

EduAi

AI-POWERED LEARNING ASSISTANT

Project 2

Second Semester 2023/2024

Supervised by: Dr. Mohammad Abushariah

Name	Student ID
Mohannad Reyad Albujoq	0201263
Mohammad Shaher Alabed	0203292
Abedalmajeed abedalatif Alajarmah	0203169
Adnan Samer Alkharouf	0205413

Abstract

EduAI aims to revolutionize the online learning experience for university students by incorporating advanced artificial intelligence technologies. Our project leverages a ChatGPT-powered chatbot and advance behavior tracking mechanisms to create a dynamic, personalized educational environment. By understanding and addressing the individual differences among students, EduAI facilitates tailored learning experiences that adapt to each student's unique needs and preferences. This innovative approach not only enhances engagement and interaction but also promotes more effective learning outcomes. The integration of these cutting-edge features into existing online learning platforms represents a significant step forward in the evolution of digital education, making it more responsive, inclusive, and effective for diverse student populations.

Acknowledgment

We begin by expressing our deep gratitude to our academic supervisor, Dr. Mohammad Abushariah for granting us the opportunity to engage in the remarkable project named "UniSwap" for the University of Jordan students. Her guidance not only steered us but also instilled in us the drive to pursue comprehensive research and broaden our horizons. Furthermore, we wish to convey our heartfelt thanks to our parents and friends, whose immense support was instrumental in completing this project within the stipulated timeframe. Our sincerest appreciation goes to everyone who contributed to the realization of this project's success. Their unwavering belief in us and their continuous encouragement have consistently fueled our motivation throughout this enriching journey.

Table of Contents

1. Chapter One: Introduction	- 10 -
1.1 Project Motivation	- 10 -
1.2 Problem Statement	- 10 -
1.3 Project Aim and Objectives	- 11 -
1.4 Project Scope	- 11 -
1.5 Software and Hardware Requirements	- 11 -
1.6 Project Limitations	- 12 -
1.7 Project Expected Output	- 12 -
1.8 Project Schedule	- 13 -
1.8.1 PERT Diagram	- 14 -
1.8.2 GANTT Chart	- 14 -
1.8.3 Staff Allocation Chart.....	- 15 -
1.9 Report Outline.....	- 15 -
2. Chapter Two: Related Existing System	- 18 -
2.1 Introduction	- 18 -
2.2 Existing Systems	- 18 -
2.2.1 Udemy	- 18 -
2.2.2 Coursera.....	- 19 -
2.3 Overall Problems of Existing Systems	- 19 -
2.4 Overall Solution Approach	- 20 -
2.5 Summary.....	- 20 -
3. Chapter Three: System Requirements Engineering and Analysis	- 21 -
3.1 Introduction	- 21 -
3.2 Feasibility Study	- 21 -
3.2.1 Technical Feasibility	- 21 -
3.2.2 Operational Feasibility.....	- 21 -
3.2.3 Economic Feasibility	- 22 -
3.3 Requirements Elicitation Techniques (Questionnaire).....	- 23 -
3.4 Targeted Users	- 26 -
3.5 Functional Requirements Definition	- 27 -
3.6 Functional Requirements Specification	- 30 -

3.7 Non-Functional Requirements.....	- 46 -
3.8 Summary.....	- 46 -
4. Chapter Four: System Design	- 47 -
 4.1 Introduction	- 47 -
 4.2 Context Diagram	- 47 -
 4.3 Entity Relationship Diagram (ERD).....	- 48 -
 4.4 UML Use Case Diagram.....	- 49 -
 4.5 UML Sequence Diagram	- 53 -
 4.5.1 student sequence	- 53 -
 4.5.1 instructor sequence	- 61 -
 4.5.1 admin sequence.....	- 76 -
 4.6 UML Class diagram.....	- 98 -
 4.7 graphical user interface design / low fidelity prototype	- 99 -
Chapter five: System Implementation.....	- 105 -
 5.1 database implementation.....	- 105 -
 5.2 Graphical user interface implementation	- 110 -
 5.3.1 Student interface	- 117 -
 5.3.1 Instructor interface.....	- 120 -
 5.3.1 admin interface	- 135 -
Chapter six: Testing and installation.....	- 147 -
 6.1 Testing.....	- 147 -
 6.1.1 Heuristic Evaluation.....	- 147 -
 6.1.2 Cooperative Evaluation	- 154 -
 6.2 Installation	- 169 -
 6.3 User manual.....	- 170 -
 7.1 Conclusions	- 227 -
 7.3 Strengths	- 227 -
 7.3 Weaknesses.....	- 227 -
 7.4 Future Direction	- 227 -
References	- 228 -

<i>Table 1: Abbreviations</i>	- 9 -
<i>Table 2: Project Schedule</i>	- 13 -
<i>Table 3: Staff Allocation</i>	- 15 -
<i>Table 4: Economic Feasibility</i>	- 22 -
<i>Table 5: Functional Requirements Definition</i>	- 27 -
<i>Table 6: Register Tabular form</i>	- 30 -
<i>Table 7: Non-Functional Requirements</i>	- 46 -
<i>Table 8: Visibility of System Status</i>	- 147 -
<i>Table 9: Match Between System and the Real World</i>	- 148 -
<i>Table 10: User Control and Freedom</i>	- 148 -
<i>Table 11: Consistency and Standards</i>	- 149 -
<i>Table 12: Help Users Recognize, Diagnose, and Recover from Errors</i>	- 150 -
<i>Table 13: Error Prevention</i>	- 150 -
<i>Table 14: Recognition Rather Than Recall</i>	- 151 -
<i>Table 15: Flexibility and Minimalist Design</i>	- 152 -
<i>Table 16: Aesthetic and Minimalist Design</i>	- 152 -
<i>Table 17: Help and Documentation</i>	- 153 -
<i>Table 18: Participants Details</i>	- 154 -
<i>Table 19: Cooperative Evaluation 1</i>	- 155 -
<i>Table 20: Cooperative Evaluation 2</i>	- 157 -
<i>Table 21: Cooperative Evaluation 3</i>	- 159 -
<i>Table 22: Cooperative Evaluation 4</i>	- 160 -
<i>Table 23: Cooperative Evaluation 5</i>	- 162 -
<i>Table 24: Cooperative Evaluation 6</i>	- 164 -
<i>Table 25: Task Completion Times in Minutes and Seconds</i>	- 166 -
<i>Table 26: Participants Responses to the Post-Test Questionnaire</i>	- 167 -
<i>Table 27: Test (Post-test Questionnaire)</i>	- 168 -

Figure 1: Pert Diagram	- 14 -
Figure 2: Gantt chart	- 14 -
Figure 3: Questionnaire figure 1	- 23 -
Figure 4: Questionnaire figure 2	- 23 -
Figure 5: Questionnaire figure 3	- 24 -
Figure 6: Questionnaire figure 4	- 24 -
Figure 7: Questionnaire figure 5	- 25 -
Figure 8: Questionnaire figure 6	- 25 -
Figure 9: context diagram	- 47 -
Figure 10: ERD	- 48 -
Figure 11: UML use case diagram: EduAI System	- 49 -
Figure 12: UML use case diagram: 1	- 49 -
Figure 13: UML use case diagram: 2	- 50 -
Figure 14: UML use case diagram: 3	- 51 -
Figure 15: UML use case diagram: 4	- 52 -
Figure 16: Sequence Diagram: Student Delete Course	- 53 -
Figure 17: Sequence Diagram: Sequence Diagram: Student Display Course	- 54 -
Figure 18: Sequence Diagram: Sequence Diagram: Student Attempt Quiz	- 55 -
Figure 19: Sequence Diagram: Sequence Diagram: Student Display Courses List	- 56 -
Figure 20: Sequence Diagram: Student Dashboard	- 56 -
Figure 21: Sequence Diagram: Student Display Courses Content	- 57 -
Figure 22: Sequence Diagram: Student Enroll Course	- 57 -
Figure 23: Sequence Diagram: Student Display My Courses	- 58 -
Figure 24: Sequence Diagram: Student Edit Profile	- 59 -
Figure 25: Sequence Diagram: Student Register	- 60 -
Figure 26: Sequence Diagram: Instructor Add New Section	- 61 -
Figure 27: Sequence Diagram: Instructor Edit Profile	- 62 -
Figure 28: Sequence Diagram: Instructor Add New YouTube Link	- 63 -
Figure 29: Sequence Diagram: Instructor Chat with GPT	- 63 -
Figure 30: Sequence Diagram: Instructor Display Courses	- 64 -
Figure 31: Sequence Diagram: Instructor Dashboard	- 64 -
Figure 32: Sequence Diagram: Instructor Delete Course	- 65 -
Figure 33: Sequence Diagram: Instructor Delete Item	- 66 -
Figure 34: Sequence Diagram: Instructor Delete Section	- 66 -
Figure 35: Sequence Diagram: Instructor Download Material	- 67 -
Figure 36: Sequence Diagram: Instructor Edit Material	- 67 -
Figure 37: Sequence Diagram: Instructor Edit Course	- 68 -
Figure 38: Sequence Diagram: Instructor Edit Post	- 68 -
Figure 39: Sequence Diagram: Instructor Edit Question	- 69 -
Figure 40: Sequence Diagram: Instructor Edit Quiz	- 69 -
Figure 41: Sequence Diagram: Instructor Add New Section	- 70 -
Figure 42: Sequence Diagram: Instructor Edit Item Position	- 70 -
Figure 43: Sequence Diagram: Instructor Add Quiz Question	- 71 -
Figure 44: Sequence Diagram: Instructor Display Quizzes List	- 71 -
Figure 45: Sequence Diagram: Instructor Edit Section Position	- 72 -
Figure 46: Sequence Diagram: Instructor: Display Student Quiz Result	- 72 -
Figure 47: Sequence Diagram: Instructor Display Courses Content	- 73 -
Figure 48: Sequence Diagram: Instructor Add Material	- 73 -
Figure 49: Sequence Diagram: Instructor Add New Course	- 74 -
Figure 50: Sequence Diagram: Instructor Add New Post	- 74 -
Figure 51: Sequence Diagram: Instructor Add New Quiz	- 75 -

Figure 52:Sequence Diagram: Admin Edit Profile	- 76 -
Figure 53:Sequence Diagram: User Contact us	- 77 -
Figure 54:Sequence Diagram: Admin Delete Course	- 78 -
Figure 55:Sequence Diagram: Admin Delete Instructor	- 79 -
Figure 56:Sequence Diagram: Admin Delete Student	- 80 -
Figure 57:Sequence Diagram: Admin Display Courses List	- 81 -
Figure 58:Sequence Diagram: Admin Edit Instructor	- 82 -
Figure 59:Sequence Diagram: Admin Edit Student List	- 83 -
Figure 60:Sequence Diagram: User Change Language	- 84 -
Figure 61:Sequence Diagram: User Filter By Entity	- 84 -
Figure 62:Sequence Diagram: User Login	- 85 -
Figure 63:Sequence Diagram: User Logout	- 86 -
Figure 64:Sequence Diagram: User Maximize Or Minimize Website	- 86 -
Figure 65:Sequence Diagram: User Display privacy and policy	- 87 -
Figure 66:Sequence Diagram: Admin Publish Toggle	- 88 -
Figure 67:Sequence Diagram: User Reset Password	- 89 -
Figure 68:Sequence Diagram: User Search	- 90 -
Figure 69:Sequence Diagram: User Selection Page	- 91 -
Figure 70:Sequence Diagram: User Show Password	- 92 -
Figure 71:Sequence Diagram: User Term Of Use	- 93 -
Figure 72:Sequence Diagram: Admin view Student	- 94 -
Figure 73:Sequence Diagram: Admin Add Instructor	- 95 -
Figure 74:Sequence Diagram: Admin Add Student	- 96 -
Figure 75:Sequence Diagram: Admin Course Overview	- 97 -
Figure 76:Sequence Diagram: Admin Dashboard	- 97 -
Figure 77: UML Class Diagram	- 98 -
Figure 78: low fidelity: Home Page	- 99 -
Figure 79: low fidelity: Register Page	- 100 -
Figure 80: low fidelity: Login Page	- 101 -
Figure 81: low fidelity: Students Pages	- 101 -
Figure 82:low fidelity: Admin Pages	- 102 -
Figure 83: low fidelity: Instructor Pages	- 103 -
Figure 84:Database Design	- 104 -
Figure 85:Database Implementation 1	- 105 -
Figure 86:Database Implementation 2	- 105 -
Figure 87:Database Implementation 3	- 105 -
Figure 88:Database Implementation 4	- 105 -
Figure 89:Database Implementation 5	- 106 -
Figure 90:Database Implementation 6	- 106 -
Figure 91:Database Implementation 7	- 106 -
Figure 92:Database Implementation 8	- 106 -
Figure 93:Database Implementation 9	- 106 -
Figure 94:Database Implementation 10	- 107 -
Figure 95:Database Implementation 11	- 107 -
Figure 96:Database Implementation 12	- 107 -
Figure 97:Database Implementation 13	- 107 -
Figure 98:Database Implementation 14	- 108 -
Figure 99:Database Implementation 15	- 108 -
Figure 100:Database Implementation 16	- 108 -
Figure 101:Database Implementation 17	- 108 -
Figure 102:Database Implementation 18	- 109 -

