icons on the navigation bar were free icons from Iconscout.

The initial logo design featured an owl symbolizing academics and wisdom, but it added unnecessary complexity. A simpler logo using the initials "CW" was chosen instead. However, feedback revealed that the initials resembled Arabic letters (ان), leading to further revisions (Figure 4.8a). The final logo adopted a more straightforward and readable style, ensuring clarity and professionalism (Figure 4.8b).

|  |  |
| --- | --- |
| (a) | (b) |
| Figure 4.8 Coursework Wizard Logo (a) Old (b) New | |

Throughout the prototyping phase, feedback was collected iteratively from friends and classmates that acted as student personas, and the supervisor, representing the staff persona. Their suggestions were incorporated into the design to ensure the website remained clean, user-friendly, and visually appealing. Changes suggested were:

* Replace calendar with timeline: The first design for the timeline was a simple calendar with highlighted dates to show deadlines (Figure 4.9a). This was not effectively visualising the deadlines and thus the feature was redesigned once more before the final design was achieved (Figure 4.9b).

|  |  |
| --- | --- |
| A screenshot of a calendar  Description automatically generated(a) | A screenshot of a computer  Description automatically generated  (b) |
| Figure 4.9 Rejected Designs for Timeline (a) Design 1 (b) Design 2 | |

* Make courses a clickable tab: The dashboard was cluttered with buttons to view the coursework. It was suggested to instead make the tab clickable and remove the buttons.
* Coursework deletion: Editing published coursework can be a hassle and create unnecessary problems. Thus, edit coursework buttons were replaced by delete button.
* Coursework addition: The website should divide subtasks into dependent (sequential subtasks with prerequisites) and independent subtasks and lock them until the relevant lecture material is covered during lectures. The staff should be able to reorder the subtasks and delete them during the adding process.
* Subtask weight: It was recommended to assign each subtask a weightage for calculating progress only for staff members. However, weightage was added to the website but never used for calculations.
* Sharing class performance: The staff can release the class performance to students so that they can see how many students have completed. The share button was added to the staff portal but without any functionality.
* Course colours: The colours of the courses should be consistent across the website.
* Board: Rename to progress board to make the functionality self-explanatory to students.

While the prototype captured most of the website's functionality, a few features were added later during development. This iterative approach was made possible by the agile nature of the project, which allowed for incremental development and adjustments throughout the process.

## System Design

The system design of the project involved defining the data structure, user interactions, and logical flow of the website. The focus was on creating a micro-LMS that supports efficient coursework management for both students and staff.

### System Architecture

The system built on Next.js 14 consists of two primary user types: students and staff. Students can view and manage their coursework tasks, while staff can upload coursework and track student progress. Both user roles interact with the system through a well-defined UI designed to enhance usability and time management. No specific architectural pattern, such as MVC, was formally adopted but the modular approach naturally organised the system into distinct layers of functionality (Figure 4.10). The system consists of 4 parts:

1. Frontend: Responsible for the user interface components. It performs client-side rendering and user interactions with the GUI to enable smooth transitions between server-rendered and client-rendered pages.
2. APIs: Responsible for the backend and functionality. It handles user authentication, role-based access control for staff and students and is responsible for fetching and updating data in the database.
3. Database: It stores all the data of the website like the users, subtasks, and submissions.
4. Local Host: Since the website is hosted locally, the environment is on a single laptop.

|  |
| --- |
| Figure 4.10 Coursework Wizard Architecture |

The communication between the Next.js client and the database is handled using two types of requests: GET, for fetching data from the database, and POST, for updating the database. The requests are received by the middle layer, called the Express API server, which passes them to the in-memory cache, Redis. POST requests are submitted to a publish/subscribe system, while GET requests first check for data in the cache. If the data is found in the cache, it is passed to the client; otherwise, it is fetched from the database and stored in the cache for future requests. POST requests are sent to the worker server, which subscribes to the publish/subscribe system to receive messages. The data in the MySQL database is then updated and saved (Figure 4.11) [87].

|  |
| --- |
| Figure 4.11 Next.js Architecture [87] |

### Database Design

The database was designed after completing the front-end development and played a critical role in managing the system's data. The schema was modelled using a relational database approach, implemented on MySQL due to its reliability and familiarity. The design captures key entities, relationships, and constraints required for uninterrupted operation. The main tables and their purpose include:

* User: A parent table to manage login credentials and roles (student or staff).
* Student and Staff: Two separate tables that extend the user table to store additional role-specific attributes.
* Course and Coursework: Tables to manage courses, their associated coursework, and deadlines.
* Subtask: Tracks individual coursework components with attributes such as start/end dates, weights, file attachments, and whether the subtasks are locked or unlocked.
* Enrolment: Manages the many-to-many relationship between students and courses.
* Submission: Stores data about students' submissions, including status for the Kanban board and file attachments.

The system supports key relationships such as:

* A course is managed by a single staff member.
* Students can enrol in multiple courses, while each course can have multiple students.
* Each coursework contains subtasks, which can be visualised and tracked by students.
* Submissions link students to their progress on individual subtasks.

The Entity-Relationship Diagram (ERD) for the system is shown in Figure 4.12. The ERD was finalised after completing the front-end development. Feedback from supervisor and backend development of the website resulted in iterations of the ERD, highlighting the agile nature of Scrum.

|  |
| --- |
| A screenshot of a computer  Description automatically generated  Figure 4.12 Entity-Relationship Diagram (ERD) for Coursework Wizard |

The design followed a modular and relational database structure to ensure:

1. Data Integrity: Foreign keys were used to establish relationships and maintain consistency across tables, ensuring that data dependencies are enforced.
2. Recoverability: Regular GitHub commits, pushes, and local backups were maintained throughout development to mitigate the risk of data loss.

The database was initially poorly structured, not normalised, and impractical for some key functionalities. The design encountered scalability challenges due to its dependency on manually populated submission records. For instance:

* Manual Subtask Entries: To track submissions and allow updates to subtask statuses on the Kanban board, each subtask for every student required a separate row in the submissions table to be manually entered. If a student has 48 subtasks, 48 rows are required for that student. Although this solution allowed the Kanban board to function and dynamically change subtask status upon submission, it is highly impractical and caused issues when integrated with the staff portal (Figure 4.13).

|  |
| --- |
| Figure 4.13 Submission Table |

* Partial Normalisation: While the schema followed principles of relational databases, it was not fully normalised to the third normal form, as certain redundancies were retained to simplify development. These redundancies limited query efficiency and complicated integration of staff portal functionalities with the existing student portal.

### Key Design Decisions

1. Security: Role-based access was implemented at the API level to ensure students and staff access only their respective privileges.
2. Simplicity: The schema was designed to balance functionality and simplicity, aligning with the limited time and resources available.
3. Focus on Core Functionality: The database was designed to support core student and staff functionalities, such as progress tracking and coursework management, rather than broader scalability.
4. Student Portal Visualisation: Visualisations and gaming modules were used to make the design effective and achieve objectives. Progress for every coursework was represented by a progress bar, completion percentage, emoji, and colour coded tabs. Upon successful submission, the colour of the bar and tab changed to green, otherwise remained orange. Every coursework also had an emoji that varied from sad to joyful depending on the number of submissions made. Stars were also rewarded for on-time submission. Subtasks that have not been covered in lectures were disabled and overlayed with a lock emoji. Every coursework also had a section displaying the next upcoming deadline along with the subtask and course name.
5. Staff Portal Visualisation: The staff dashboard had colour-coded doughnut graphs to represent submission statistics for each coursework where green meant completed, yellow meant in progress, and red meant pending. Furthermore, the sortable students list also had colour codes for every student to make it easier to track students with minimal submissions.

## Implementation

After developing the prototype, the supervisor was consulted multiple times to gather feedback until the design achieved maximum satisfaction. Once the prototype was finalised, the project transitioned from the role of designer to front-end developer for working on the GUI of the website. The first step was to install Next.js 14 and create a default application using the command prompt. The application was run in development mode, with the code edited on VS Code and the website tested on Google Chrome.

During the installation, Tailwind CSS configuration was skipped to reduce complexity due to a lack of prior experience with the technology. Instead, all styling was done manually from scratch, using the prototype as a reference. However, despite having the prototype for reference, the overall frontend development was not very smooth because the frame size of the prototype did not match the display size of the actual website. The prototype frames had more height and less width, whereas the actual website was opposite. Therefore, the font sizes and content organisation had to be readjusted. Moreover, the side navigation bar would highlight the Dashboard when courses or account pages were accessed. If statement was added to highlight the relevant icon on the side bar if the web page belonged to the account or course folders.

The next step was to create the database in MySQL and populate it with dummy data to make the website dynamic so that it can load and display the accounts of individual users. Following this, the backend of the website was developed using APIs. A connection to the local MySQL database was established and named “pool”. Individual API files were created for each action, utilising SQL commands to perform database operations. Upon successful execution, the corresponding functionality was implemented on the website. A simple formula was used to calculate the work progress:

During implementation, unfamiliarity with the technology posed several challenges, causing delays and setbacks. Despite being a moderately popular React framework, Next.js 14, which enforces the App Router configuration, had limited tutorials available. Most of the resources focused on older versions, making it difficult to understand the new folder structure and script naming conventions. The official Next.js documentation became the primary resource for resolving these issues and for learning how to organise files correctly to make the website functional. The newer version required each page to have a separate folder with a page.js file. The same issue persisted when navigation components were added to the website since older versions imported router from router, whereas the newer version imported it from navigation.

At the start of the implementation phase, choosing Next.js felt like a disadvantage due to the steep learning curve. However, once the front-end development was complete, the decision seemed justified because Next.js eliminated the need for separate backend development with Node.js, offering a full-stack development solution. Nevertheless, some challenges persisted. For instance, while working on the timeline and Kanban board, many layout issues arose that could have been resolved easily with Tailwind CSS. Unfortunately, since Tailwind CSS was not configured at the start, installing it later distorted the entire website layout because Tailwind comes with default styling. As a result, Tailwind was disabled and alternate libraries like react-dnd-html5-backend and react-dnd were installed. Moreover, styling the timeline was challenging because all individual bars representing subtask duration had to be properly styled to prevent overlapping and misplacement. The solution was easy but hard to figure out and only required the bars to be styled as flex display and absolute.

One major issue encountered during evaluation was the slow speed of the website. Initially, it was assumed that this was due to local hosting. However, further research revealed that the development mode of Next.js is slower because it continuously reflects code changes and refreshes the application in real-time. Although building the website could have improved performance, this decision was not pursued because the project had already entered the evaluation phase. Building the website required resolving issues which caused the Kanban board to stop functioning.

The project also faced delays because of unfamiliarity with most of the technologies used. Learning JavaScript, followed by React.js and then Next.js was time-consuming, and minimal time was spent recreating small projects from tutorials. As a result, the website was implemented simultaneously while watching tutorials. This approach led to problems, as the implementation often outpaced the learning process. Many front-end components were initially incompatible with the backend, and key concepts such as props were learned later, requiring modifications to the front-end code during backend development.