<i>Figure 103: Database Implementation 19</i>	- 109 -
<i>Figure 104: Graphical user interface implementation 1</i>	- 110 -
<i>Figure 105: Graphical user interface implementation 2</i>	- 111 -
<i>Figure 106: Graphical user interface implementation 3</i>	- 112 -
<i>Figure 107: Graphical user interface implementation 4</i>	- 113 -
<i>Figure 108: Graphical user interface implementation 5</i>	- 114 -
<i>Figure 109: Graphical user interface implementation 6</i>	- 115 -
<i>Figure 110: Graphical user interface implementation 7</i>	- 115 -
<i>Figure 111: Graphical user interface implementation 8</i>	- 116 -
<i>Figure 112: Student GUI 1</i>	- 117 -
<i>Figure 113: Student GUI 2</i>	- 118 -
<i>Figure 114: Student GUI 3</i>	- 118 -
<i>Figure 115: Student GUI 4</i>	- 119 -
<i>Figure 116: Instructor GUI 1</i>	- 120 -
<i>Figure 117: Instructor GUI 2</i>	- 120 -
<i>Figure 118: Instructor GUI 3</i>	- 121 -
<i>Figure 119: Instructor GUI 4</i>	- 121 -
<i>Figure 120: Instructor GUI 5</i>	- 122 -
<i>Figure 121: Instructor GUI 6</i>	- 122 -
<i>Figure 122: Instructor GUI 7</i>	- 123 -
<i>Figure 123: Instructor GUI 8</i>	- 123 -
<i>Figure 124: Instructor GUI 9</i>	- 124 -
<i>Figure 125: Instructor GUI 10</i>	- 124 -
<i>Figure 126: Instructor GUI 11</i>	- 125 -
<i>Figure 127: Instructor GUI 12</i>	- 125 -
<i>Figure 128: Instructor GUI 13</i>	- 126 -
<i>Figure 129: Instructor GUI 14</i>	- 126 -
<i>Figure 130: Instructor GUI 15</i>	- 127 -
<i>Figure 131: Instructor GUI 16</i>	- 127 -
<i>Figure 132: Instructor GUI 17</i>	- 128 -
<i>Figure 133: Instructor GUI 18</i>	- 128 -
<i>Figure 134: Instructor GUI 19</i>	- 129 -
<i>Figure 135: Instructor GUI 20</i>	- 129 -
<i>Figure 136: Instructor GUI 22</i>	- 130 -
<i>Figure 137: Instructor GUI 23</i>	- 130 -
<i>Figure 138: Instructor GUI 24</i>	- 131 -
<i>Figure 139: Instructor GUI 25</i>	- 131 -
<i>Figure 140: Instructor GUI 26</i>	- 132 -
<i>Figure 141: Instructor GUI 27</i>	- 132 -
<i>Figure 142: Instructor GUI 28</i>	- 133 -
<i>Figure 143: Instructor GUI 29</i>	- 133 -
<i>Figure 144: Instructor GUI 30</i>	- 134 -
<i>Figure 145: Instructor GUI 31</i>	- 134 -
<i>Figure 146: Admin GUI 1</i>	- 135 -
<i>Figure 147: Admin GUI 2</i>	- 136 -
<i>Figure 148: Admin GUI 3</i>	- 136 -
<i>Figure 149: Admin GUI 4</i>	- 137 -
<i>Figure 150: Admin GUI 5</i>	- 137 -
<i>Figure 151: Admin GUI 6</i>	- 138 -
<i>Figure 152: Admin GUI 7</i>	- 138 -
<i>Figure 153: Admin GUI 8</i>	- 139 -

Figure 154: Admin GUI 9	- 139 -
Figure 155: Admin GUI 10	- 140 -
Figure 156: Admin GUI 11	- 140 -
Figure 157: Admin GUI 12	- 141 -
Figure 158: Admin GUI 13	- 141 -
Figure 159: Admin GUI 14	- 142 -
Figure 160: Admin GUI 15	- 142 -
Figure 161: Admin GUI 16	- 143 -
Figure 162: Admin GUI 17	- 143 -
Figure 163: Admin GUI 18	- 144 -
Figure 164: Admin GUI 19	- 144 -
Figure 165: Admin GUI 20	- 145 -
Figure 166: Admin GUI 21	- 145 -
Figure 167: Admin GUI 22	- 146 -
Figure 168: Admin GUI 23	- 146 -
Figure 169: User guide 1	- 170 -
Figure 170: User guide 2	- 171 -
Figure 171: User guide 3	- 172 -
Figure 172: User guide 4	- 173 -
Figure 173: User guide 7	- 174 -
Figure 174: User guide 5	- 174 -
Figure 175: User guide 6	- 174 -
Figure 176: User guide 8	- 175 -
Figure 177: User guide 9	- 176 -
Figure 178: User guide 10	- 177 -
Figure 179: User guide 11	- 178 -
Figure 180: User guide 12	- 179 -
Figure 181: User guide 13	- 180 -
Figure 182: User guide 14	- 181 -
Figure 183: User guide 15	- 182 -
Figure 184: User guide 16	- 183 -
Figure 185: User guide 17	- 184 -
Figure 186: User guide 18	- 185 -
Figure 187: User guide 19	- 186 -
Figure 188: User guide 20	- 187 -
Figure 189: User guide 21	- 188 -
Figure 190: User guide 22	- 189 -
Figure 191: User guide 23	- 190 -
Figure 192: User guide 24	- 191 -
Figure 193: User guide 25	- 192 -
Figure 194: User guide 26	- 193 -
Figure 195: User guide 27	- 194 -
Figure 196: User guide 28	- 195 -
Figure 197: User guide 29	- 196 -
Figure 198: User guide 30	- 197 -
Figure 199: User guide 31	- 198 -
Figure 200: User guide 32	- 199 -
Figure 201: User guide 33	- 200 -
Figure 202: User guide 34	- 201 -
Figure 203: User guide 35	- 202 -
Figure 204: User guide 36	- 203 -

<i>Figure 205: User guide 37</i>	- 204 -
<i>Figure 206: User guide 38</i>	- 205 -
<i>Figure 207: User guide 39</i>	- 206 -
<i>Figure 208: User guide 40</i>	- 207 -
<i>Figure 209: User guide 41</i>	- 208 -
<i>Figure 210: User guide 42</i>	- 209 -
<i>Figure 211: User guide 43</i>	- 210 -
<i>Figure 212: User guide 44</i>	- 211 -
<i>Figure 213: User guide 45</i>	- 212 -
<i>Figure 214: User guide 46</i>	- 213 -
<i>Figure 215: User guide 47</i>	- 214 -
<i>Figure 216: User guide 48</i>	- 215 -
<i>Figure 217: User guide 49</i>	- 216 -
<i>Figure 218: User guide 50</i>	- 217 -
<i>Figure 219: User guide 51</i>	- 218 -
<i>Figure 220: User guide 52</i>	- 219 -
<i>Figure 221: User guide 53</i>	- 220 -
<i>Figure 222: User guide 54</i>	- 221 -
<i>Figure 223: User guide 55</i>	- 222 -
<i>Figure 224: User guide 56</i>	- 223 -
<i>Figure 225: User guide 57</i>	- 224 -
<i>Figure 226: User guide 58</i>	- 225 -
<i>Figure 227: User guide 59</i>	- 226 -

Table 1: Abbreviations

Abbreviations	definition
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
API	Application Programming Interfaces
GUI	Graphcal User Interface
ROI	Return on Investment
NPV	Net Present Value

1. Chapter One: Introduction

1.1 Project Motivation

The motivation behind our project lies in addressing the individual differences among university students, acknowledging that each student learns in a distinct way. In response to the competitive and demanding nature of academic pursuits, our web-based platform seeks to provide not just courses but a holistic learning environment. The incorporation of a ChatGPT-powered chatbot aims to offer real-time support, enhancing user engagement and facilitating a more effective learning journey.

1.2 Problem Statement

Many online learning websites treat all students the same, making it challenging for university students with different learning styles. This one-size-fits-all approach often leads to students feeling disconnected and struggling to stay engaged in their courses. The lack of personalized assistance and quick answers to their questions can be frustrating. Our project recognizes these issues and aims to solve the problem by adding a helpful chatbot and tracking how students use the website. This way, we want to make sure each student gets the support they need and enjoys a more personalized and effective learning experience.

1.3 Project Aim and Objectives

The primary aim of our project is to enhance the online learning experience for university students by introducing a dynamic and personalized approach through the integration of a ChatGPT-powered chatbot and behavior tracking mechanisms and solving individual differences between university students.

1.4 Project Scope

The scope of our project, titled "EduAI" is defined by the integration of innovative features into the online learning platform, aiming to create a more dynamic and personalized educational experience for university students.

1.5 Software and Hardware Requirements

- Hardware: Any Smart Mobile Device or Computer Device.
- Software: Any Web Browser.

1.6 Project Limitations

Our project, "EduAi," aims to make online learning better, but it's important to know some things we can't do. First, not everyone might want to use the new features we're adding, like the chatbot and behavior tracking. Some students might find it hard to use them. Second, while we try our best to understand how students learn, the chatbot might not always get it right, especially if students don't type their questions in a way it understands. Third, we depend on another tool called ChatGPT, and if that tool has issues or changes, it might affect how our chatbot works.

1.7 Project Expected Output

What we hope to achieve is an online platform "EduAi" and it will be a friendly website where students can learn in their own way. We're adding a helpful chat friend, and the website will remember how each student likes to learn. So, every student gets a special learning experience. The website will be easy to use, with lots of interesting courses, and the chat friend will be there to help students whenever they need it. EduAi wants to make learning online easy and fun for everyone.

1.8 Project Schedule

Table 2:Project Schedule

Task	Description	Start Time	Duration	Dependencies	Staff
T1	Introduction	16/OCT /2023	7 Days	NONE	All Staff
T2	Technical, Operational, and Schedule Feasibility	23/OCT /2023	7 Days	T1, M1	Abdelmajeed Mohannad
T3	Economic Feasibility	24/OCT /2023	7 Days	T1, M1	Abdelmajeed Mohannad
T4	Identify System Stakeholders and Requirements Elicitation	16/OCT/ 2023	8 Days	None	All staff
T5	Specifications, Functional, and non-functional Requirements, Domain. & Wireframe	23/OCT/ 2023	20 Days	T4, M2	Abdelmajeed Mohannad Adnan
T6	System analysis	11/NOV/2023	14 Days	T2, T3, T5, M3, M4	Mohammed Adnan
T7	Architecture Design	24/Nov /2023	8 Days	T6, M5	Abdelmajeed Mohammed
T8	Implementation	20/FEB/ 2024	60 Days	T6, T7, M6	All Staff
T9	Documenting User Manual	20/APR /2024	10 Days	T8, M7	All Staff

1.8.1 PERT Diagram

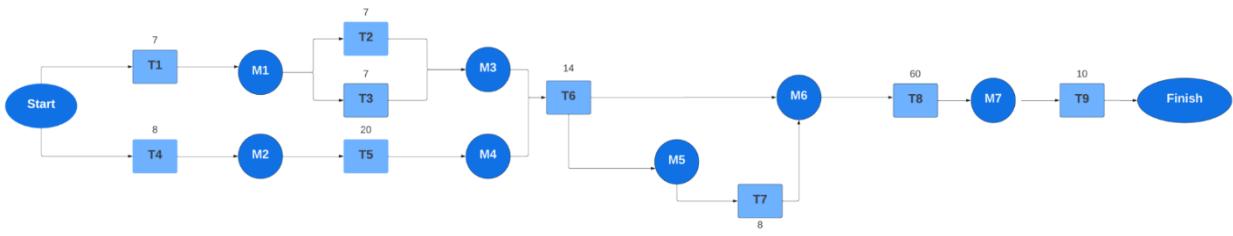


Figure 1: Pert Diagram

1.8.2 GANTT Chart

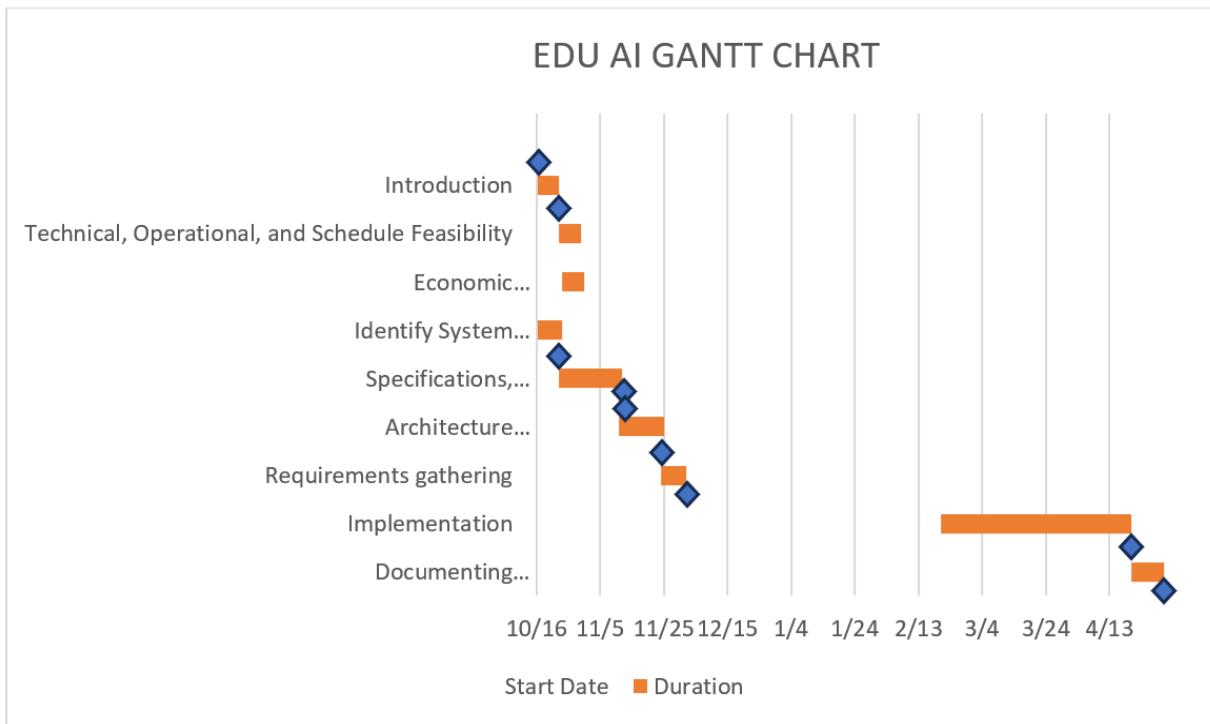


Figure 2: Gantt chart

1.8.3 Staff Allocation Chart

Table 3: Staff Allocation

Weeks Interval	0-1	1-2	2-5	5-7	7-8	8-19
Mohannad	T1&T4	T2&T3	T5			T8&T9
Adnan	T1&T4		T5	T6		T8&T9
Mohammad	T1&T4			T6	T7	T8&T9
Abedalmajeed	T1&T4	T2&T3				

1.9 Report Outline

Chapter one:

shows the project motivation in Section 1.1, in additional the problem statement is stated in Section 1.2, Section 1.3 list the project aim and objectives, whereas the project scope is identified in section 1.4, Section 1.5 provide the project software and hardware requirement and Section 1.6, highlights the project limitation, furthermore project expected output is addressed in Section 1.7, whereas the project is scheduled is Section 1.8, the report outline in finally present in Section 1.9.

Chapter two:

show an introduction in Section 2.1, the existing system is elaborated in Section 2.2, in addition the overall problems of existing system are stated in Section 2.3, Section 2.4, over-all solution approach.

Chapter three:

show an introduction in Section 3.1, requirements elicitation techniques are elaborated in Section 3.2, in addition the targeted users are stated in Section 3.3, Section 3.4, Functional requirements definition and functional requirement specification is identified in Section 7 3.5, Section 3.6 highlights the non-functional requirements.

Chapter four:

an introduction in Section 4.1, Context Diagram elaborated in Section 4.2, Section 4.3 provide the Entity Relationship Diagram, 4.4 UML Use Case Diagram, in addition the UML, Sequence Diagram is stated in Section 4.5, Section 4.6 include UML Sequence Diagram, Section, 4.7 Graphical User Interface (GUI) Design and 4.8 Summary.

Chapter five:

This chapter presents all aspects of the system implementation including the database implementation, graphical user interface implementation/high fidelity prototype, and any details pertaining to the implementation tools used.

Chapter six:

This chapter presents the system testing and installation aspects including the heuristic evaluation, cooperative evaluation, system installation, and user manual.

Chapter seven:

This chapter presents the project conclusions and future work including the overall weaknesses and strengths, and future directions.

2. Chapter Two: Related Existing System

2.1 Introduction

In this chapter, we embark on an exploration of existing systems in the online education landscape, drawing parallels with popular platforms like Udemy and innovative technologies such as ChatGPT. By understanding the strengths and weaknesses of these systems, we gain valuable insights to refine our project, "EduAI," and highlight the unique features it brings to the realm of online education.

2.2 Existing Systems

2.2.1 Udemy

Udemy, a popular online learning platform, has played a main role in making education accessible globally. It offers a diverse range of courses taught by industry experts. However, Udemy mostly follows a standardized model, where courses are pre-recorded, and interactions are limited to discussion forums or messaging systems. The platform might lack the personalization required to provide individual learning preferences and address specific queries effectively.

2.2.2 Coursera

Coursera is a famous website for online learning. It has helped many people around the world to learn new things. It has lots of different courses taught by experts from top schools and companies. Unlike Udemy, Coursera has a more organized way of teaching with set schedules and regular homework. Many of its courses are part of bigger learning paths or degree programs, which give a more complete learning experience. But this might not be good for learners who want to learn at their own pace. Coursera tries to balance having a set way of teaching with meeting the different needs of learners, but there might still be some room to do better in meeting individual learning needs and answering questions effectively.

2.3 Overall Problems of Existing Systems

The overarching issues with existing systems lie in their inability to provide to the unique learning needs of individual students. The lack of personalization, dynamic interaction, and real-time support hinders the overall effectiveness of online education platforms. Students often feel disengaged, leading to lower completion rates and decreased satisfaction.

2.4 Overall Solution Approach

Our project, "EduAI" aims to address the shortcomings identified in existing systems by introducing a dynamic and personalized learning experience. The integration of a ChatGPT-powered chatbot and behavior tracking mechanisms seeks to bridge the gap between the one-size-fits-all approach and the diverse learning needs of university students. By learning from the limitations of existing systems, we strive to create an innovative solution that enhances user engagement and fosters effective learning.

2.5 Summary

This chapter has provided an overview of the current state of online education systems, emphasizing their limitations in catering to individual learning styles. The subsequent chapters will delve into the detailed design and implementation of "EduAI," outlining how our project addresses these challenges and contributes to the evolution of online education.

3. Chapter Three: System Requirements Engineering and Analysis

3.1 Introduction

This chapter includes in detail the system requirements and system analysis with a full description of the system's functionality, including the definition and specification for each requirement.

3.2 Feasibility Study

3.2.1 Technical Feasibility

Our software is developed using the following web development tools:

- Front-end: HTML, CSS, JavaScript.
- Back-end-end: PHP (Laravel).
- Database: MySQL.

The Software is developed by:

- Visual Studio code.
- XAMMP.

The software will be available to users on their web browser. These Technical resources will allow us to update and upgrade the system as required.

3.2.2 Operational Feasibility

The program should work for every student at the university Providing a reliable and easy-to-use program for any student. This program will help the student to minimize:

- Time.
- Money.
- Effort.

3.2.3 Economic Feasibility

Table 4: Economic Feasibility

Year	Year 0	Year 1	Year 2	Year 3
Hardware	3000\$	800\$	1000\$	2000\$
Software	3000\$	950\$	800\$	1000\$
DB Server	300\$	800\$	2000\$	2000\$
Internet Connection	200\$	1000\$	1500\$	1500\$
Developers	2000\$	2000\$	1000\$	1500\$
Maintenance cost	5000\$	1500\$	2500\$	3000\$
Total Cost	13500\$	7050\$	8800\$	11000\$
Benefits	0\$	7500\$	12500\$	26500\$
Discount Rate	1	0.955	0.860	0.772
Accumulated Cost	13500\$	20232.75\$	27800.75\$	36292.75\$
Accumulated Benefits	0\$	7162.5\$	17912.5\$	38370.5\$

ROI (Return on Investment) = (Estimated Benefits – Estimated Costs) / (Estimated Costs)

$$= (38370.5 - 36292.75) / 36292.75 = 2077.75 = 0.057 = 5\%.$$

Annual ROI = ROI/ (Project lifetime)

$$\text{Annual ROI} = 0.057 / 3 = 1.9\%.$$

NPV = Accumulated Benefits – Accumulated Costs

$$= 38370.5 - 36292.75 = 2077.75.$$

Payback Period: Software will start gaining profits by the end of the Third year of the software.

3.3 Requirements Elicitation Techniques (Questionnaire)

The survey was sent to a group of university students and asked them about our project.

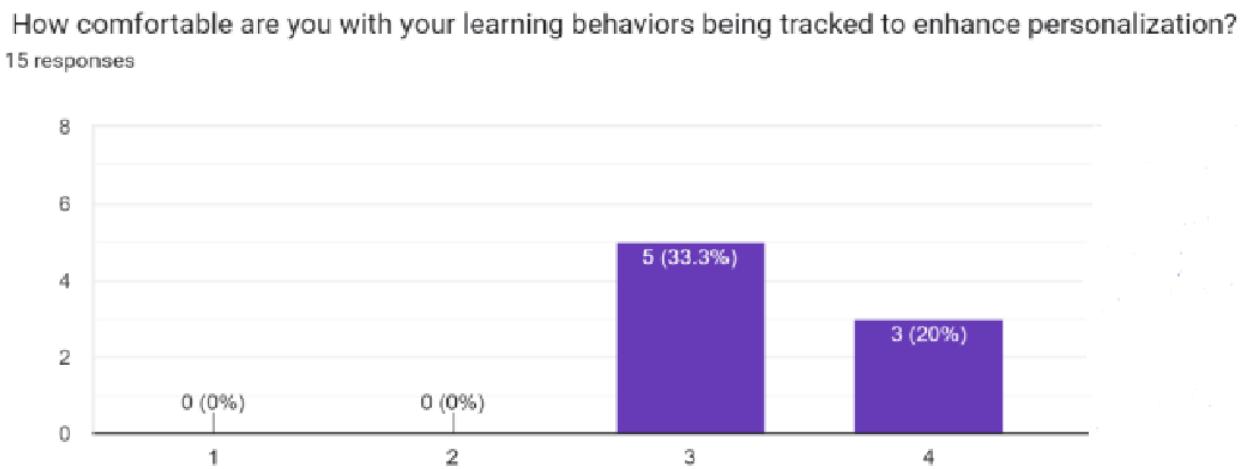


Figure 3: Questionnaire figure 1

What challenges do you face when navigating online courses?

15 responses

It does not take into account individual capabilities

when I want to return to the records

Trustworthiness of these courses

Weakness in obtaining information from the first time (difficulty in learning)

Validity and credibility of the content.

Will I really be better after having specific course.

A lot of spam content

recorded lectures

I don't know what knowledge he gained, but the reason for the lack of evaluation in the courses

Losing motivation, getting lost while navigating through various resources

Figure 4: Questionnaire figure 2

Do you prefer interactive elements (quizzes) in online courses?