Due to time constraints, the student portal was made fully dynamic, while the staff portal was only partially dynamic, with many components hardcoded. APIs were created for essential staff functionalities such as login, profile management, coursework, and timeline management. However, features like the doughnut graphs for displaying submission statistics, the student list, and identifying students with zero submissions were hardcoded. While hardcoding exists, the focus of this project was on delivering a usable design that incorporates time and project management functionalities to assist students in their academics and prepare them for their professional lives.

A few requirements, like reminders/notifications, rewards, Eisenhower Matrix, automated feedback, forget password, and journal submissions, were deprioritised, completely disregarded or not made fully functional during implementation. Features like notifications and rewards were added to the UI but were not made functional. Despite these challenges, the project demonstrated significant progress, with the core functionalities for students and staff implemented successfully to deliver the MVP (Screenshots of final website in Appendix E).

## Assumptions

The following assumptions were made during the development of the Coursework Wizard to define the project scope and simplify implementation considering limited time and resources:

* Course Limit: The system was designed to accommodate a maximum of two courses for staff and four courses for students, reflecting the common practice of the university.
* Individual Projects Only: The system is tailored exclusively for individual projects, assuming that students will not work in groups.
* Familiarity with Canvas LMS: It is assumed that users are already familiar with the Canvas LMS interface, as Coursework Wizard’s design is inspired by Canvas.
* Staff Responsibility for Subtasks: Staff are responsible for breaking coursework into subtasks and assigning deadlines, like the WBS described in 2.5.1. This ensures consistency across courses and reduces the complexity for students, who are not required to define their own subtasks.
* Non-Grading System: The system does not support grading functionality, as it focuses solely on task management, progress tracking, and deadline visualisation.
* Submission Deadline: The system assumes that all due dates are schedule at midnight.

# Chapter 5. Testing and Evaluation

## Introduction

The aim of testing was to verify that the Coursework Wizard functions as intended, while the aim of evaluation was to validate that the system meets user needs and fulfils the project’s requirements. Various methods were used during development and post-completion to ensure the system's functionality and usability.

## Testing

### Testing Strategy

During the development phase, testing was performed continuously. After every small change, the website was run in development mode and tested on the browser to check its functionality. This immediate testing ensured that issues were identified and resolved early.

Once the student portal was completed, development shifted to the staff portal. After the staff portal was implemented, the entire website was tested as a whole. However, during integration testing, several issues were encountered where the student portal stopped working properly after the addition of the staff portal. These problems were resolved through debugging and incremental adjustments. This approach of continuous and integration testing ensured that the website remained functional at all stages of development.

During the development, unit and integration testing were performed to ensure that each component was functioning properly independently and when combined with others. Once the system was developed, system testing was performed to test the system all together along with Heuristic evaluation to ensure usability. Ten students and two staff members were recruited for usability testing to verify the core functionalities for the MVP and get detailed feedback for further improvements.

### Unit Testing

Unit tests were conducted to verify individual components and their functionality. Table 5.1 summarises the test cases, expected results, actual results, and status.

Table 5.1 Unit Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Test Case | Expected Result | Actual Result | Status | Notes |
| TC0001 | Verify user login with valid username and password. | Login successful. | User dashboard open | Passed | Nil |
| TC0002 | Verify user login with invalid username and password. | Login failed. Show error message – invalid credentials. | Error message | Passed | Nil |
| TC0003 | Verify student login with staff username and password. | Login failed. Show error message – invalid credentials. | Error message | Passed | Nil |
| TC0004 | Verify staff login with student username and password. | Login failed. Show error message – invalid credentials. | Student dashboard open | Failed | The API was not checking for role on staff login. The test passed on second attempt. |
| TC0005 | Verify password is changed upon reset using forget password. | Login successful using new password. | Student dashboard open | Passed | Nil |
| TC0006 | Verify the upcoming deadline of the next subtask is displayed. | Display deadline for closet subtask with no submission yet. | Closest subtask deadline displayed | Passed | Nil |
| TC0007 | Verify proper navigation icons are highlighted when clicked. | Icon of page open should be light green with black icon. | Dashboard highlighted when courses and account is open | Failed | Modified code by adding if statements to check if the page link is for the course pages, profile or settings, and highlight icon accordingly. The test passed on second attempt. |
| TC0008 | Verify coursework 1 of each course is displayed. | 4 colour-coded and clickable coursework should be displayed on the dashboard. | 4 coursework displayed | Passed | Nil |
| TC0009 | Verify progress bar of each coursework displays the completion percentage and is properly colour coded. | Completion percentage should be displayed on the centre of the bar and the colour should be green or orange depending on the progress. | 17% completion orange coloured bar | Passed | Nil |
| TC0010 | Verify the completion percentage is displayed for the coursework and visualised using different emojis. | Completion percentage should be displayed in simple numbers and visualised using sad or happy emojis. | 17% work completed with a sad emoji saying, “Move Quickly!” | Passed | Nil |
| TC0011 | Verify the colour of the subtask changes from orange to green when submission is made. | The subtask’s background colour should change to green upon submission. | Colour changed to green | Passed | Nil |
| TC0012 | Verify all dependent and independent subtasks for every coursework is displayed and colour coded. | Each course should have 2 lines dedicated for subtasks. First line for dependent subtasks and second for independent subtasks. | All subtasks are displayed | Passed | Nil |
| TC0013 | Verify subtasks can be moved from one column to another. | Subtasks should be able to move between columns. | Subtasks can be moved | Passed | Nil |
| TC0014 | Verify subtasks that have not been taught are locked. | Locked subtasks should be disabled and have a lock icon on them. | Locked subtasks are disabled | Passed | Nil |
| TC0015 | Verify profile is viewed. | Profile should be displayed. | Profile picture not displayed | Failed | Changed attachment data type from blob to varchar. The test passed on second attempt. |
| TC0016 | Verify profile is edited and saved. | Profile should be edited and saved. | Profile edited and displayed | Passed | Nil |
| TC0017 | Verify the staff can delete coursework. | The website should ask for confirmation and remove the coursework from the database. | Coursework deleted after asking for confirmation | Passed | Nil |
| TC0018 | Verify the staff can add new coursework and view it in the timeline. | Coursework should be added to the database and available across the website after validation. | Coursework added and updated on the timeline | Passed | The website asked for validation to ensure the dependent subtasks don’t overlap and the number of total subtasks matches the sum of dependent and independent subtasks. |
| TC0019 | Verify all students are displayed in the list and can be filtered and sorted. | Students belonging to each course and coursework should be displayed with colour coded status to show their work progress. | Students with colour coded status displayed | Passed | The page allows the staff to select the list of students to be displayed and sort them. |
| TC0020 | Verify the staff dashboard displays the number of students with zero submissions and visualises submissions for each coursework separately. | Number of students with zero submissions should be displayed along with colour-coded doughnut graphs for each coursework. | Statistics and graphs displayed | Passed | The data is hardcoded. |

### Integration Testing

Integration testing ensures that individual components of the system work together as expected. For Coursework Wizard, this involved testing the interactions between the frontend, backend APIs, and the MySQL database. Each new feature was integrated incrementally and tested to confirm that it functioned properly within the larger system.

During development, components were continuously integrated into the website. After completing the student portal, the staff portal was implemented and integrated. However, this integration introduced unexpected issues, such as the student portal breaking after the staff portal was added. Specifically, the Kanban board stopped functioning, and subtasks failed to turn green even after successful submissions. This happened because the key constraints in the database tables had been modified during staff portal development. When certain records were deleted, cascading deletions were triggered, which unintentionally removed all records from the submission table. As a result, the submission table was empty, preventing status changes for subtasks. This issue directly affected the progress visualisation feature. To resolve this, the database schema was reviewed, the constraints were fixed, and the deleted records were re-added.

This iterative approach to integration testing ensured that issues were identified and resolved promptly, maintaining the overall usability and functionality of the Coursework Wizard.

### Heuristic Evaluation

A HE was performed to assess the system’s usability based on Nielsen’s usability heuristics. The evaluation identified areas of strength and minor issues as mentioned below:

H1. Visibility of System Status:The system consistently displayed its state and actions clearly. For example, the "Forget Password" process included a step indicator and provided a confirmation page after submitting queries. Similarly, when an action was performed, the system gave immediate feedback to indicate whether the action was successful. The sidebar highlighted the icon of the current page, ensuring users always knew where they were. However, some inconsistencies were observed. For instance, while the coursework names were displayed, they lacked context confusing the user which coursework they are viewing.

H2. Match Between System and the Real World:The system followed a natural and logical order for steps, especially in processes like "Forget Password" and "Add Coursework." The content was tailored specifically for Heriot-Watt University students and staff, and unnecessary information was avoided. Visuals were used extensively to make the content more visible. However, some issues were noted, such as using course codes instead of course names, which might be confusing. Simple and student-focused language was used, but improvements could include standardising coursework naming conventions.

H3. User Control and Freedom:Users had the freedom to resubmit files multiple times without restrictions. Exiting any page was straightforward, as the logout button was always visible. Additionally, deleting a course triggered a confirmation prompt to prevent unintentional deletions. However, some limitations existed. For example, staff members could not go back to edit details after reaching the review page when adding coursework.

H4. Consistency and Standards:The website maintained consistency across all pages. The same colour scheme, button styles, themes, and fonts were used throughout, giving the platform a cohesive look and feel. Even though the student and staff portals offered different functionalities, their design was consistent. Inspired by Canvas and JIRA, the interface had familiar patterns to enhance usability for users familiar with these platforms.

H5. Error Prevention:Input validation was implemented effectively in forms such as "Contact Us" and "Add Coursework," where only specific data types were allowed. A confirmation dialog was used for critical actions like deleting a course. However, some usability issues were noted, such as the confusing "Re/Submit" label on the submit button for subtasks and the lack of hover tooltips on tabs, which could have guided users better.

H6. Recognition Rather Than Recall:All information was presented clearly and prominently, minimising the need for users to memorise anything. Features like colour-coded keys for doughnut graphs and hover tooltips made the interface intuitive. However, the Kanban board lacked detailed instructions or tooltips, which could have helped users.

H7. Flexibility and Efficiency of Use:The system was simple and efficient, with no unnecessary steps added. Default states, such as "To-Do" for all subtasks, made the system predictable and easy to use. However, accessibility features like ARIA were only partially implemented, and input methods were limited to keyboards and mice, which restricted usability for users with special needs.

H8. Aesthetic and Minimalist Design:The website’s design struck a balance between aesthetic appeal and academic focus, with a calm and peaceful interface. Content was clutter-free, well-organised, and visually appealing. However, the large size of components made the interface feel "busy" to some users.

H9. Help Users Recognise, Diagnose, and Recover from Errors:Error handling was implemented in the console with detailed messages pointing to the exact location of errors. However, end users might not recognise or understand these errors unless they accessed the developer console. Data validation errors, such as invalid inputs in coursework forms, displayed clear and helpful messages for users to resolve the issue.

H10. Help and Documentation:The system included a "Help" page with FAQs for students, staff, and users who were not logged in. If a query was not answered in the FAQ, users could submit an inquiry via the contact form. However, there were no tutorials, tooltips, or detailed user guides, limiting the overall accessibility of help resources.