15 responses

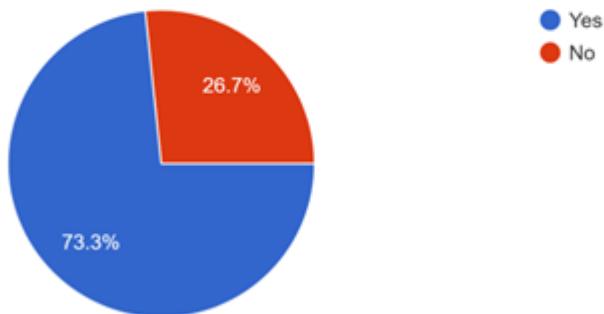


Figure 5: Questionnaire figure 3

Are there any concerns or considerations you have regarding behavior tracking?

15 responses

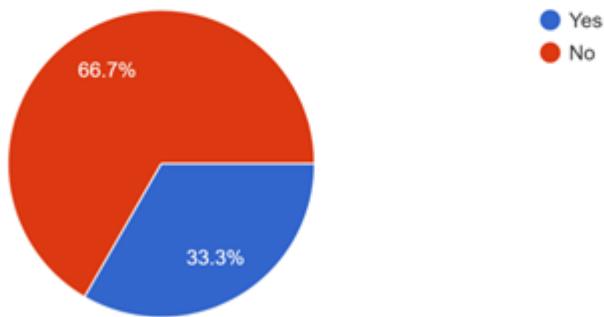


Figure 6: Questionnaire figure 4

Would you find value in having a chatbot to assist you with course-related queries?
15 responses

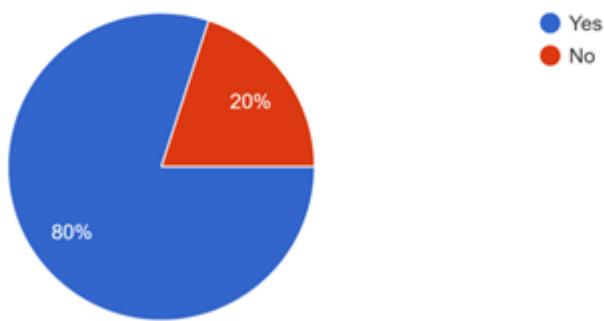


Figure 7: Questionnaire figure 5

How frequently do you engage in online learning activities for your courses?
15 responses

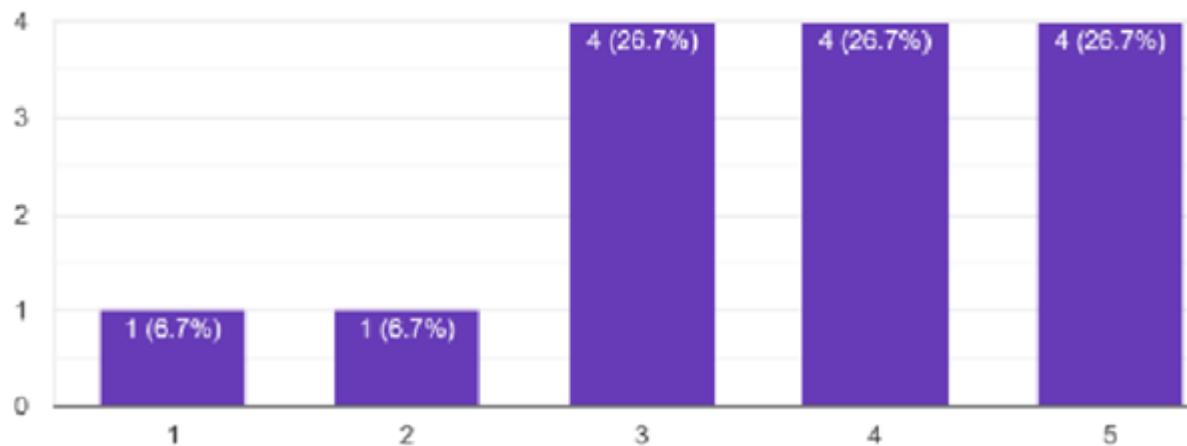


Figure 8: Questionnaire figure 6

3.4 Targeted Users

University Students: Undergraduate and graduate students across various disciplines and academic levels. Those seeking personalized assistance, interactive learning experiences, and quick access to course-related information.

3.5 Functional Requirements Definition

Table 5:Functional Requirements Definition

ID	Requirement Name	Description
1	Register	This feature would allow students to create a new account by providing their personal information.
2	Log in	This feature would allow users to log in to their account using the login page.
3	Forget password	A method that user can used when he/she forget the password.
4	Log out	This feature would allow users to logout of their account and end their session.
5	Greet user	short piece of text that typically displayed the name of the user on the dashboard.
6	Contact us	Feature on a website typically allows users to send a message or inquiry to the EduAi owner.
7	Log in selection	enables users to choose their login form based on their role, such as student, instructor, or admin.
8	Manage course by admin	This requirement entails granting administrators the ability to view, delete, publish, unpublish courses, and access course overviews (number of students and number of students who complete the course).
9	Manage student by admin	This requirement involves enabling administrators to view, add, edit, and delete student profiles within the system.
10	Manage instructor by admin	This requirement involves granting administrators the ability to view, add, edit, and

		delete instructor profiles within the system.
11	Display number of students	This feature allow admin in view the number of students within the system.
12	Display number of instructors	This feature allow admin to display number of instructors within the system.
13	manage course by instructor	allows instructors to view, edit, delete, and add courses within the system.
14	Manage lesson by instructor	Instructors can manage lessons by adding, viewing, editing, deleting sections and items, and downloading items within sections. They can also adjust the position of sections and items using drag-and-drop functionality, facilitating streamlined content organization and customization.
15	Manage course by student	Students can view all courses on the website and manage their own courses by adding them to their list, tracking ongoing progress, viewing completed courses, and filtering courses based on year, department, or title.
16	Mange item	Instructor can add material, books, YouTube links, posts and quiz.
17	Manage Lesson by student	feature allows students to mark their progress by checking boxes as they complete lessons. Additionally, students can download associated files to support their learning journey, also students can attempt quizzes.
18	Profile management	This requirement allows different user roles such as admin, instructor, and student to edit their respective profiles

19	Manage quiz by instructor	Instructor can view quiz list, quiz result, and delete, add, edit quizzes.
20	Attempt quiz	Students can take quizzes and view the result.
21	Display term of use and policy	A specific page on the website that display all term of use for our website.
22	Chat with bot	Instructor and student can chat with bot to improve the quality of learning and teaching methods.
23	Suggest material for students	The chatbot after each attempted quiz will suggest a material for students to improve their information and give a resource to study based on question fall in.

3.6 Functional Requirements Specification

1- Register

Description: Students can create a new account by providing their personal information.

Specifications:

- The system must provide a " Register " option as a designated registration page.
- Students will be prompted to enter their required personal details, including full name, email, password, department, and current year.
- Validation checks will ensure the email address is in the correct format.
- When a successful submission comes, the system will create a new account.
- An error message will be displayed if any of the provided information is invalid or incomplete, prompting users to correct the entries.

Table 6: Register Tabular form

Condition	Action
Password is < 8 letters or digits	Show an error message.
Students email is not valid	Show a message to the user that the email is not valid.
Password is >= 8 letters or digits	Sign up successfully
Students email is valid	Sign up successfully

Password must include at least:

- One uppercase letter (A-Z).
- One lowercase letter (a-z).
- One digit (0-9).
- One special character (e.g., !, @, #, \$, etc.).

2- Log In

Description: Users, including students, admins, and instructors, can log in to their respective accounts using the login page. Each role has its own login form based to their specific permissions and functionalities.

Specifications:

- The system must present a "Log In" option as a login page.
- Users will be required to enter their registered email address and password.
- After submission, the system will verify the data against the database.
- When a successful authentication, users will be directed to their respective dashboard or landing page based to their role (student dashboard, admin dashboard, instructor dashboard).

3- Forget password

The "Forgot Password" feature is designed to help users securely reset their passwords if they forget them. When a user clicks on the "Forgot Password" link on the login page, they are prompted to enter their registered email address. The system then sends a new password to the user's email.

4- Log Out

Description: Users can log out of their account to end their current session.

Specifications:

- The system shall provide a "Log Out" option accessible within the user interface, typically

located in the header.

- When selecting the "Log Out" option, the system will terminate the user's current session and clear any session-related data stored locally.
- After logging out, users will be redirected to the selection login page.
- The "Log Out" functionality will invalidate the current session token, ensuring that the user cannot access restricted areas or perform actions without re-authentication.

5- Greet User

Description: Display a personalized greeting message, typically showing the name of the logged-in user on top of the page.

Specifications:

- The system must retrieve the name of the logged-in user when successful authentication.
- On the dashboard, a short piece of text will be displayed, incorporating the user's name, such as "Welcome, [Full+ Name]."

6-Contact us

Description: The "Contact Us by Email" feature enables users to send inquiries, feedback, or messages to the platform's support or customer service team through an email-based communication system. Additionally, users can also use the 'Contact Us' feature to request assistance from an administrator for creating an instructor profile."

Specifications:

- User Interface: There must be a designated "Contact Us" section or page accessible.
- The contact will be done by using the email address.

7- Log in selection

Specification:

- User Interface: Present a selection interface with options for roles:
Student, Instructor, Admin.
- Include a "Login" button for initiating the login process.

- Detect the selected role when user interaction.

8- Manage course by admin:

Allow administrators to manage courses, including view, delete, and publish, unpublish via an intuitive and secure web interface. Also display course overviews (number of students and number of students who complete the course).

8.1 view course:

1- User Interface:

- Display course list with search filter.
- Show course details including course name and for which department the course belongs.

8.2 delete course:

1- Confirmation Prompt:

- Display a confirmation prompt before proceeding with course deletion.
- Include a warning message about the irreversible action.

2- Access Control:

- Restrict deletion functionality to admin accounts only.
- Implement authentication checks to ensure authorized access.

8.3-publish, unpublish:

Implement "Publish" and "Unpublish" functionalities for admins to control the visibility of courses to the public.

1-Publish Course:

- Allow admins to select a course and mark it as "Published."
- Update the course status to indicate it is visible to the public.

2-Unpublish Course:

- Enable admins to choose a published course and mark it as "Unpublished."
- Update the course status to indicate it is not visible to the public.

8.4- Display course overview:

Including the number of enrolled students and the number of students who have completed the course.

Specification:

1- Course Overview Section:

- Include a dedicated section on the course page for the overview.
- Display key metrics such as enrollment count and completion count.

2- Enrollment Count:

- Show the total number of students currently enrolled on the course.
- Update enrollments count dynamically based on real-time data.

3- Completion Count:

- Display the number of students who have successfully completed the course.
- Update completion count dynamically based on course completion criteria.

9- Manage student by admin

9.1- View Student

- Specification: Provide administrators with a comprehensive list of all student profiles in the system.
- Details Display: Include student names, IDs, email, department, and year.
- Filtering Options: Allow administrators to filter and search for specific student profiles based on criteria such as name, IDs, and email.

9.2 Add Student

- Specification: Allow administrators to create new student profiles within the system.
- Input Fields: Include fields for student name, email, password, department, and year.
- Validation Checks: Implement validation checks to ensure data accuracy and completeness during profile creation.

9.3 Edit Student

- Specification: Enable administrators to modify existing student profiles as needed.
- Editable Fields: Provide options to edit student names, email, password, year, and department.

9.4 Delete Student

- Description: Provide a secure mechanism for administrators to delete student profiles with confirmation prompts and cascading effect handling.

9.5 Delete Student Profile

- Specification: Allow administrators to delete student profiles when necessary.
- Confirmation Prompt: Implement a confirmation prompt to prevent accidental deletions.
- Cascading Deletion: Handle cascading effects such as removing associated records or data linked to the deleted student profile.

10- Manage instructor by admin

10.1 View Instructor

- Specification: Provide administrators with access to a list of all instructor profiles in the system.
- Details Display: Include instructor names, IDs and joining date.
- Filtering Options: Allow administrators to filter and search for specific instructor profiles based on criteria such as name, ID and joining date.

10.2 Add Instructor Profile

- Specification: Allow administrators to create new instructor profiles within the system.
- Input Fields: Include fields for instructor name, ID, password, joining date and address.
- Validation Checks: Implement validation checks to ensure data accuracy and completeness during profile creation.

10.3 Edit Instructor Profile

- Specification: Enable administrators to modify existing instructor profiles as needed.
- Editable Fields: Provide options to edit instructor names, password, email, and joining date.
- Audit Logging: Log all changes made to instructor profiles, including timestamps and admin IDs, for audit trail purposes.

10.4 Delete Instructor Profile

- Specification: Allow administrators to delete instructor profiles when necessary.
- Confirmation Prompt: Implement a confirmation prompt to prevent accidental deletions.
- Cascading Deletion: Handle cascading effects such as removing associated records or data linked to the deleted instructor profile.