Having verified the system’s technical functionality, the next section focuses on evaluating its usability and alignment with user requirements

## Evaluation

### Usability Testing

Usability testing was conducted with 10 students who were given a task sheet outlining specific functional actions to perform on the website (Appendix G). First, the students were introduced to the system and distributed the information sheets. Once they agreed, they were asked to sign a consent form before proceeding with the test (Appendix H). During the testing, students followed the task sheet while thinking aloud, allowing their navigation patterns, errors, and observations to be recorded. The sessions were screen-recorded for further analysis to identify recurring patterns or challenges faced by users. The entire plan is summarised in Figure 5.1.

|  |
| --- |
| A close-up of a diagram  Description automatically generated  Figure 5.1 Usability Test Plan |

After completing the tasks, students were asked to fill out an SUS (System Usability Scale) survey and a post-usability questionnaire to provide feedback (Appendix F). The questionnaire was designed to have less open-ended questions and more Likert scales questions. The average SUS score was 80, indicating good usability. The evaluation data and calculations are presented in Appendix I. The feedback highlighted the following:

Strengths:

* All participants found logging in and out of the system straightforward. The logout button was always visible, making it easy to exit any page without confusion.
* Participants appreciated the website's calm and academic feel. The colour scheme, typography, and visuals were well-received, with specific praise for the use of green in the design. The average score received was 7.7 out of 10. Majority participants were able to identify clickable content and 7 out of 10 participants found the content size appropriate.
* The system provided clear feedback for most actions, such as indicating progress updates after submitting subtasks or confirming password changes. This aligns with heuristic principles for system status visibility.
* Features like "drag-and-drop" on the Kanban board and locked subtasks were understood by some participants, showing potential for usability with further clarification.
* The participants found the website very similar to Canvas.

Weaknesses:

* Most participants struggled with the drag-and-drop functionality on the "Board" page. The lack of instructions or tooltips made this task difficult for users unfamiliar with such interfaces.
* Many participants did not notice the progress bars for coursework, indicating that their placement needs improvement.
* The use of course codes instead of course names in some places confused participants. A more consistent naming convention, such as "Advanced Interaction Design: Application Design Coursework" was suggested.
* The labelling of the submit button as “Re/Submit” confused participants, as they assumed it was only for resubmissions.
* While a "Help" page was available, most participants avoided it or did not fully utilise the FAQs. Additionally, many clicked links for solutions without reading the provided guidance.
* Several participants used the browser's back button instead of the website's navigation features, indicating that the interface could better guide users.
* Alert windows displaying error messages were ignored by participants indicating lack of visibility.
* Some participants struggled with unclear instructions on the task sheet, which affected task performance. Additionally, testing was conducted when participants were in a hurry as it was getting late, leading to rushed interactions and incomplete observations.
* Participants expected the "Change Password" option to be in settings or profile sections, which differed from its actual location. Some also wanted a "show/hide" feature for passwords during updates.

The test sample size was very small, but the results show that the usability of Coursework Wizard is “Good” as per the scale mentioned in chapter 2.3.2. However, the results could have been more accurate if the tasks on the task sheet were clearer, and testing was scheduled earlier in the day. Moreover, few participants were not comfortable using Windows OS.

### Staff Feedback

The staff feedback evaluation involved two professors from Heriot-Watt University. Initially, five staff members were contacted, but only two responded. Due to scheduling conflicts, individual sessions were conducted with each professor. Since the staff portal was primarily hardcoded, asking staff members to test it directly was not feasible. Instead, the evaluation focused on demonstrating the website and gathering their opinions through a semi-structured interview format (Appendix I).

Student Portal:

* Both staff members agreed that the student portal provides benefits by helping students manage their deadlines and coursework visually. They appreciated the timeline and dashboard, which reduce memory load by showing all coursework deadlines in one place. They noted that students no longer need to remember deadlines, which is particularly beneficial for younger students or those new to university.
* Staff Member 1 emphasised that the student portal was an improvement over Canvas, which lacks tools for task prioritisation.
* However, both professors highlighted that some features might not be practical for older students or postgraduate students who are expected to independently manage their work.

Staff Portal:

* The staff portal received mixed reviews. Both professors expressed concerns about the additional workload placed on staff by requiring them to break down coursework into subtasks. Staff Member 1 estimated that 60–70% of coursework might become more time-consuming to manage, especially for written assignments. They suggested that it is better for students to learn time and project management themselves.
* Staff Member 2 disagreed with the idea of staff "babysitting" students, arguing that this level of micromanagement is impractical in real-life scenarios and does not align with the goal of preparing students for independence. They suggested that the staff portal should focus more on marking, identifying late submissions, and other administrative tasks rather than planning coursework for students.
* Additionally, Staff Member 2 raised a concern about the ability to verify if students are genuinely completing their work. They pointed out that even with incremental submissions, students could use generative AI tools to create the entire coursework, break it into subtasks, and submit them one by one. This weakens the argument that breaking coursework into subtasks can help staff identify the work is completed by the student.

Kanban Board:

* While both professors found the Kanban board useful, they pointed out its lack of instructions and clarity. Staff Member 1 recommended using a clearer icon to represent the board, as many students may not be familiar with project management tools.
* Staff Member 2 was less enthusiastic, stating that the Kanban board felt unnecessary and overly playful, potentially turning education into a "circus".

Usability, Look, and Feel:

* Both professors praised the system's usability, stating it was intuitive, familiar, and easy to navigate. Staff Member 1 noted that the design would be approachable even for first-time users, as all information is visible, and the interface avoids unnecessary clutter.
* Staff Member 2 suggested adding more guidelines for certain features, such as the Kanban board, to enhance usability.

Forget Password Process:

* The "Forget Password" feature was appreciated for its simplicity and clarity. Both professors liked the step indicators (e.g., Step 1 of 4) and the immediate feedback provided at each stage, which aligned with usability standards.

Visualisations:

* The professors agreed that visualisations, like progress bars, effectively reduce memory load. However, Staff Member 1 noted a limitation: when a coursework has only one subtask, no progress is displayed. They suggested introducing additional features, such as a Pomodoro timer, to encourage time tracking. However, they also acknowledged that students often rush to meet deadlines, which may limit the timer's effectiveness.

Share Button:

* Staff Member 1 suggested that the share button could be repurposed to export statistics as a CSV file for internal use by staff, rather than sharing data directly with students. Staff Member 2 believed that sharing progress data with students could be motivating but recommended keeping it optional.

Gamification:

* Opinions on gamification features like the leaderboard were divided. Staff Member 1 suggested anonymising rankings to avoid increasing anxiety among students, proposing that only percentages (e.g. Top 3%) should be shown. They also acknowledged that the effect of ranking systems depends on the student’s personality, therefore it should be optional.
* Staff Member 2 was less supportive, stating that gamification might not suit the platform’s purpose of preparing students for professional environments. They believed that the dashboard and timeline provided sufficient motivation without adding gamified elements.

Additional Suggestions:

* Staff Member 1 proposed adding a "reminder" feature to the staff portal, allowing professors to send emails to students who have not started their coursework. However, Staff Member 2 opposed this idea, arguing that it would increase the workload and promote micromanagement.
* Push notifications for students (e.g. a deadline reminder email one week before the due date) were also suggested as a useful feature.

Overall, the staff feedback highlighted several strengths of the system, such as its usability, visualisations, and focus on helping students manage their coursework effectively. However, the additional workload on staff by the staff portal was a major concern. Both professors emphasised the importance of refining features to balance functionality for students while minimising the burden on staff. The student portal was rated highly, with its features seen as a valuable improvement over existing systems like Canvas. However, the staff portal requires further development to address the concerns raised and better align with professional life.

## Summary of Findings

The evaluation results, including SUS scores and feedback from students and staff, validated that Coursework Wizard achieved its goals of providing task and time management tools alongside progress visualisation. The average SUS score of 80 indicated good usability, and students appreciated the website's aesthetics, simplicity, and focus on academic functionality.

The testing and evaluation revealed several areas for improvement across the system:

* The Kanban board lacked clarity and instructions, making it difficult for users to understand and use effectively.
* The progress bar and some visual elements were overlooked by many students, suggesting the need for improved placement.
* Inconsistent terminology, such as using course codes instead of names in some areas, confused users and disrupted the flow.
* Accessibility features were minimal, with limited support for users with special needs, such as screen readers or voice commands.
* Hardcoded elements in the staff portal restricted flexibility and practicality.

Despite these challenges, the project demonstrated significant progress in delivering a functional MVP. Coursework Wizard was appreciated for its usability, intuitive design, and focus on addressing key academic challenges faced by students. These findings provide a strong foundation for future improvements, including enhancing the Kanban board, increasing accessibility, and replacing hardcoded elements with fully dynamic components to support scalability.

Overall, the system successfully met its core objectives and showed great potential to evolve into a robust platform that balances student and staff needs, offering a valuable tool for academic and professional growth.

# Chapter 6. Conclusion and Future Work

## Potential for Industry Implementation and Future Work

If this project was to be developed in the industry, the website would be fully dynamic and get integrated into an LMS such as Canvas or Moodle. The website would adapt to different screen sizes and include an application for mobile users. Inclusive UX design would allow all kinds of users to utilise Coursework Wizard by supporting people with varying disabilities like colour-blindness and hearing impairment.

Moreover, to support the non-functional requirements, the website could be connected to Cloudflare’s nameserver, which would provide a dashboard to visualise website traffic, security events, and performance metrics in real time. By using these tools, the development team could identify bottlenecks and take actions to optimise the website. With Cloudflare’s reverse proxy, website traffic would first go through Cloudflare’s servers and then reach the website, increasing security, performance, and reliability:

* Performance: Website contents will be stored on the nearest Cloudflare server, reducing response time because data will be loaded from the nearest server.
* Security: Cloudflare’s Web Application Firewall (WAF) will protect the website from attacks by filtering malicious or suspicious requests.
* Scalability and Reliability: When website traffic increases near deadlines, the load balancer will distribute load and scale up by using more servers to ensure high availability, reduced downtime, and fault tolerance.

Future work could include piloting the system in a real academic setting. For example, one lecturer could adopt the platform for a semester with a small group of students. This pilot phase would provide valuable insights into how the system performs under real-world conditions and highlight further areas for improvement. Features like advanced analytics, mobile compatibility, and enhanced accessibility could then be added based on this feedback.

## Limitations

Although Coursework Wizard met its core objectives, several limitations were identified during testing and evaluation:

* Restricted course and coursework capacity: The system is limited to four courses for students and two coursework per course. These restrictions were necessary to simplify database design and system performance in development mode.
* Progress tracking limitations: The progress bar feature requires multiple subtasks for accurate tracking. If coursework contains only one subtask, progress cannot be monitored.
* No support for group projects: The system is not designed to support group projects, limiting its application to individual tasks only.
* Limited user base: The system targets Heriot-Watt University students and staff, assuming users are already familiar with Canvas. This restricts its wider adoption.
* Static features in staff portal: Many components of the staff portal, such as submission statistics, are hardcoded, reducing flexibility and scalability.
* No grading capabilities: Coursework Wizard excludes any grading functionality, which might limit its usefulness for academic staff who require integrated assessment tools.
* Scalability and performance constraints: The system was tested in development mode, which limits its speed and scalability. It was not deployed or optimised for larger audiences.
* Manual subtask creation: Staff must manually break coursework into subtasks, which will be time-consuming and impractical for certain types of assignments.
* Limited accessibility and mobile support: Accessibility features like screen readers and ARIA attributes are minimal, and the system is not optimised for mobile devices.

While these limitations arose due to time constraints, limited resources, and limited technology competency, they highlight opportunities for future development to improve scalability, accessibility, and usability.

## Conclusions and Final Reflections on the Project

Developing Coursework Wizard was both challenging and rewarding. The project successfully delivered an MVP that achieved its primary objectives of providing task and time management tools, as well as visualisations for tracking progress. The student portal was praised for its usability, simplicity, and ability to reduce memory load by visualising coursework deadlines and progress indicators.

However, the journey was not without challenges. Working as a solo developer meant managing multiple roles, from designing and implementing features to gathering user feedback and conducting testing. Learning new technologies like Next.js, managing database constraints, and handling the steep learning curve for components added to the complexity of the project. The hardcoded elements in the staff portal and the partially dynamic backend further highlighted areas that need improvement.

Reflecting on the development process, I regret not starting earlier, as the time constraints made it difficult to achieve the full potential of the project. Ironically, this is what Coursework Wizard aims to change for students, helping them break down tasks into manageable parts and focus on the process, not just the deadlines. By integrating project management tools like the Kanban board with LMS features, this platform bridges a gap, preparing students not only for academic success but also for professional life.

One of the most valuable aspects of the project was the feedback received from students and staff. While students appreciated the intuitive design and functionality, staff raised critical points about the workload and practicality of certain features, such as manually breaking coursework into subtasks. These insights not only validated the system’s strengths but also provided a clear direction for future iterations.

Despite its limitations, Coursework Wizard demonstrated potential as a platform to help students manage their coursework effectively while preparing them for professional environments. Looking ahead, piloting the system in a real academic setting would provide essential data to refine its features further.

In conclusion, Coursework Wizard has laid a strong foundation for a scalable and user-friendly academic tool. With further development, it has the potential to address the needs of both students and staff while evolving into an industry-ready platform.