11 - Display number of students

Specification: Add a section in the admin interface showing the total number of registered students.

- Update Frequency: Ensure the count updates periodically or upon admin request to reflect real-time data.
- Simple Interface: Present the student count in a clear, easy-to-read format without excessive

details.

- Access Control: Limit access to this feature to authorized administrators for data security.

12- Display number of instructors

Specification: Add a dedicated section in the admin dashboard to display the total number of instructors in the system.

- Update Frequency: Ensure the count updates in real-time or upon admin request to reflect current data accurately.
- Clear Presentation: Display the instructor count prominently and in a straightforward format for easy understanding.
- Access Control: Limit access to this feature to authorized administrators to maintain data security.

13- manage course by instructor

13.1 View Courses

Specification: Provide instructors with a list of courses they are assigned to teach.

- Details: Display key information such as course title, description, year, and department.
- Filtering: Allow instructors to filter courses based on parameters like course year or department and title.

13.2 Add Course

Specification: Enable instructors to create new courses within the system.

- Input Fields: Include fields for course title, description, image, department, and year.
- Validation: Implement validation checks to ensure accurate data entry.

13.3 Edit Course

Specification: Allow instructors to modify existing course details.

- Editable Fields: Provide options to edit course title, description, department, year, and image.
- Validation: Validate edited information to maintain data integrity.

13.4 Delete Course

Specification: Allow instructors to delete courses if necessary.

- Confirmation: Implement confirmation prompts to prevent accidental deletions.
- Cascade Deletion: Handle cascading effects, such as removing enrolled students, grades, and related data.

13.5 publish/ unpublish course:

The instructor should have the ability to publish and unpublish courses.

14. Manage lesson by instructor

14.1 View Lessons sections

Specification: Provide instructors with a clear overview of all lessons they are responsible for.

14.2 Add Lesson sections:

Specification: Enable instructors to create new lessons and organize them within the course structure.

- Input Fields: Include fields for lesson title.
- Validation: Validate input data to ensure completeness and accuracy.

14.3 Edit Lesson sections:

Specification: Allow instructors to modify existing lesson details.

- Editable Fields: Provide options to edit lesson title.
- Validation: Validate edited information to maintain consistency and data integrity.

14.4 Delete Lesson

Specification: Allow instructors to delete lessons when necessary.

- Confirmation: Implement confirmation prompts to prevent accidental deletions.
- Cascade Deletion: Handle deletion of associated sections, items, and related data.

14.5 Drag-and-Drop Functionality

Specification: Implement drag-and-drop functionality for instructors to rearrange sections and items within lessons.

- User Interface: Design an intuitive and responsive UI for smooth drag-and-drop interactions.
- Feedback: Provide visual feedback (e.g., highlighting, snapping) during drag-and-drop actions for clarity.

15-Manage course by student

15.1 View All Courses

Specification: Allow students to browse and view all available courses on the website.

- Details: Display course titles, descriptions, year, and department.

15.2 Add Courses to List

Specification: Enable students to add courses to their personal list or dashboard for easy access.

- User Interface: Provide an intuitive interface with an "Add to My Courses" button or similar functionality.
- Visibility: Ensure added courses are visible and accessible only to the respective student.

15.3 Track Ongoing Progress

Specification: Allow students to track their progress within enrolled courses.

- Progress Tracking: Display completion status, grades, and progress metrics for each course unit or module.
- Visual Feedback: Provide visual cues or progress bars to indicate completed and pending tasks.

15.4 View Completed Courses

Specification: Provide students with a separate section to view and access completed courses.

- Completion Status: Display completed courses with details such as completion date, grades, and certificates if applicable.

15.5 Filter Courses

Specification: Implement filtering options for students to find courses based on specific criteria.

- Filter Categories: Allow filtering by year, department, title, instructor, or other relevant attributes.
- User-Friendly Filters: Design filters that are easy to use and understand, enhancing the search experience for students.

16-Manage Items

16.1 Add New Item

Specification: Enable users to add new items within a section, such as books, YouTube links, or posts.

- Item Types: Support multiple item types, including books (PDFs or e-books), YouTube links (videos), and posts (text content).
- Input Fields: Include fields for item title, description, content (text or URL), file upload for books, and YouTube video embed code.

16.2 Book Addition

Specification: Allow users to add new books or documents as items within a section.

- File Upload: Provide an option to upload book files in supported formats (e.g., PDF, EPUB).
- Metadata: Include fields for book title, author, publication date, and description for easy identification and organization.

16.3 YouTube Link

Specification: Enable users to add YouTube video links as items within a section.

- Link Input: Provide a field to enter the YouTube video URL or embed code.
- Thumbnail Display: Automatically fetch and display video thumbnails for visual identification.

16.4 Post Creation

Specification: Allow users to create new posts or articles using a WYSIWYG (What You See Is What You Get) editor.

- Content Editor: Integrate a WYSIWYG editor with formatting options (e.g., text styles, lists, images, hyperlinks) for content creation.
- Rich Media Support: Allow embedding of images, videos, or other multimedia content within posts for enhanced engagement.

16.5 Item Editing

Specification: Enable users to edit existing items within a section.

- Edit Options: Provide options to edit item titles, descriptions, content, and associated metadata (e.g., book details, video information).
- Version Control: Implement version control or revision history for item edits to track changes.

16.6 Item Deletion

Specification: Allow users to delete items from a section when necessary.

- Confirmation Prompt: Implement a confirmation prompt to prevent accidental deletions.
- Cascade Deletion: Handle deletion of associated data (e.g., book files, YouTube links) and update section content accordingly.

17-Manage Lesson by Student

17.1 Mark Lesson Progress

Specification: Enable students to mark their progress by checking checkboxes or indicators as they complete lessons.

- Visual Feedback: Provide visual cues such as progress bars or completion percentages to track lesson completion.
- Save Progress: Ensure student progress is saved and accessible across sessions for ongoing tracking.

17.2 Download Associated Files

Specification: Allow students to download files associated with lessons to support their learning.

- File Types: Support various file formats such as PDFs, documents, presentations, and multimedia content.
- Download Options: Provide download links or buttons within lesson modules for easy access to associated files.

17.3 Lesson Navigation

Specification: Design intuitive navigation within lessons for seamless progression.

- Lesson Structure: Organize lessons into sections or modules with clear titles and descriptions.

18-Profile management:

18.1 User Role Differentiation

Specification: Clearly define and differentiate user roles such as admin, instructor, and student within the system.

- Role-Based Access: Implement role-based access control to ensure users can only edit their own profiles based on their assigned role.

18.2 Profile Editing

Specification: Provide a user-friendly interface for users to edit their profiles.

- Editable Fields: Include fields such as name, contact information, bio, profile picture, and other relevant details based on user roles.
- Validation: Validate user input to maintain data accuracy and completeness.

18.3 Admin Profile Editing

Specification: Allow admins to edit their profiles with administrative information.

- Admin-specific Fields: Include fields for name and password.

18.4 Instructor Profile Editing

Specification: Enable instructors to update their profiles with teaching-related information.

- Instructor-specific Fields: Include fields name and password.

18.4 Student Profile Editing

Specification: Allow students to modify their profiles with academic and personal information.

- Student-specific Fields: Include fields for student name, department, password, and year.

19- Manage Quiz by Instructor

19.1 View Quiz List

Specification: Provide instructors with a list of all quizzes they have created.

- Details: Display quiz titles, descriptions, creation dates, and relevant course information.

19.2 View Quiz Results

Specification: Enable instructors to access and view quiz results for each student.

- Individual Results: Display student names, scores, attempt dates, and detailed question-level performance.
- Aggregate Data: Provide summary statistics such as average score, highest score, and grade distribution.

19.3 Add Quiz

Specification: Allow instructors to create new quizzes within the system.

- Quiz Details: Include fields for quiz title, validity date, duration, and for which section of course this quiz is related.
- Question Types: multiple-choice.

19.4 Edit Quiz

Specification: Enable instructors to modify existing quiz details and questions.

- Editable Fields: Include fields for quiz title, validity date, duration, and for which section of course, this quiz is related.
- Question Editing: Allow instructors to add, remove, or edit individual quiz questions and answers.

19.5 Delete Quiz

Specification: Allow instructors to delete quizzes when necessary.

- Confirmation: Implement confirmation prompts to prevent accidental deletions.
- Cascade Deletion: Handle deletion of associated quiz questions, student attempts, and related data.

20- Attempt Quiz

Description: Enable students to attempt specific quizzes, view grades, and have the option to reattempt the quiz.

Specifications:

- The system must provide a dedicated page allowing students to access and attempt quizzes associated with a course.
- Students will be able to select a specific quiz to attempt from the available list within the course page.
- When completing the quiz, the system will automatically grade the quiz and display the

result to the student.

- The graded result, including the score or percentage, will be shown immediately after quiz submission.

21- Display term of use and policy

The "Terms of Use and Privacy Policy" page is designed to provide users with comprehensive information about the legal terms and conditions for using the website, as well as the privacy practices regarding the collection, use, and protection of user data. The page must be easily accessible via a clearly labeled link in the footer of every page on the website, ensuring users can find it effortlessly.

22- chat with bot

The "Chat with Bot" feature is designed to allow instructors and students to engage with an AI bot to enhance the quality of learning and teaching methods. This feature aims to provide immediate assistance, answer questions, and offer suggestions for improving educational practices.

Key features include:

- Accessibility and User Interaction: Both instructors and students can access the chat bot through a designated interface within the educational platform.
- The chat bot will be available 24/7 to provide support and answer queries related to course content, teaching methods, and learning strategies.
- Functionality and Personalization: The bot will be equipped with natural language processing capabilities to understand and respond to a wide range of questions.
- It will offer personalized responses based on the user's role (instructor or student) and previous interactions.

23- suggest material for students

The "Suggest Material for Students" feature is designed to enhance the learning experience by providing personalized study recommendations after each quiz attempt. This feature aims to help students improve their understanding and retention of information by offering targeted resources based on their quiz performance. And after student answering questions wrong the AI will assist you addressing the problem and solve it.

3.7 Non-Functional Requirements

Table 7: Non-Functional Requirements

Requirement	Description
Performance	The system must be flexible and must be fast at executing user functions.
Usability	The system must be easy to use, effective and efficient and must give the user a smooth experience to satisfy him.
Reliability	The system should be reliable, which means it should always be available 24/7 and provide information.
Responsiveness	ability of a website to adapt its layout and content to different screen sizes and devices.
Security	The system must keep the user's information safe and private, so it must be highly secured.

3.8 Summary

The outlined project proposes a comprehensive university platform featuring registration, course management, user interaction, and AI support. It utilizes web development tools like HTML, CSS, JavaScript, PHP (Laravel), and MySQL. The feasibility study covers technical, operational, and economic aspects, demonstrating a positive ROI of 5% and a payback period within three years. Targeting university students, the functional requirements encompass user authentication, course management, profile customization, and admin functionalities. Non-functional requirements emphasize performance, usability, reliability, responsiveness, and stringent security measures to ensure a seamless and secure user experience on the platform.

4. Chapter Four: System Design

4.1 Introduction

This chapter includes many important figures that describe our application process: it will include a context diagram, data flow diagram (DFD), entity relation diagram (ERD), use cases, diagrams, sequences diagrams, class diagrams.