# References

|  |  |
| --- | --- |
| [1] | M. R. Peltier, E. M. Bemiss, C. Shimek, A. V. Wig, L. J. Hopkins, S. G. Davis, R. Q. Scales and W. D. Scales, “Examining learning experiences designed to help teacher candidates bridge coursework and fieldwork,” *Teaching and Teacher Education,* vol. 107, p. 103468, 2021. |
| [2] | K. Gregory and S. Morón-García, “Assignment submission, student behaviour and experience,” *Journal of the Higher Education Academ,* vol. 4, no. 1, pp. 16-28, 2009. |
| [3] | M. Nieberding and A. F. Heckler, “Patterns in assignment submission times: Procrastination, gender, grades, and grade components,” *Physical Review Physics Education Research,* vol. 17, no. 1, p. 013106, 2021. |
| [4] | K. Kosidou, C. Dalman, P. Fredlund, B. K. Lee, R. Galanti, G. Isacsson and C. Magnusson, “School performance and the risk of suicide attempts in young adults: a longitudinal population-based study,” *Psychological Medicine,* vol. 44, no. 6, pp. 1235-1243, 2014. |
| [5] | J. L. Olleras, M. Dagwayan, A. M. Dejacto, J. R. Mangay, M. Ebarsabal, D. J. Diaz, C. Puti-an, A. Lendio, J. Nadera, J. Taneo, C. Cabello and A. Minyamin, “The life of the laters: Students procrastination in accomplishing academic deadlines in online learning,” *Psychology and Education: A Multidisciplinary Journal,* vol. 2, no. 5, pp. 444-454, 2022. |
| [6] | M. M. Althobaiti and P. Mayhew, “Assessing the Usability of Learning Management System: User Experience Study,” in *E-Learning, E-Education, and Online Training*, Novedrate, 2016. |
| [7] | M. Bassey, “The Assessment of Students by Formal Assignments,” 1971. |
| [8] | C. M. Stanton, “Assessing Students: How Shall We Know Them? (Book Review),” *Journal of Higher Education,* vol. 50, no. 6, p. 785, 1979. |
| [9] | J. T. Richardson, “Coursework versus examinations in end-of-module assessment: a literature review,” *Assessment & Evaluation in Higher Education,* vol. 40, no. 3, pp. 439-455, 2015. |
| [10] | J. Millman and K. Mitchell, “Assessing Students: How Shall We Know Them By Derek Rowntree (Book Review),” *Journal of Educational Measurement,* vol. 17, no. 1, p. 78, 1980. |
| [11] | N. Ryazanova, A. Semak and E. Kazakova, “ADDIE educational technology for coursework design in environmental education for sustainable development in Russia,” *E3S Web of Conferences 265,* vol. 265, p. 07001, 2021. |
| [12] | P. Stankov, “The death of exams? Grade inflation and student satisfaction when coursework replaces exams,” *International Review of Economics Education,* vol. 46, 2024. |
| [13] | S. French, A. Dickerson and R. A. Mulder, “A review of the benefits and drawbacks of high-stakes final examinations in higher education,” *Higher Education,* pp. 1-26, 2023. |
| [14] | N. A. A. Tuah and L. Naing, “Is Online Assessment in Higher Education Institutions during COVID-19 Pandemic Reliable?,” *Siriraj Medical Journal,* vol. 73, no. 1, pp. 61-68, 2021. |
| [15] | C. Sin, D. Soares and O. Tavares, “Coursework in industrial doctorates: a worthwhile contribution to students’ training?,” *Higher Education Research & Development,* vol. 40, no. 6, pp. 1298-1312, 2021. |
| [16] | P. Scarfe, K. Watcham, A. Clarke and E. Roesch, “A real-world test of artificial intelligence infiltration of a university examinations system: A “Turing Test” case study,” *PLoS ONE,* vol. 19, no. 6, 2024. |
| [17] | N. Selwyn, Telling Tales on Technology: Qualitative Studies of Technology and Education, London: Routledge, 2020. |
| [18] | M. Theobald, H. Bellhäuser and M. Imhof, “Deadlines don't prevent cramming: Course instruction and individual differences predict learning strategy use and exam performance,” *Learning and Individual Differences,* vol. 87, p. 101994, 2021. |
| [19] | Z. Yu, “A Literature Review on MOOCs Integrated With Learning Analytics,” *Journal of Information Technology Research (JITR),* vol. 14, no. 2, pp. 67-84, 2021. |
| [20] | H. Melissa and K. Peacock, “Replacing power with flexible structure: Implementing flexible deadlines to improve student learning experiences,” *Teaching & Learning Inquiry,* vol. 10, 2022. |
| [21] | J. D. Capelle, K. Senker, S. Fries and A. Grund, “Deadlines make you productive, but what do they do to your motivation? Trajectories in quantity and quality of motivation and study activities among university students as exams approach,” *Frontiers in Psychology,* vol. 14, p. 1224533, 2023. |
| [22] | S. Breese, E. Maicus, M. Peveler and B. Cutler, *Correlation of a flexible late day policy with student stress and programming assignment plagiarism,* 2018. |
| [23] | F. E. V. Castro, J. Leinonen and A. Hellas, “Experiences With and Lessons Learned on Deadlines and Submission Behavior,” in *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research*, Koli, 2022. |
| [24] | Y. Wang, N. Luo and J. Zhou, “Mining Assignment Submission Time to Detect At-Risk Students with Peer Information,” in *International Educational Data Mining Society*, Durham, 2022. |
| [25] | A. B. H. M. Maliki, A. M. M. Isa, M. N. Nazarudin, M. R. Abdullah, S. M. Mat-Rasid and R. M. Musa, “Patterns in assignment submission times: Analysis of factors contributing to undergraduate students’ commitment to core-curriculum related course,” *Heliyon,* vol. 10, no. 5, pp. 1-11, 2024. |
| [26] | D. Ariely and K. Wertenbroch, “Procrastination, Deadlines, and Performance: Self-Control by Precommitment,” *Psychological Science,* vol. 13, no. 3, pp. 219-224, 2002. |
| [27] | T. A. Pychyl, Solving the procrastination puzzle: A concise guide to strategies for change, TarcherPerigee, 2013. |
| [28] | M.-C. Chiu, E. Moss and T. Richards, “Effect of Deadlines on Student Submission Timelines and Success in a Fully-Online Self-Paced Course,” in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education*, Portland, 2024. |
| [29] | A. Perry, Isn't it about Time?: How to Stop Putting Things Off and Get on with Your Life, Worth Publishing, 2008. |
| [30] | A. J. Howell, D. C. Watson, R. A. Powell and K. Buro, “Academic procrastination: The pattern and correlates of behavioural postponement,” *Personality and Individual Differences,* vol. 40, no. 8, pp. 1519-1530, 2006. |
| [31] | A. Ostermaier, “Incentives for students: effects of certificates and deadlines on student performance,” *Journal of Business Economics,* vol. 88, pp. 65-96, 2018. |
| [32] | S. H. Edwards, J. Martin and C. A. Shaffer, “Examining Classroom Interventions to Reduce Procrastination,” in *ITiCSE '15: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, New York, 2015. |
| [33] | J. Lu, M. Schmidt, M. Lee and R. Huang, “Usability research in educational technology: a state‑of‑the‑art systematic review,” *Association for Educational Communications and Technology,* vol. 70, no. 6, pp. 1951-1992, 2022. |
| [34] | J. R. Lewis and J. Sauro, “Usability and User Experience: Design and Evaluation,” in *Handbook of human factors and ergonomics*, John Wiley & Sons, Inc., 2021, pp. 972-1088. |
| [35] | J. Sauer, S. Andreas and S. Schmutz, “Usability, user experience and accessibility: towards an integrative model,” *Ergonomics,* vol. 63, no. 10, pp. 1207-1220, 2020. |
| [36] | T. Issa and P. Isaias, “Usability and Human–Computer Interaction (HCI),” in *Sustainable Design*, London, Springer, 2022, pp. 19-36. |
| [37] | P. Vlachogianni and N. Tselios, “Perceived usability evaluation of educational technology using the System Usability Scale (SUS): A systematic review,” *Journal of Research on Technology in Education,* vol. 54, no. 3, pp. 392-409, 2021. |
| [38] | A. A. Tawfik, J. Gatewood, J. J. Gish-Lieberman and A. J. Hampton, “Toward a Definition of Learning Experience Design,” *Tech Know Learn,* vol. 27, p. 309–334, 2022. |
| [39] | N. Kostaras and M. Xenos, “Assessing Educational Web-site Usability using Heuristic Evaluation Rules,” in *Proceedings of 11th Panhellenic Conference in Informatics*, Patras, 2007. |
| [40] | A. Abdulfaraj and A. Steele, “Detailed Usability Heuristics: A Breakdown of Usability Heuristics to Enhance Comprehension for Novice Evaluators,” in *International Conference on Human-Computer Interaction*, Heraklion, 2020. |
| [41] | O. A. Rotaru, S. Vert, R. Vasiu and D. Andone, “Standardised Questionnaires in Usability Evaluation. Applying Standardised Usability Questionnaires in Digital Products Evaluation,” *International Conference on Information and Software Technologies,* pp. 39-48, 2020. |
| [42] | T. Chen, L. Peng, B. Jing, C. Wu, J. Yang and G. Cong, “The Impact of the COVID-19 Pandemic on User Experience with Online Education Platforms in China,” *Sustainability,* vol. 12, no. 18, p. 7329, 2020. |
| [43] | S. Djamasbi, D. M. Strong, E. V. Wilson and C. Ruiz, “Designing and Testing User-Centric Systems with both User Experience and Design Science Research Principles,” in *Twenty-second Americas Conference on Information Systems*, San Diego, 2016. |
| [44] | I. Maslov, S. Nikou and P. Hansen, “Exploring user experience of learning management system,” *International Journal of Information and Learning Technology,* vol. 38, no. 4, pp. 343-363, 2021. |
| [45] | E. d. Kock, J. v. Biljon and A. Botha, “User experience of academic staff in the use of a learning management system tool,” in *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, Johannesburg, 2016. |
| [46] | C. Geisler, Designing for user engagement on the web: 10 basic principles, Routledge, 2013. |
| [47] | P. Zaharias and P. Christopher, “Quality management of learning management systems: A user experience perspective,” *Current Issues in Emerging eLearning,* vol. 3, no. 1, p. 5, 2016. |
| [48] | A. Minichiello, J. R. Hood and D. S. Harkness, “Bringing user experience design to bear on STEM education: A narrative literature review,” *Journal for STEM Education Research,* vol. 1, pp. 7-33, 2018. |
| [49] | C. Hillmann, UX for XR, Singapore: Apress, 2021. |
| [50] | Bradley and V. Malcolm, “Learning Management System (LMS) Use with Online Instruction,” *International Journal of Technology in Education,* vol. 4, no. 1, pp. 68-92, 2021. |
| [51] | C. Sulun, “The Evolution and Diffusion of Learning Management Systems: Tha Case of Canvas LMS,” in *Driving educational change: Innovations in action*, Columbus, 2018, pp. 86-102. |
| [52] | S. A. Raza, W. Qazi and J. Salam, “Social Isolation and Acceptance of the Learning Management System (LMS) in the time of COVID-19 Pandemic: An Expansion of the UTAUT Model,” *Journal of Educational Computing Research,* vol. 59, no. 2, pp. 183-208, 2020. |
| [53] | K. Falcone, *A case study of faculty experience and preference of using blackboard and canvas LMS (Doctoral dissertation, University of Phoenix),* 2018. |
| [54] | A. Saleh, H. Abuaddous, I. Alansari and O. Enaizan, “The evaluation of user experience on learning management systems using UEQ,” *International Journal of Emerging Technologies in Learning (iJET),* vol. 17, no. 7, pp. 145-162, 2022. |
| [55] | C. B. Mpungose and S. B. Khoza, “Postgraduate Students’ Experiences on the Use of Moodle and Canvas Learning Management System,” *Technology, Knowledge and Learning,* vol. 27, no. 1, pp. 1-16, 2020. |
| [56] | K. Kamata, W. Suamuang and S. Suksakulchai, “A Comparison Study of Student’s Achievement Using Assignment Log,” *International Journal of Information and Education Technology,* vol. 12, no. 12, pp. 1443-1447, 2022. |
| [57] | D. Rigby, “Management Tools and Techniques: A Survey,” *California Management Review,* vol. 43, no. 2, pp. 138-160, 2001. |
| [58] | G. Fletcher, Management and Visualisation, Routledge, 2023. |
| [59] | R. O. Pelumi, O. I. Olateju, I. I. Oshin and W. G. Odukaiye, “Work Breakdown Structure and Timely Delivery of Project.(A Study of Lagos State University),” *Lasu Journal of Transport & Logistics,* vol. 4, no. 1, pp. 1-10, 2022. |
| [60] | S. A. Murray, J. Davis, H. D. Shuler, E. C. Spencer and A. H. Jr., “Time management for STEMM students during the continuing pandemic,” *Trends in biochemical sciences,* vol. 47, no. 4, pp. 279-283, 2022. |
| [61] | M. Olsson, P. Mozelius and J. Collin, “Visualisation and Gamification of e‑Learning and Programming Education,” *The Electronic Journal of e-Learning,* vol. 13, no. 6, pp. 441-454, 2015. |
| [62] | A. Nussbaumer, C. Steiner and D. Albert, “Visualisation Tools for Supporting Self-Regulated Learning through Exploiting Competence Structures,” in *Proceedings of I-KNOW '08 and I-MEDIA '08*, Graz, 2008. |
| [63] | S. R. Emmons, R. P. Light and K. Börner, “MOOC Visual Analytics: Empowering Students,Teachers, Researchers, and Platform Developers ofMassively Open Online Courses,” *Journal of the Association for Information Science and Technology,* vol. 68, no. 10, pp. 2297-2521, 2017. |
| [64] | G. S. Ramaswami, T. Susnjak and A. Mathrani, “Capitalizing on Learning Analytics Dashboard for Maximizing Student Outcomes,” in *2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, Melbourne, 2019. |
| [65] | K. Almalki, O. Alharbi, W. Al-Ahmadi and M. Aljohani, “Anti-procrastination Online Tool for Graduate Students Based on the Pomodoro Technique,” in *Learning and Collaboration Technologies. Human and Technology Ecosystems: 7th International Conference*, Copenhagen, 2020. |
| [66] | M. Kokoç, G. Akçapınar and M. N. Hasnine, “Unfolding Students’ Online Assignment Submission Behavioral Patterns using Temporal Learning Analytics,” *Educational Technology & Society,* vol. 24, no. 1, pp. 223-235, 2021. |
| [67] | Y. Yang, D. Hooshyar, M. Pedaste, M. Wang, Y.-M. Huang and H. Lim, “Prediction of students’ procrastination behaviour through their submission behavioural pattern in online learning,” *Journal of Ambient Intelligence and Humanized Computing,* pp. 1-18, 2020. |
| [68] | D. S. Fleur, W. v. d. Bos and B. Bredeweg, “Learning Analytics Dashboard for Motivation and Performance,” in *International Conference on Intelligent Tutoring Systems*, Athens, 2020. |
| [69] | G. W. Britain, *Design Analytics Dashboards to Support Students and Instructors,* Doctoral dissertation, 2020. |
| [70] | K. Graham, “TechMatters: Getting on the “Kanban”-wagon: Using KanbanFlow for Time and Project Management,” *Loex Quarterly,* vol. 43, no. 3, pp. 4-7, 2016. |
| [71] | J. McLean and R. Canham, “Managing the Electronic Resources Lifecycle with Kanban,” *Open Information Science,* vol. 2, no. 1, pp. 34-43, 2018. |
| [72] | C. Zito, *Project Journal: What it is, why use it, and more,* Dubai: Heriot-Watt University, 2024. |
| [73] | H. Zantout, *Introducing the Project Journal (for F21RP and F21MP),* Dubai: Heriot-Watt University, 2024. |
| [74] | V. Grinshkun and E. Osipovskaya, “Teaching in the Fourth Industrial Revolution:,” in *Proceedings of the 4th International Conference on Informatization of Education and E-learning Methodology:*, Krasnoyarsk, 2020. |
| [75] | S. Elayyan, “The future of education according to the fourth industrial revolution,” *Journal of Educational Technology,* vol. 4, no. 1, pp. 23-30, 2021. |
| [76] | S. Scepanovič, “The Fourth Industrial Revolution and Education,” in *2019 8th Mediterranean Conference on Embedded Computing (MECO)*, Budva, 2019. |
| [77] | M. A. Mulyani, S. Yusuf, P. Siregar, J. Nurihsan, A. Razzaq and M. Anshari, “Fourth Industrial Revolution and Educational Challenges,” in *2021 International Conference on Information Management and Technology (ICIMTech)*, Jakarta, 2021. |
| [78] | A. Oke and F. A. P. Fernandes, “Innovations in Teaching and Learning: Exploring the Perceptions of the Education Sector on the 4th Industrial Revolution (4IR),” *Journal of Open Innovation: Technology, Market, and Complexity,* vol. 6, no. 2, p. 31, 2020. |
| [79] | A. Villafiorita, Introduction to Software Project Management, CRC Press, 2014. |
| [80] | A. Dingle, Software Essentials, CRC Press, 2014. |
| [81] | D. Emari, “How to use agile method for solo projects,” 2023. [Online]. Available: https://dania.hashnode.dev/how-to-use-agile-method-for-solo-projects. [Accessed 7 April 2024]. |
| [82] | J. N. Brito, C. Rebelo and M. A. Brito, “Scrum solo application in a project with a strong integration component,” in *36th IBIMA*, Granada, 2020. |
| [83] | A. Nyström, *Agile Solo - Defining and Evaluating an Agile Software Development Process for a Single Software Developer,* Chalmers University of Technology, 2011. |
| [84] | Lucidchart, “Scrum for one: A tutorial on adapting Agile Scrum methodology for individuals,” [Online]. Available: https://www.lucidchart.com/blog/scrum-for-one. [Accessed 7 April 2024]. |
| [85] | Lucidchart, “Scrum board vs Kanban board: Choosing the right Agile tool,” [Online]. Available: https://www.lucidchart.com/blog/kanban-vs-scrum?anonId=0.e0e2beae18ed5236a1d&sessionDate=2024-04-13T01%3A47%3A45.679Z&sessionId=0.7d2503c818ed5236a1f. [Accessed 7 April 2024]. |
| [86] | L. Nielsen, Personas-user focused design, London: Springer, 2013. |
| [87] | B. Nandi, “Nextjs, Express & Redis on Microservice Architecture,” Medium, 10 June 2023. [Online]. Available: https://medium.com/@bjnandi/nextjs-express-redis-on-microservice-architecture-3fd7daafcd68. [Accessed 27 November 2024]. |
| [88] | O. Alhadreti, “Assessing Academics’ Perceptions of Blackboard Usability Using SUS and CSUQ: A Case Study during the COVID-19 Pandemic,” *International Journal of Human–Computer Interaction,* vol. 37, no. 11, pp. 1003-1015, 2021. |
| [89] | G. Rakoczi, “The Future of LMS platforms: what will be the challenges, roles and opportunities for decades to come?,” in *The Future of Education 13th Edition 2023*, Austria, 2023. |
| [90] | C. Gackenheimer, “What Is React?,” in *Introduction to React*, 2015, pp. 1-20. |
| [91] | J. Rachowicz, “When, How, And Why Use Node.js as Your Backend? [2022 Update],” 2023. [Online]. Available: https://www.netguru.com/blog/node-js-backend. [Accessed 7 April 2024]. |
| [92] | MongoDB, “MongoDB with Node.js,” [Online]. Available: https://www.mongodb.com/languages/mongodb-with-nodejs#:~:text=js-,The%20MongoDB%20Node.,easily%20work%20with%20their%20data.. [Accessed 7 April 2024]. |
| [93] | Figma, “What is prototyping,” [Online]. Available: https://www.figma.com/resource-library/what-is-prototyping/. [Accessed 7 April 2024]. |
| [94] | R. Orozco, C. Benjet, G. Borges, M. F. M. Arce, D. F. Ito, C. Fleiz and J. A. Villatoro, “Child and Adolescent Psychiatry and Mental Health,” *Child and Adolescent Psychiatry and Mental Health,* vol. 12, no. 9, pp. 1-10, 2018. |
| [95] | Y. Sharma, A. Suri, R. Sijariya and L. Jindal, “Role of education 4.0 in innovative curriculum practices and digital literacy– A bibliometric approach,” *E-Learning and Digital Media,* p. 20427530231221073, 2023. |
| [96] | O. Zawacki-Richter and C. Latchem, “Exploring four decades of research in Computers & Education,” *Computers & Education,* vol. 122, pp. 136-152, 2018. |
| [97] | M. Yorke, P. Bridges and H. Woolf, “Mark distributions and marking practices in UK higher education: some challenging issues,” *Active learning in higher education,* vol. 1, no. 1, pp. 7-27, 2000. |
| [98] | P. Bridges, A. Cooper, P. Evanson, C. Haines, D. Jenkins, D. Scurry, H. Woolf and M. Yorke, “Coursework Marks High, Examination Marks Low: Discuss,” *Assessment & Evaluation in Higher Education,* vol. 27, no. 1, pp. 35-48, 2002. |
| [99] | C. Kayembe and D. Nel-Sanders, “Challenges and opportunities for education in the Fourth Industrial Revolution,” *African Journal of Public Affairs,* vol. 11, no. 3, pp. 79-94, 2019. |
| [100] | S. Z. Mirsharapovna, S. M. Shadjalilovna, A. A. Kakhramonovich and K. R. Malikovna, “Pros and Cons of Computer Technologies in Education,” *Texas Journal of Multidisciplinary Studies,* vol. 14, pp. 26-29, 2022. |
| [101] | J. Millman, “Assessment in Higher Education (Book Review),” *Journal of Higher Education (Columbus),* vol. 49, no. 5, p. 517, 1978. |
| [102] | Cloudflare, “What is a reverse proxy? | Proxy servers explained,” [Online]. Available: https://www.cloudflare.com/en-gb/learning/cdn/glossary/reverse-proxy/. [Accessed 14 April 2024]. |
| [103] | G. Gibbs and L. Lucas, “Coursework Assessment, Class Size and Student Perfor-mance: 1984–1994,” *Journal of Further and Higher Education,,* vol. 21, no. 2, pp. 183-192, 1997. |
| [104] | LinkedIn, “How do you manage SDLC risks?,” [Online]. Available: https://www.linkedin.com/advice/0/how-do-you-manage-sdlc-risks-skills-software-development-life-c#:~:text=Risks%20are%20uncertainties%20that%20can,a%20systematic%20and%20proactive%20approach.. [Accessed 8 April 2024]. |
| [105] | J. Shaturaev, “Economies and Management as A Result of The Fourth Industrial Revolution: An Education Perspective,” *Indonesian Journal of Educational Research and Technology,* vol. 3, no. 1, pp. 51-58, 2023. |
| [106] | M. Thomlinson, N. Challis and M. Robinson, “Coursework, what should be its nature and assessment weight?,” *MSOR Connections,* vol. 10, no. 2, pp. 31-34, 2010. |

# Bibliography

Bridges, P. et al., 2002. Coursework Marks High, Examination Marks Low: Discuss. *Assessment & Evaluation in Higher Education,* 27(1), pp. 35-48.

Cloudflare, n.d. *What is a reverse proxy? | Proxy servers explained.* [Online]   
Available at: https://www.cloudflare.com/en-gb/learning/cdn/glossary/reverse-proxy/ [Accessed 14 April 2024].

Gibbs, G. & Lucas, L., 1997. Coursework Assessment, Class Size and Student Perfor-mance: 1984–1994. *Journal of Further and Higher Education,,* 21(2), pp. 183-192.

Kayembe, C. & Nel-Sanders, D., 2019. Challenges and opportunities for education in the Fourth Industrial Revolution. *African Journal of Public Affairs,* 11(3), pp. 79-94.

LinkedIn, n.d. *How do you manage SDLC risks?.* [Online]   
Available at: https://www.linkedin.com/advice/0/how-do-you-manage-sdlc-risks-skills-software-development-life-c#:~:text=Risks%20are%20uncertainties%20that%20can,a%20systematic%20and%20proactive%20approach. [Accessed 8 April 2024].

Millman, J., 1978. Assessment in Higher Education (Book Review). *Journal of Higher Education (Columbus),* 49(5), p. 517.

Mirsharapovna, S. Z., Shadjalilovna, S. M., Kakhramonovich, A. A. & Malikovna, K. R., 2022. Pros and Cons of Computer Technologies in Education. *Texas Journal of Multidisciplinary Studies,* Volume 14, pp. 26-29.

Sharma, Y., Suri, A., Sijariya, R. & Jindal, L., 2023. Role of education 4.0 in innovative curriculum practices and digital literacy– A bibliometric approach. *E-Learning and Digital Media,* p. 20427530231221073.

Shaturaev, J., 2023. Economies and Management as A Result of The Fourth Industrial Revolution: An Education Perspective. *Indonesian Journal of Educational Research and Technology,* 3(1), pp. 51-58.

Thomlinson, M., Challis, N. & Robinson, M., 2010. Coursework, what should be its nature and assessment weight?. *MSOR Connections,* 10(2), pp. 31-34.

Yorke, M., Bridges, P. & Woolf, H., 2000. Mark distributions and marking practices in UK higher education: some challenging issues. *Active learning in higher education,* 1(1), pp. 7-27.

Zawacki-Richter, O. & Latchem, C., 2018. Exploring four decades of research in Computers & Education. *Computers & Education,* Volume 122, pp. 136-152.

# Appendix A: Heuristic Evaluation

Below is a list of Nielsen’s heuristics created by [40]:

1. Visibility of system status
   1. State: The current state of the system and available actions
   2. Location: Where the user is currently
   3. Progress: How much more is left to complete a task
   4. Closure: Notification upon task completion
2. Match between system and the real
   1. Understandability: Use content that the target audience can understand easily
   2. Natural and logical order: Present information in steps that are followed normally in the real-world. For example, e-shopping
   3. Appropriateness: The content should be appropriate for the target audience
3. User control and freedom
   1. Reversibility: All actions should be reversible - recover deleted files
   2. Emergency exit: Exit undesirable situations without extensive procedures
   3. Informing users: Inform the user about the critical action he/she is taking
4. Consistency and standards
   1. Consistency: Element usage should be the same throughout the system
   2. Standards: Use knowledge of previous similar systems and apply it
5. Error prevention
   1. Instructions: Clear instructions and requirements for performing tasks
   2. Constraints: Do not allow certain input from the user like numbers for name
   3. Confirmation: To avoid unintentional actions, the system should ask before executing serious and irreversible actions
   4. Notification: Notify users about critical changes and updates
   5. Autosaving: If the system fails, all user data will be lost. To prevent this, the system should autosave time-consuming data
   6. Flexible inputs: Allow alternate ways of entering input to enhance flexibility
   7. Defaults: Add default states that are preferred by people and inform them
6. Recognition rather than recall
   1. Availability: Make information visible at all times so that users do not need to memorise it. For example, directions on streets
   2. Suggestions: Provide accurate suggestions to users because they might be unfamiliar with the system and want they want
7. Flexibility and efficiency of use
   1. Flexibility: The system is usable for all kinds of users
   2. Efficiency: Do not add unnecessary steps to compete a task
8. Aesthetic and minimalist design
   1. Aesthetic: If the system is aesthetically pleasing, users will perceive it as usable and ignore minor usability issues
   2. Organisation: Organise sections in a sensible order by categorising similar elements together and separating different sections
   3. Simplicity: Simple and uncluttered interface with only necessary content
9. Help users recognise, diagnose, and recover from errors
   1. Recognising errors: When an error occurs, the system should clearly display the error message and the user should be able to understand that an error has occurred
   2. Understanding errors: Location and reason of the error
   3. Recovering from errors: Display instructions and steps for resolving the error
10. Help and documentation
    1. Help: There should be a contact point for providing help to users
    2. Documentation: Easy to follow user guides, FAQs, and tutorials

# Appendix B: SUS Survey

Figure 1 shows the questions of an SUS survey [88].

|  |
| --- |
| A white box with black text  Description automatically generatedAppendix B – Figure 1 The standard SUS Survey [88] |

# Appendix C: Professional, Legal, Ethical and Social Issues

**Professional Issues**

The work done during this project will be compliant with the British Computer Society (BCM) Code of Conduct. The development of the website and its testing will follow software engineering practices, ensuring a professional development process. Moreover, all code will be properly organised, commented, and documented for increased readability, and the website will adhere to usability standards. Any outside work referred to will be clearly referenced. Software and tools will only be utilised if their licenses permit it.

**Legal Issues**

All relevant laws and regulations will be followed for this project. The data collected from the usability test will be obtained by participants’ consent and will be completely anonymised to protect their privacy. Throughout the project, all software and other copyrighted materials will be used with permission from their respective licenses. After the development of the website, the developer will ensure that only authorised individuals have access to the website.

**Ethical Issues**

Since data will be obtained from human subjects during testing, all participants will be provided with information sheets describing the project and the testing procedure. Additionally, they will be provided with consent forms explaining how their anonymised data will be used. The participants will be given the freedom to withdraw from the test at any point. The research will be transparent and will adhere to General Data Protection Regulation Compliance. Finally, the website will also ensure user privacy with the help of authorisation.

**Social Issues**

This project aims to reduce the negative social impact of deadline mismanagement practiced by students. The website will monitor and visualise students’ progress to motivate them, help them combat procrastination, and prevent heightened stress levels caused by delaying assignments out of a lack of self-regulation. Additionally, the website will aim to promote positive user experiences by following usability guidelines and helping students manage their coursework efficiently.