4.2 Context Diagram

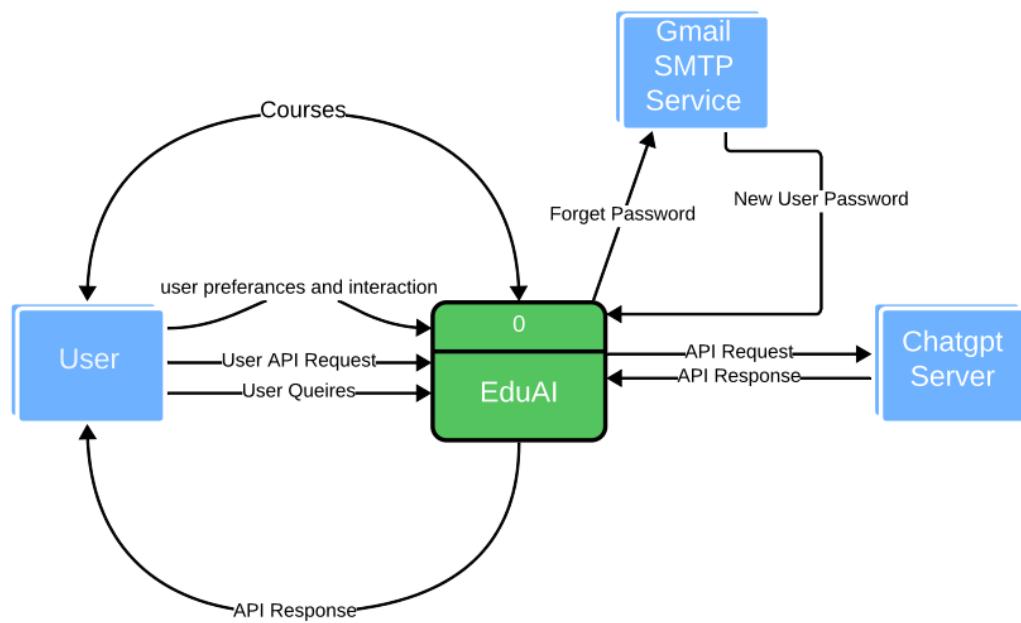


Figure 9: context diagram

4.3 Entity Relationship Diagram (ERD)

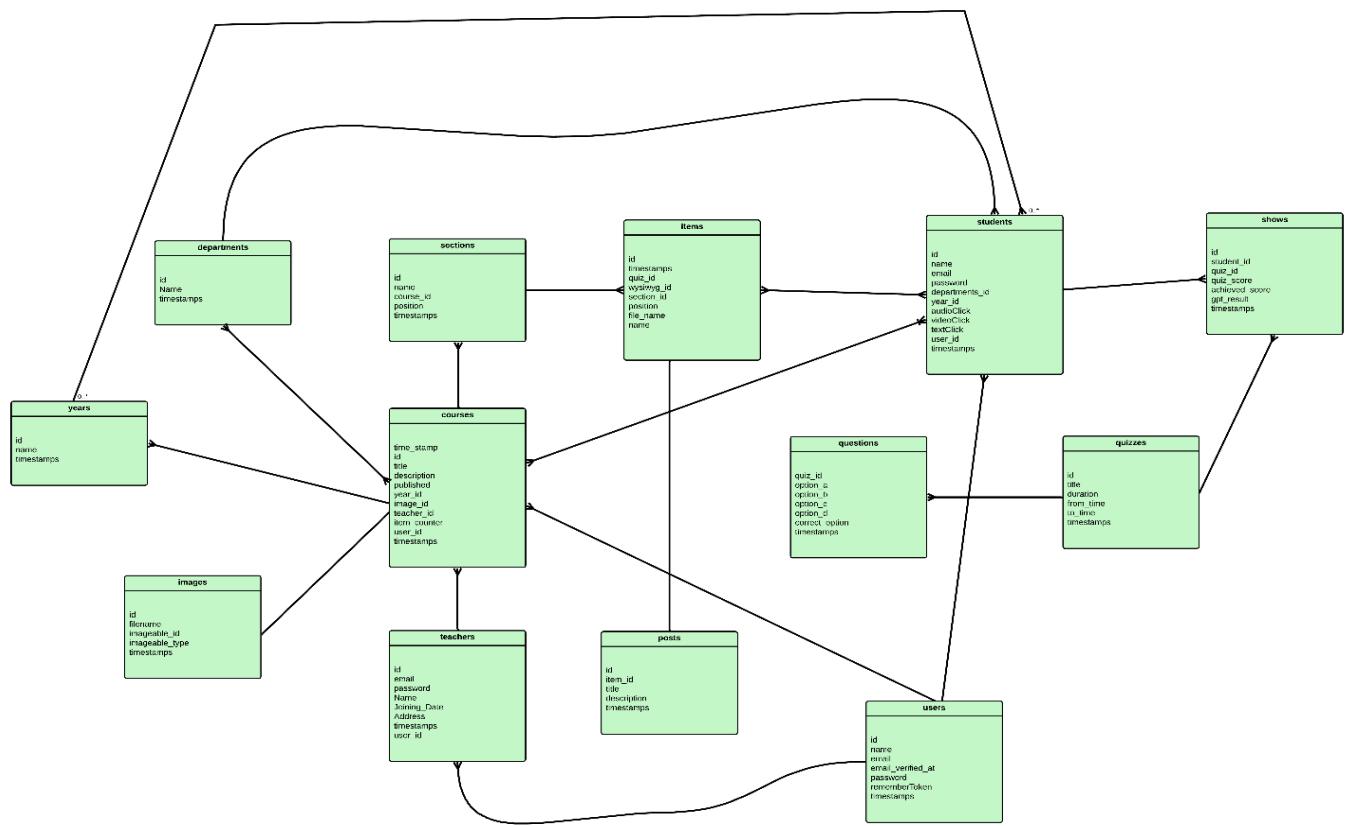


Figure 10: ERD

4.4 UML Use Case Diagram

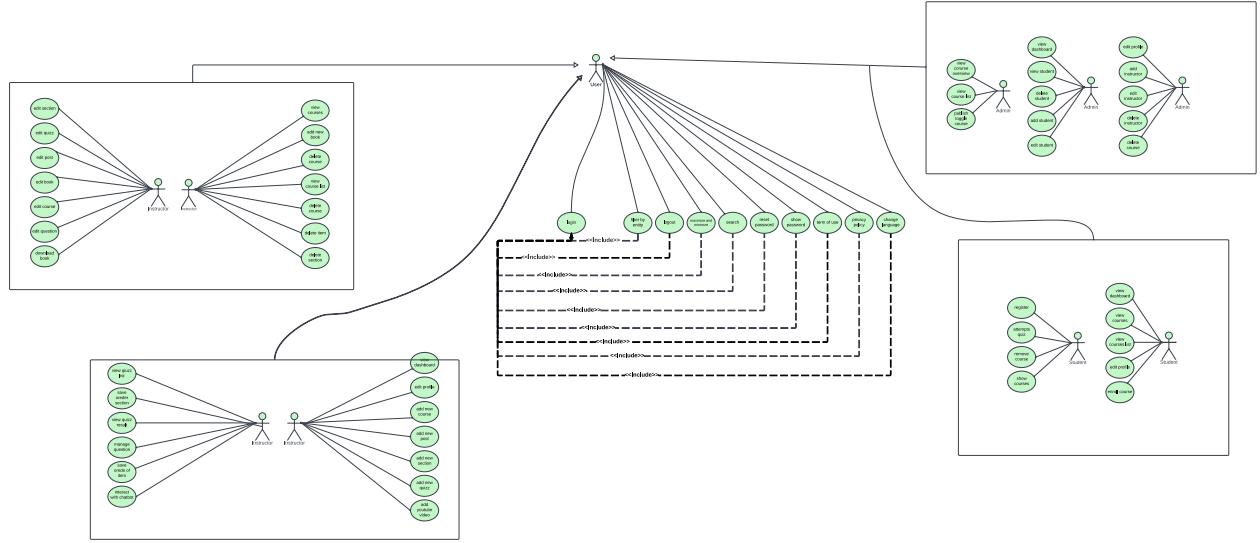


Figure 11: UML use case diagram: EduAI System

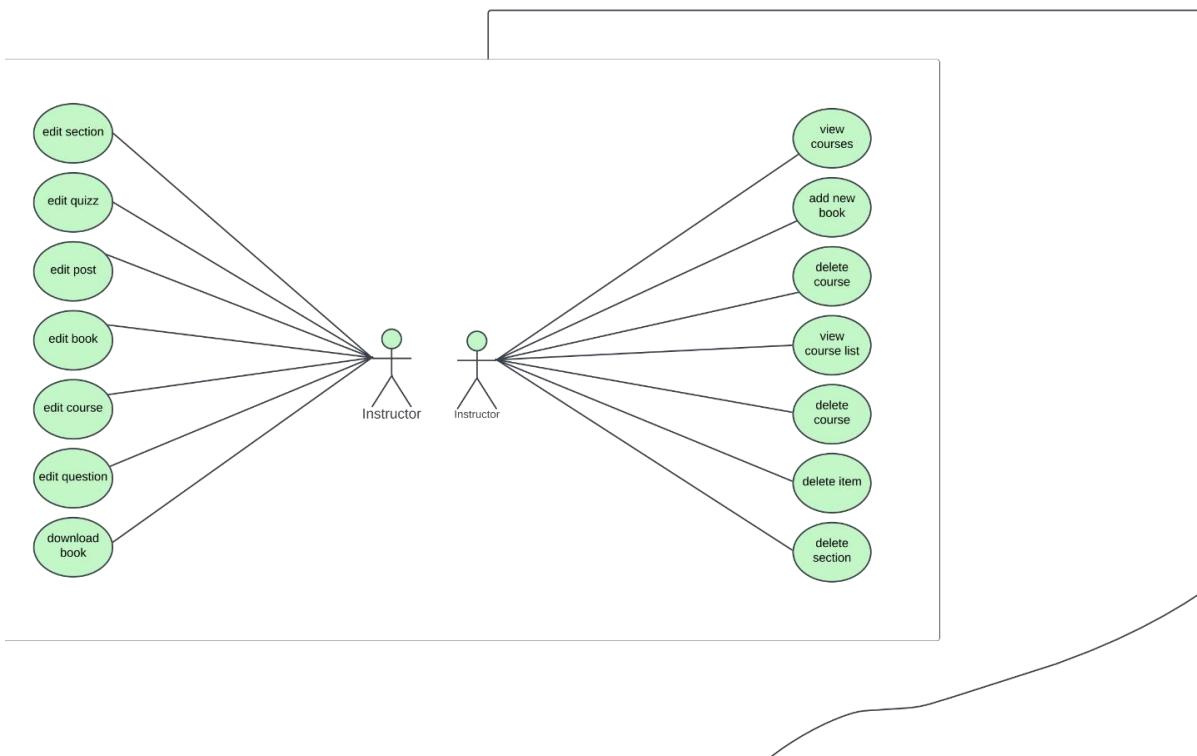


Figure 12: UML use case diagram: 1

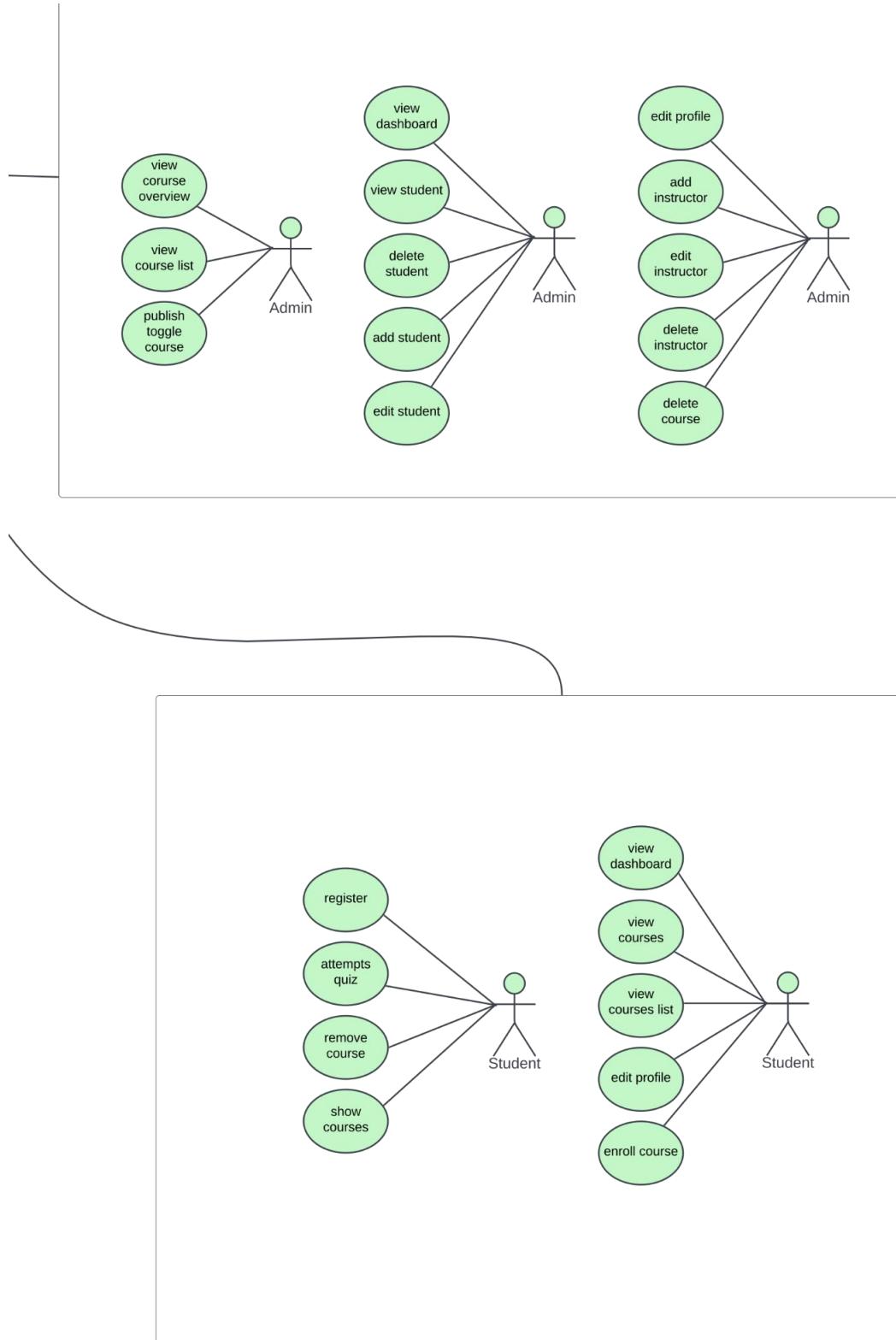


Figure 13: UML use case diagram: 2

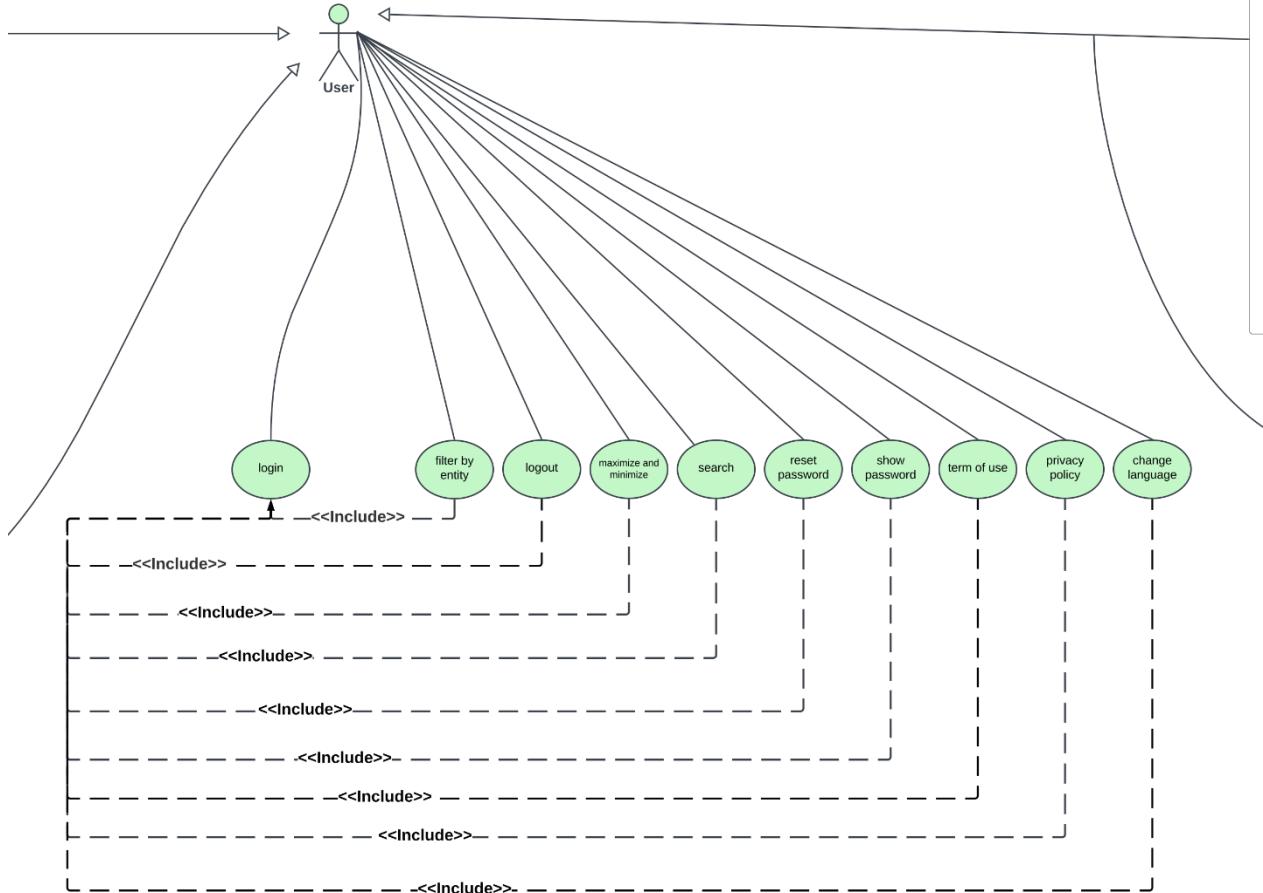


Figure 14: UML use case diagram: 3



Figure 15: UML use case diagram: 4

4.5 UML Sequence Diagram

4.5.1 student sequence

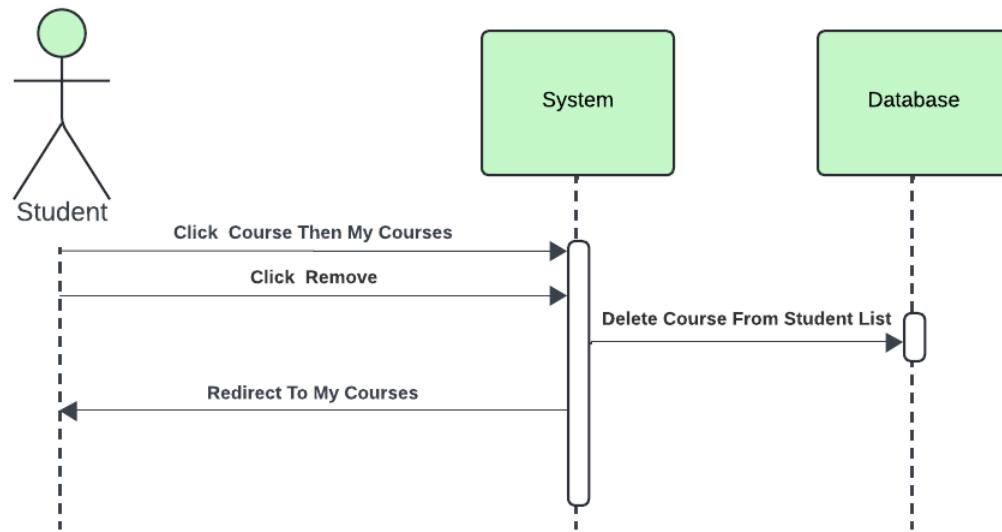


Figure 16:Sequence Diagram: Student Delete Course

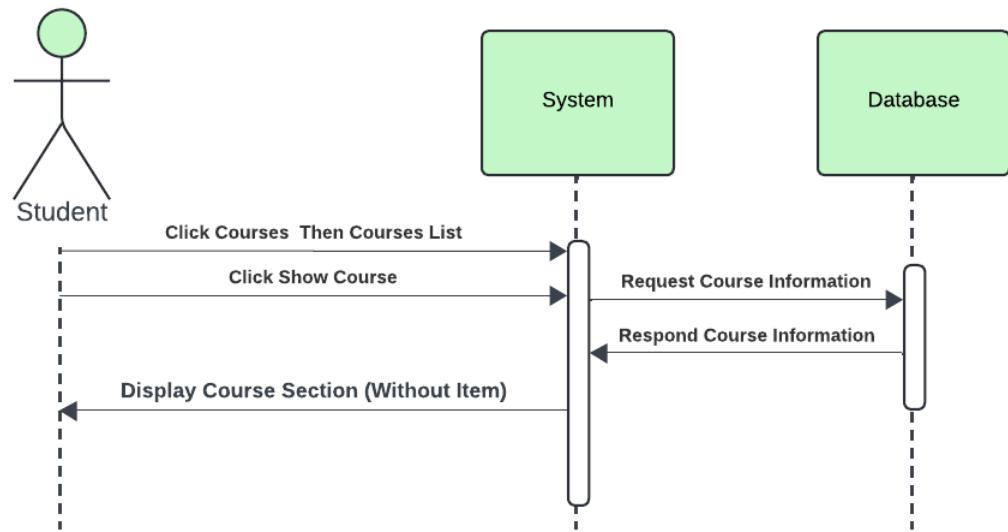


Figure 17: Sequence Diagram: Sequence Diagram: Student Display Course

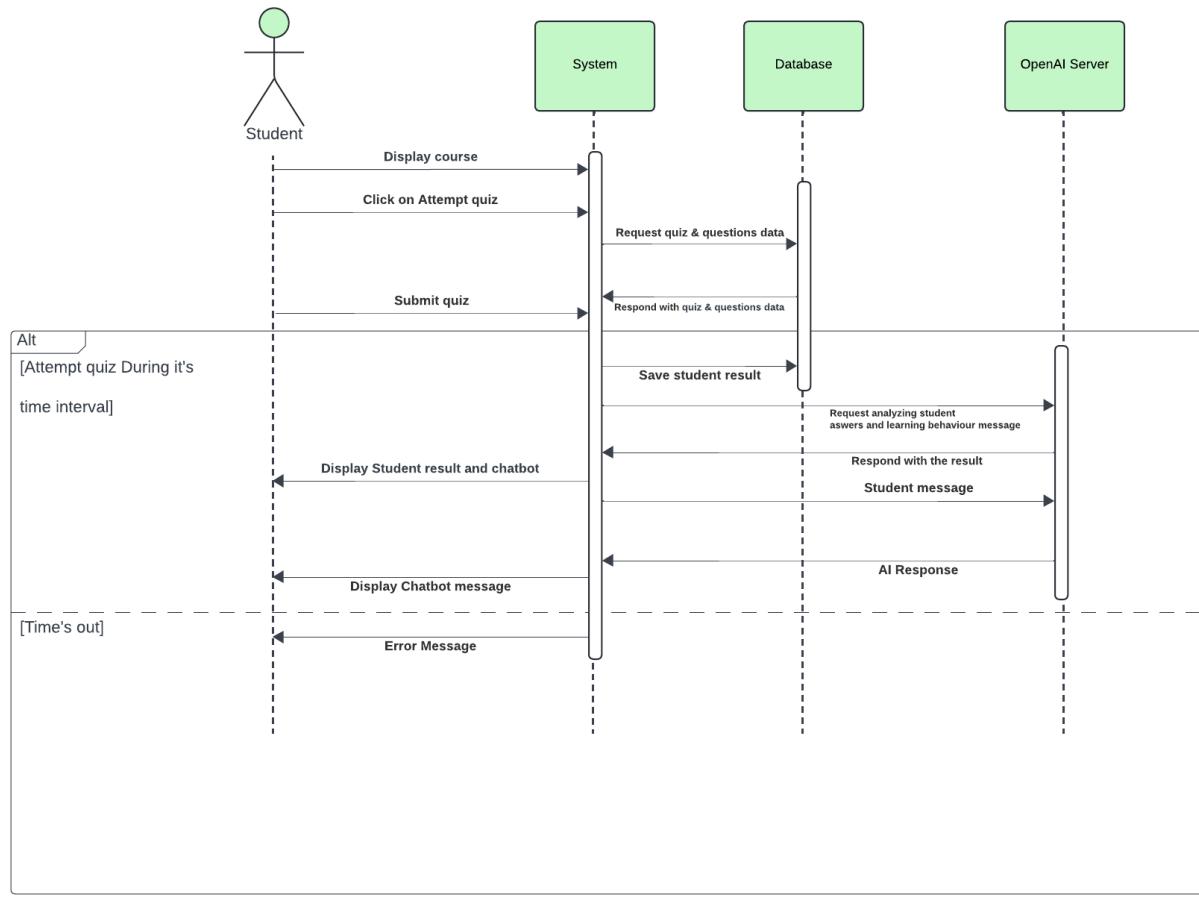


Figure 18: Sequence Diagram: Student Attempt Quiz

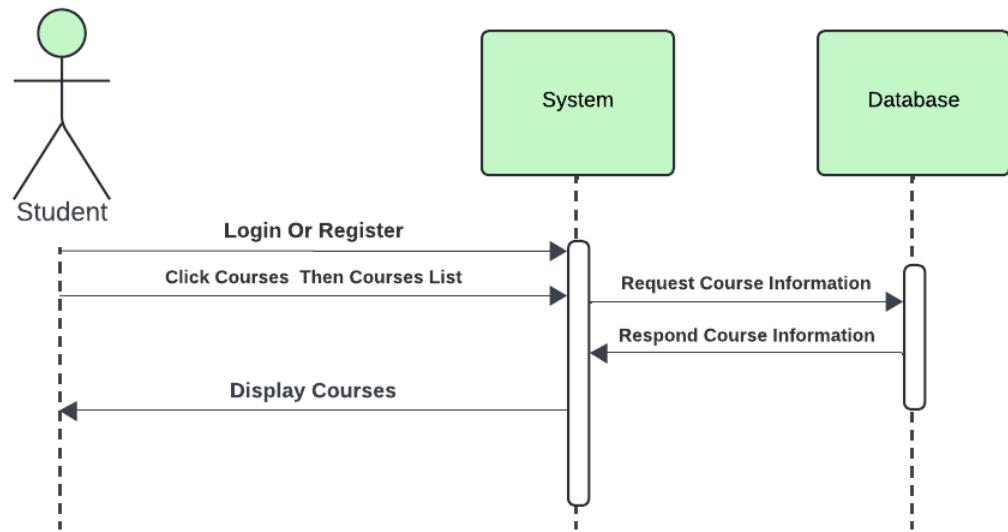


Figure 19: Sequence Diagram: Sequence Diagram: Student Display Courses List