# Appendix D: Prototype

|  |
| --- |
| A screenshot of a computer  Description automatically generated  Appendix D – Figure 1 Rejected Prototype in EditorA screenshot of a computer  Description automatically generated  Appendix D – Figure 2 Rejected Prototype |
| A screenshot of a computer  Description automatically generated  Appendix D – Figure 3 Final Prototype in Editor |
| Appendix D – Figure 4 Home Page |
| Appendix D – Figure 5 Help Page |
| Appendix D – Figure 6 Contact Us Page |
| Appendix D – Figure 7 Contact Us Submission Confirmation |
| Appendix D – Figure 8 Student Login |
| Appendix D – Figure 9 Staff Login 1 |
| Appendix D – Figure 10 Staff Login 2 |
| Appendix D – Figure 11 Forget Password 1 |
| Appendix D – Figure 12 Forget Password 2 |
| Appendix D – Figure 13 Forget Password 3 |
| Appendix D – Figure 14 Forget Password 4 |
| Appendix D – Figure 15 Password Reset Confirmation |
| Appendix D – Figure 16 Student Dashboard |
| Appendix D – Figure 17 Student Courses |
| Appendix D – Figure 18 Student Course Page |
| Appendix D – Figure 19 Student Coursework View Page |
| Appendix D – Figure 20 Student Timeline |
| Appendix D – Figure 21 Subtask Modal on Student Timeline |
| Appendix D – Figure 22 Student Kanban Board |
| Appendix D – Figure 23 Student Account |
| Appendix D – Figure 24 View Student Profile |
| Appendix D – Figure 25 Student Settings |
| Appendix D – Figure 26 Student Notifications |
| Appendix D – Figure 27 Student Contact Us |
| Appendix D – Figure 28 Student Contact Us Confirmation |
| Appendix D – Figure 29 Student Help Page |
| Appendix D – Figure 30 Staff Dashboard |
| Appendix D – Figure 31 Staff Course Page |
| Appendix D – Figure 32 Staff Coursework View Page |
| Appendix D – Figure 33 Staff Add New Coursework 1 |
| Appendix D – Figure 34 Staff Add New Coursework 2 |
| Appendix D – Figure 35 Staff Add New Coursework 3 |
| Appendix D – Figure 36 Staff Add New Coursework Confirmation |
| Appendix D – Figure 37 Students List |
| Appendix D – Figure 38 View Individual Student Submission Status |

# Appendix E: Final Website

|  |
| --- |
| Appendix E – Figure 1 Home Page |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 2 Help Page |
| A screenshot of a contact form  Description automatically generated  Appendix E – Figure 3 Contact Us Page |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 4 Contact Us Confirmation Page |
| A screenshot of a login form  Description automatically generated  Appendix E – Figure 5 Student Login |
| A screenshot of a login form  Description automatically generated  Appendix E – Figure 6 Staff Login 1 |
| A screenshot of a login form  Description automatically generated  Appendix E – Figure 7 Staff login 2 |
| A screenshot of a computer login  Description automatically generated  Appendix E – Figure 8 Forgot Password 1 |
| A screenshot of a login page  Description automatically generated  Appendix E – Figure 9 Forgot Password 2 |
| A screenshot of a login form  Description automatically generated  Appendix E – Figure 10 Forgot Password 3 |
| A screenshot of a login form  Description automatically generated  Appendix E – Figure 11 Forgot Password 4 |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 12 Password Rest Confirmation |
| A screenshot of a computer screen  Description automatically generated  Appendix E – Figure 13 Student Dashboard |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 14 Student Courses |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 15 Student Course Page |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 16 Student Coursework View Page |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 17 Student Locked Subtaaks |
| A screenshot of a calendar  Description automatically generated  Appendix E – Figure 18 Student Timeline |
| A screenshot of a research paper  Description automatically generated  Appendix E – Figure 19 Student Timeline Modal Subtask View |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 20 Student Kanban Board |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 21 Student Notifications |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 22 Student Account |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 23 Student View Profile |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 24 Student Edit Profile |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 25 Student Settings |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 26 Student Rewards |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 27 Student Contact Us |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 28 Student Help Page |
| A screenshot of a computer screen  Description automatically generated  Appendix E – Figure 29 Staff Dashboard |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 30 Staff Course Page |
| Appendix E – Figure 31 Staff Coursework View Page |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 32 Staff Coursework Deletion Confirmation |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 33 Staff Add Coursework 1 |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 34 Staff Add Coursework 2 |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 35 Staff Add Coursework Review |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 36 Staff Coursework Added Confirmation |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 37 Staff View Subtask Timeline Modal |
| A screenshot of a computer  Description automatically generated  Appendix E – Figure 38 Students List |

# Appendix F: Questionnaires

|  |
| --- |
| Appendix F – Figure 1 System Usability Scale (SUS) Survey |
| Appendix F – Figure 2 Post-Usability Questionnaire |

# Appendix G: Testing Documents

|  |
| --- |
| Appendix G – Figure 1 Task Sheet for Students 1 |
| Appendix G – Figure 2 Task Sheet for Students 2 |
| Appendix G – Figure 3 Information Sheet for Students 1 |
| Appendix G – Figure 4 Information Sheet for Students 2 |
| Appendix G – Figure 5 Consent Form for Participants |

# Appendix H: Signed Consent Forms

|  |
| --- |
| Appendix H – Figure 1 Consent Form 1 |
| Appendix H – Figure 2 Consent Form 2 |
| Appendix H – Figure 3 Consent Form 3 |
| Appendix H – Figure 4 Consent Form 4 |
| Appendix H – Figure 5 Consent Form 5 |
| Appendix H – Figure 6 Consent Form 6 |
| Appendix H – Figure 7 Consent Form 7 |
| Appendix H – Figure 8 Consent Form 8 |
| Appendix H – Figure 9 Consent Form 9 |
| Appendix H – Figure 10 Consent Form 10 |
| Appendix H – Figure 11 Consent Form 11 |
| Appendix H – Figure 12 Consent Form 12 |

# Appendix I: Evaluation Results

**Student Usability Test Observation**

Participant 1

Task 1: Easy

Task 2: Unable to find the submit button

Noticed subtask colour change

Only submitted one subtask

Task 3: Easy

Task 4: Went to dashboard first

Did not know it had to be dragged

Thought it had to be loaded when clicked

Task 5: Asked if this is the progress

Task 6: Easy

Task 7: Went to account for logout

Task 8: Tried logging in to change password

Got lost

Additional observation: Preferred going to the CW through the dashboard instead of courses

About participant: 4th year cs student, was in a hurry

Participant 2 (8 mins)

Task 1: Easy

Task 2: Could not find the submit button

Task 3: Easy

Task 4: Took a while but completed

Task 5: Easy

Task 6: Easy

Task 7: Easy

Task 8: Could not find forgot password

Went to help but read the FAQ about forgot username. Clicked on the link and reached contact us. She was entering the details. I had to stop her and guide her. Took 2.15 for this task

About Participant: Was in a hurry. Uses Macbook so was slow on windows. Wanted better structured tasks and instructions about dragging (on board). CS 4th year student

Participant 3

Task 1: Easy. Noticed name

Task 2: Easy. Noticed locked subtasks and understood the purpose

Task 3: Went to coursework page instead of timeline

Repeated the task but viewed F21DF instead of F21SF

Task 4: Went to coursework page again instead of board

Repeated the task but failed to drag

Task 5: Easy

Task 6: Did it separately

Task 7: Easy

Task 8: Logged in again instead of using forget password

Completed it

About participant: Also, in a hurry. CS 4th year student

Participant 4

Task 1: Easy

Task 2: Noticed all changes except stars

Task 3: Did not check the subtasks

Task 4: Dragged and dropped halfway through. Needed help

Task 5: Did not notice the bar

Task 6: Went to coursework page. Some difficulty in finding edit profile but found it

Task 7: Easy

Task 8: Easy

Entered email for verification code

About participant: CS 4th year student. Read the sheets

Participant 5

Task 1: Easy

Task 2: Found it hard to find the submit button

Thought the button was for resubmit only

Noticed 100% complete

Task 3: Opened view subtask

Found it later

Task 4: Went to dashboard to do it

Then board

Was clicking on it

Figured out dragging without help

Task 5: Did not notice the bar

Task 6: Went to settings first

Task 7: Easy

Task 8: Easy

Thought email and phone number, both had to be entered so got confused

Additional Observation: Did struggle slightly

About participant: CS 4th year student. Read the sheets

Participant 6

Task 1: Easy

Task 2: Easy

Task 3: Went to course page instead of timeline

Went to timeline but did not view it

Task 4: Clicked on them

Found the drag but struggles slightly

Task 5: Did not notice the bar

Task 6: Easy

Task 7: Easy

Task 8: Easy

Additional Observation: Used back buttons of browser instead of interacting with the menu

About participant: CS 3rd year student

Participant 7

Task 1: Made mistake in password. Tried to re-enter password without closing alert

Task 2: Went to coursework from side bar instead of using the CW links on the dashboard

Easy

Noticed the locked subtasks

Noticed the progress

Task 3: Went to courses instead of timeline

Could not find timeline

Viewed F21DF instead of F21SF

Task 4: Clicked on the subtask

Thought he must go back to courses

Couldn’t figure out dragging. Needed help

Task 5: Did not notice the progress bar

Task 6: Easy

Task 7: Easy

Task 8: Went to help and followed the correct FAQ

He thought email was skipped but later realised it was an option to select either email or phone number

Needed the show button for confirm password

About participant: CS 3rd year student. Read the sheets before test

Additional Observation: Seemed to have some eye vision problem

Participant 8

Task 1: Easy

Task 2: Took long and viewed subtasks

Did not notice any changes

Was submitting same subtask again

Task 3: Went to dashboard

Then to courses -> Software Engineering

Then to timeline

Task 4: Went to dashboard

Then to courses -> database

Submitted subtasks there

Repeated the task after I told him. He clicked on the subtasks but could not figure out the dragging part

Task 5: Did not notice the progress bar

Task 6: Did the tasks separately

Task 7: Easy

Task 8: Went to help, then dashboard, then to help, checked the correct FAQ but clicked on the link without reading and reached contact us. Asked him to read. Completed

Additional Observation: Did not use the logo to go back to the main page to change password. Went for browser back button

About participant: Engineering student. Not good with computers. Made basic mistakes. Did not read any of the sheets provided before the test. Was conscious because I was making notes and many times stopped in between.

Participant 9

Task 1: Easy

Task 2: Went to courses 2 times. Confused. Then selected the CW from dashboard

Tried to submit next subtask without closing the alert

Task 3: Thought he had to click on the colour key Software Engineering. Then he realised and viewed the correct subtask

Task 4: Clicked on the subtask

Could not figure out

Task 5: Did not notice the bar

Task 6: Easy

Task 7: Easy

Task 8: Easy

About participant: Robotics student

Participant 10

Task 1: Easy, Liked the green colour and the design

Task 2: Went to courses instead of dashboard

Found it straightforward

Got confused with the button working Re/Submit

Noticed the updated work progress

Task 3: Went to courses first

Task 4: First clicked on the tasks

Found the dragging part himself but took time

Task 5: Went to e-commerce

Then went to notifications

Then saw the progress

Task 6: First went to settings

Task 7: Easy but expected a log out confirmation

Task 8: Was about to log in again but saw the forgot password option

Entered email for username

Needed show/hide to see password on new password page

About participant: Robotics student

Most common mistakes:

* Show/hide button on new password page
* Dragging on board page
* Should have course code and course name everywhere
* People did not use the help section when they got stuck
* Directly clicked the links for solution rather than reading the FAQ properly
* Most did not notice the progress bar
* Many ignored the alert window
* Some expected change password to be in settings or profile
* Most preferred using the browser back buttons
* Logout and login was easy
* Board was difficult
* Most found the Re/Submit button confusing. Most did not even notice the colour change. No one noticed the stars. Some explanation must be provided on the page
* Most were in a hurry
* Most were not confident enough to explore the website on their own, may be conscious because of me
* Most did not read the sheets and tasks before the task

My mistakes:

* Forgot to record participant 2
* It was too late to perform testing. Most of them were in a hurry since it was time for them to go home
* The tasks on the task sheet were not clear. I failed to structure them properly
* Should have recruited all participants from same year of study and program

**Student SUS Score**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Participant Number | Q1  + | Q2  - | Q3  + | Q4  - | Q5  + | Q6  - | Q7  + | Q8  - | Q9  + | Q10  - |
| 1 | 3 | 3 | 3 | 2 | 3 | 1 | 4 | 1 | 3 | 3 |
| 2 | 4 | 3 | 2 | 3 | 4 | 2 | 3 | 2 | 3 | 4 |
| 3 | 5 | 2 | 4 | 1 | 4 | 3 | 5 | 4 | 5 | 1 |
| 4 | 5 | 1 | 5 | 2 | 5 | 1 | 5 | 1 | 5 | 1 |
| 5 | 4 | 1 | 5 | 1 | 5 | 1 | 4 | 1 | 5 | 1 |
| 6 | 5 | 1 | 5 | 1 | 5 | 1 | 4 | 1 | 5 | 1 |
| 7 | 4 | 2 | 4 | 1 | 4 | 1 | 4 | 1 | 4 | 2 |
| 9 | 4 | 2 | 4 | 1 | 5 | 2 | 5 | 2 | 4 | 1 |
| 10 | 4 | 3 | 4 | 1 | 3 | 4 | 5 | 1 | 4 | 1 |
| 8 | 3 | 3 | 4 | 2 | 3 | 2 | 4 | 1 | 3 | 2 |

|  |  |  |
| --- | --- | --- |
| Participant Number | Sum of points | Final (sum \* 2.5) |
| 1 | 26 | 65 |
| 2 | 22 | 55 |
| 3 | 32 | 80 |
| 4 | 39 | 97.5 |
| 5 | 38 | 95 |
| 6 | 39 | 97.5 |
| 7 | 33 | 82.5 |
| 9 | 34 | 85 |
| 10 | 30 | 75 |
| 8 | 27 | 67.5 |
| Average SUS Score | | 80 (Good) |

**Student Post-Usability Questionnaire Results**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | How would you rate the colour palette of this website? | Were you able to clearly differentiate between links and other content? | Was the size of the content, menus, drop-downs, and other features appropriate? | If applicable, describe any major challenge you may have had during the tasks? | We would love to get your suggestions to help us improve our website. |
| 1 | 8 | Yes | No | I had an issue figuring out that I had to drag for completing a task and marking it as complete. tasks could be more well defined | Adding more context menus and keeping the change password in the settings |
| 2 | 7 | No | Yes | the sidebar (navigation bar) could be a bit smaller the courses page could have the code of the courses loved the timeline and progress features!! | less compact |
| 3 | 9 | Yes | No | The drag and drop feature was not explicitly mentioned and the update your password task is a bit confusing but other than that it was good. | Having a simple home page after logging in is key. I suggest having more pages with each having appropriate content. |
| 4 | 10 | Yes | Yes |  |  |
| 5 | 8 | Yes | Yes | the drop-downs, i thought the system would allocate it to done based on the progress. | system would allocate the drop-down of the tasks to done based on the progress |
| 6 | 8 | Yes | Yes | task drag and drop | tutorial for beginners who are not familiar with canvas |
| 7 | 7 | Yes | Yes |  | Being able to switch to the tasks from the boards section. |
| 9 | 4 | Yes | Yes | dragging the tasks from the to do list | instruction on how to move tasks from the to do list to the completed |
| 10 | 10 | Yes | Yes | N/A | password icon to see the pass, inconsistent fonts and sizes, so if you could work on that, it should be good! |
| 8 | 6 | Yes | No | not major challenge but the overall look and UI for the website can be improved to make it look more appealing | same as above |
| Average | 7.7 | 9 Yes and 1 No | 7 Yes and 3 No | Drag: 4 |  |
| Update pass: 1 |
| Task definition: 1 |

Suggestions Summary

|  |
| --- |
| password change in settings |
| more menu options |
| smaller side bar |
| less compact (less clutter), simple dashboard by adding more pages |
| add course code to all course pages |
| tutorial for beginners that don’t use canvas |
| instructions for drag and drop |
| show hide in confirm pass |
| consistent fonts and sizes |
| overall look and UI |

**Staff Testing Questions**

1. Feedback, opinions, and suggestions for the student and staff portal
2. Kanban board
3. Usability, look and feel
4. Most and least liked features
5. Forget password
6. Share button
7. Would you use it
8. Is it cluttered/ too busy?
9. Visualisations
10. Gamification like leaderboard

**Staff 1**

Student portal

* Canvas doesn’t have anything to prioritise tasks for students

Staff portal

* More work for them because of breaking down work
* May be staff needs to rethink for some cw
* For many cw (60-70%) could be added workload
* Less flexibility for older students because they have lots of other things to do
* Very difficult to breakdown written cws into subtasks
* It is better for students to learn time and project management like staff breaking into sprints
* The process of staff dividing cw into subtasks is not very practical in real-life because students need to do it themselves at that stage
* Good for younger students but not for 3rd or 4th year undergraduate students and postgraduate students

Visualisations will not be shown if there is only one subtask. There will be no progress because there are no multiple subtasks, just one subtask. Having pomodoro technique might be helpful as it will record the time spent on the subtask but this might no reflect reality if the student is rushing to complete the subtask or cw because at that moment, students just focus on the deadline and not on any other aspect

Major improvement over canvas for students as it shows the cw deadliens on the dashboard so the student doesn’t need to remember the dates; reducing memory load

Way better than Vision (system used by university before canvas), because on vision, the student had to remember the dates

Share button: sharing with students is not a good idea. However, the staff at HW need to sometimes export stats from Canvas for some internal purposes. So, the share button here can export the statistics as csv for internal purposes. Also, it might not be useful for all courses

Usability: very familiar for canvas and non-canvas users. Intuitive. Easy navigation. Would not need help for first time users. Everything is visible. Good. Nothing bad. Look and feel is good. All information is visible. No unnecessary clutter.

Kanban board could have a different icon to make it clearer. Students are not familiar with project management tools.

Gamification module like leaderboard or things similar to it would be good but it depends on the student’s personality. However, it should not show or specify the rank. Just show you are in top 3%, etc… It can be optional for students so that they can turn it on or off. Don’t mention names and anonymise it because it can raise anxiety. The ranking system is something that even schools are eliminating, so the students that join university (this generation) will not be familiar or comfortable with this ranking system.

Forgot password: the screens show the step number which is good because the website is giving proper feedback to the user after every step

Additional feedback:

* Push notifications to the user on email one week before the deadline
* Regarding cws, this platform is good. Canvas has many more features but cant be compared because this is a prototype and perhaps canvas was something similar when it had started

Additional features:

* Staff can send reminders to students from the students page. There can be a fixed template inside the system. Each row can have a tick box. When more than one row/student is selected, the button to send reminder will be activated(not greyed out) and a reminder to those students will be sent to start working

This is an mvp and covers all the basic features.

Good for students. Extra load for staff

**Staff 2**

More refinement. Idea is good

Challenge is doing the full check at once. Breaking will make more sense. Improved version of canvas

Usability: easy but need more guidelines

Gamification module: dashboard and timeline is enough. Depends on who you are targeting. Keep it serious because we are preparing adults not kids

Kanban board: simple but has no instructions. Turning education into circus (not happen). kanban board for each cw

Dashboard: clear. May be add more details. May be have to define titles

Staff portal:

* More helpful for students
* Disagree with babysitting students. Staff should not plan the cw and create subtasks
* Send reminder would not recommend
* Don’t want to micromanage
* Extra work for staff and micromanaging
* Will not help with understanding if the work is done by the student. Gen AI tools can do this

Overall:

* Helps students become independent because they are managing it themselves. 9/10. Reduce stress
* For staff, 3 or 4 / 10
* Most like: timeline and kanban (kanban if organised properly)
* Drawback for faculty: investing too much time and micromanaging
* Staff dashboard should focus on marking, who is late, etc… rather than focusing on students completing their cws
* Usability: good
* Student portal: process; Staff portal: results
* Share button will motivate students. It will be helpful
* Forget password: it is standard

# Appendix J: JIRA Sprint Planning

|  |
| --- |
| Appendix J – Figure 1 Sprint 1 Planning – Figma Prototype |
| Appendix J – Figure 2 Sprint 2 Planning – Figma Prototype Cont. |
| Appendix J – Figure 3 Sprint 3 Planning – Figma Prototype |
| Appendix J – Figure 4 Sprint 4 Planning – Front-End Development |
| Appendix J – Figure 5 Sprint 5 Planning – Front-End Development Cont. |
| Appendix J – Figure 6 Sprint 6 Planning – Database |
| Appendix J – Figure 7 Sprint 7 Planning – Student Back End |
| Appendix J – Figure 8 Sprint 8 Planning – Staff Back End |
| Appendix J – Figure 9 Sprint 9 Planning – Staff Testing and Report |

# Appendix K: JIRA Sprint Retrospective

|  |
| --- |
| Appendix K – Figure 1 Sprint 1 Retrospective |
| Appendix K – Figure 2 Sprint 2 Retrospective |
| Appendix K – Figure 3 Sprint 3 Retrospective |
| Appendix K – Figure 4 Sprint 4 Retrospective |
| Appendix K – Figure 5 Sprint 5 Retrospective |
| Appendix K – Figure 6 Sprint 6 Retrospective |
| Appendix K – Figure 7 Sprint 7 Retrospective |
| Appendix K – Figure 8 Sprint 8 Retrospective |
| Appendix K – Figure 9 Sprint 9 Retrospective |

# Appendix L: JIRA Graphs

|  |
| --- |
| Appendix L – Figure 1 JIRA Backlog |
| Appendix L – Figure 2 JIRA Kanban Board |
| A graph showing a growth  Description automatically generated with medium confidence  Appendix L – Figure 3 JIRA Cumulative Flow Diagram for Coursework Wizard |
| A graph with lines and numbers  Description automatically generated with medium confidence  Appendix L – Figure 4 JIRA Burn Up Chart for Sprint 9 |
| A graph on a white background  Description automatically generated  Appendix L – Figure 5 JIRA Burn Up Chart for Sprint 8 |
| A diagram of a graph  Description automatically generated  Appendix L – Figure 6 JIRA Burn Down Chart for Sprint 8 |

# Appendix M: Personas

|  |
| --- |
| Appendix M – Figure 1 Student Persona 1 |
| Appendix M – Figure 2 Student Persona 2 |
| Appendix M – Figure 3 Student Persona 3 |
| Appendix M – Figure 4 Student Persona 4 |
| Appendix M – Figure 5 Staff Persona 1 |
| Appendix M – Figure 6 Student Persona 2 |
| Appendix M – Figure 7 Student Persona 3 |