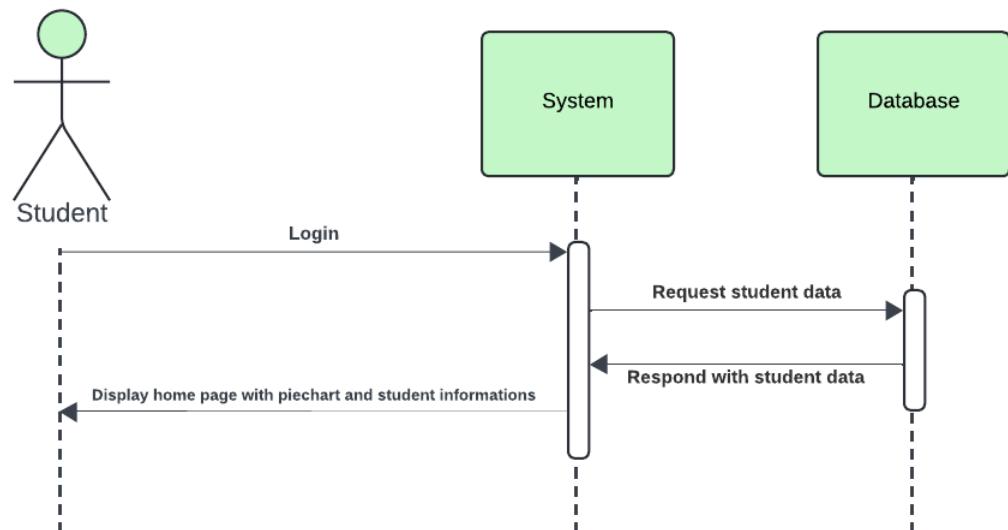


Figure 20: Sequence Diagram: Student Dashboard

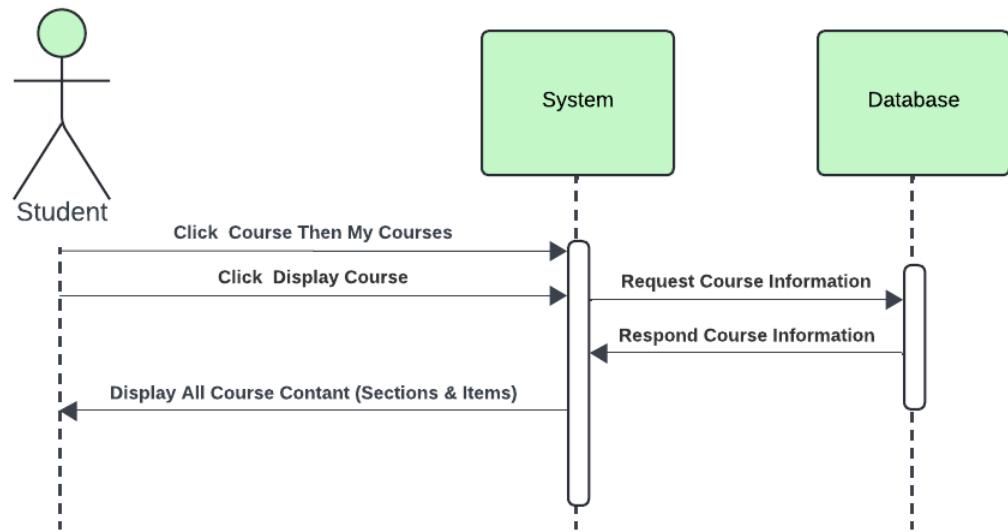


Figure 21: Sequence Diagram: Student Display Courses Content

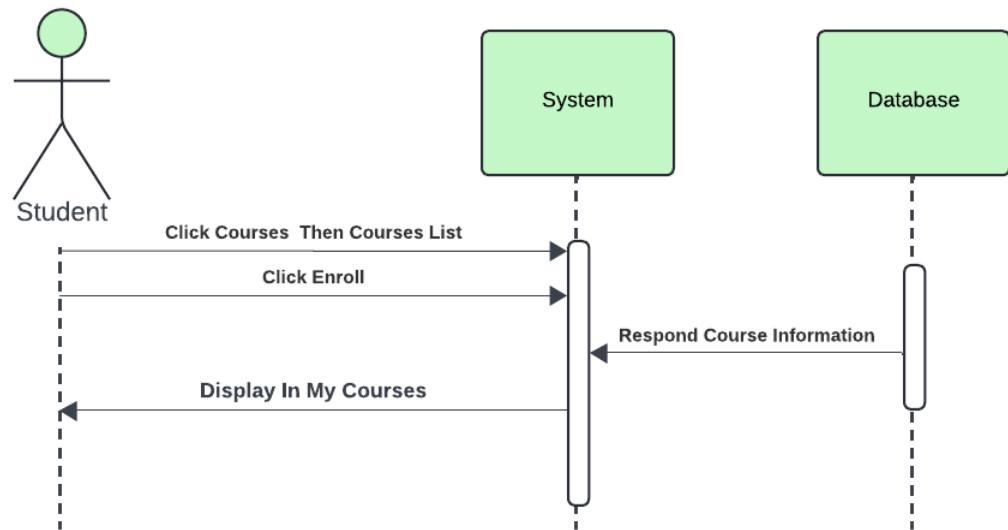


Figure 22: Sequence Diagram: Student Enroll Course

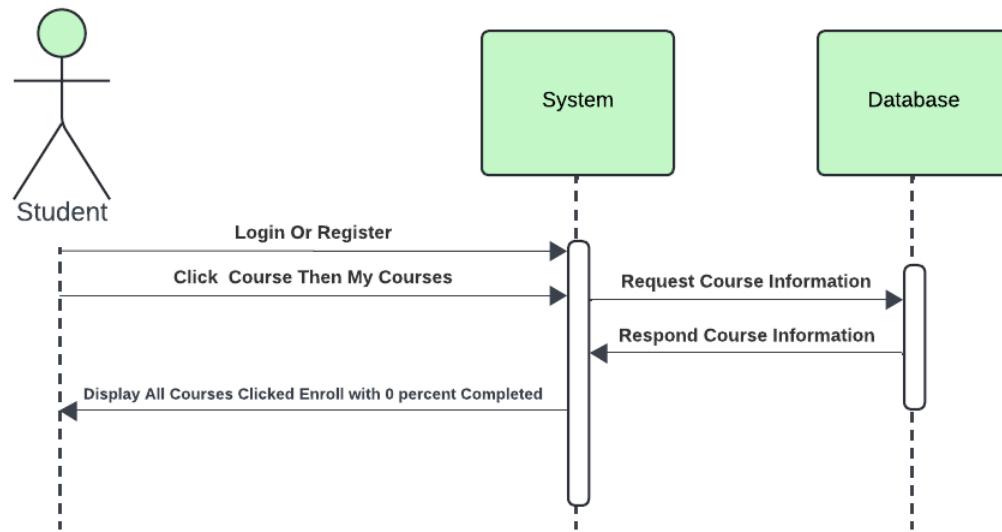


Figure 23: Sequence Diagram: Student Display My Courses

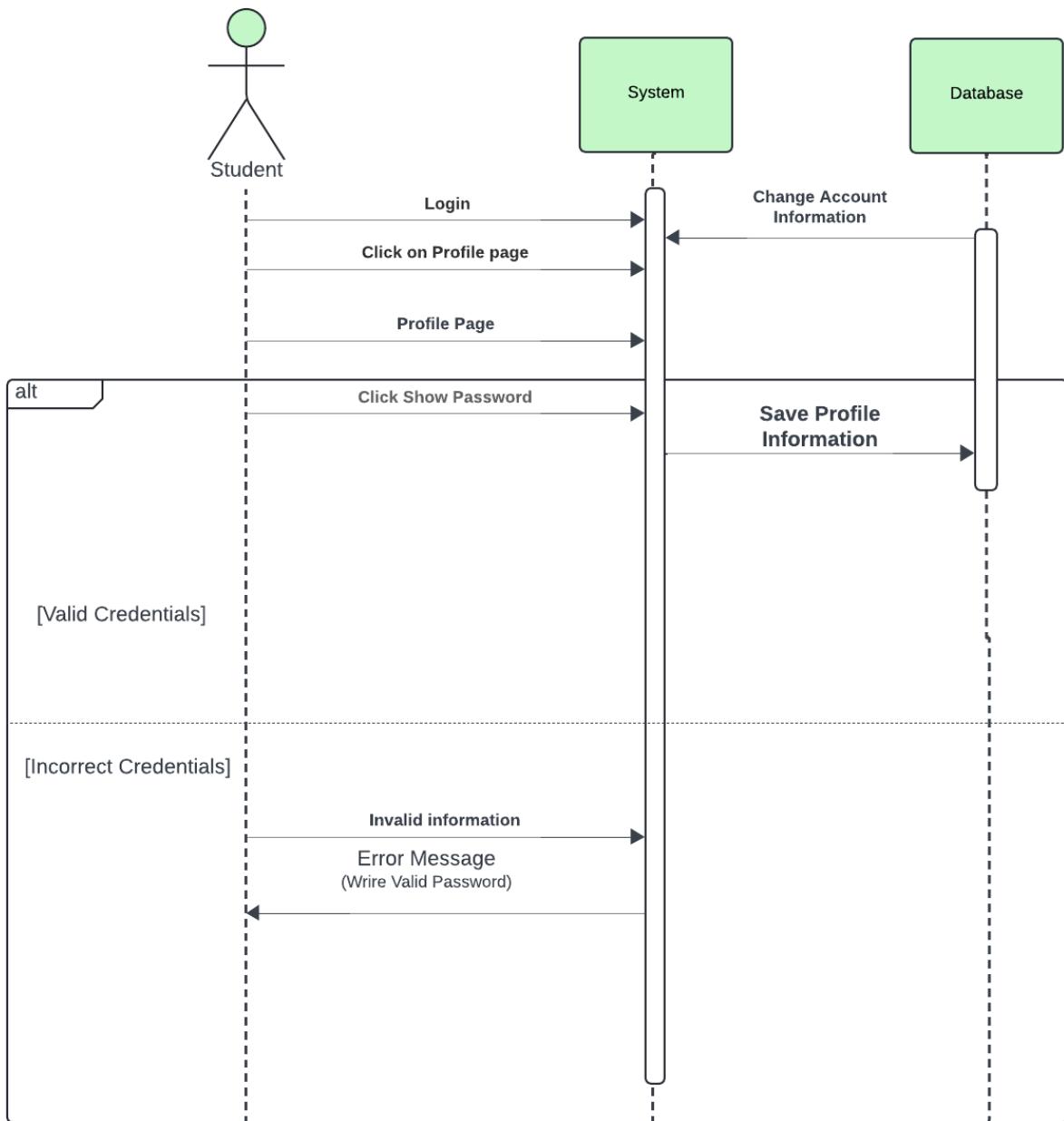


Figure 24::Sequence Diagram: Student Edit Profile

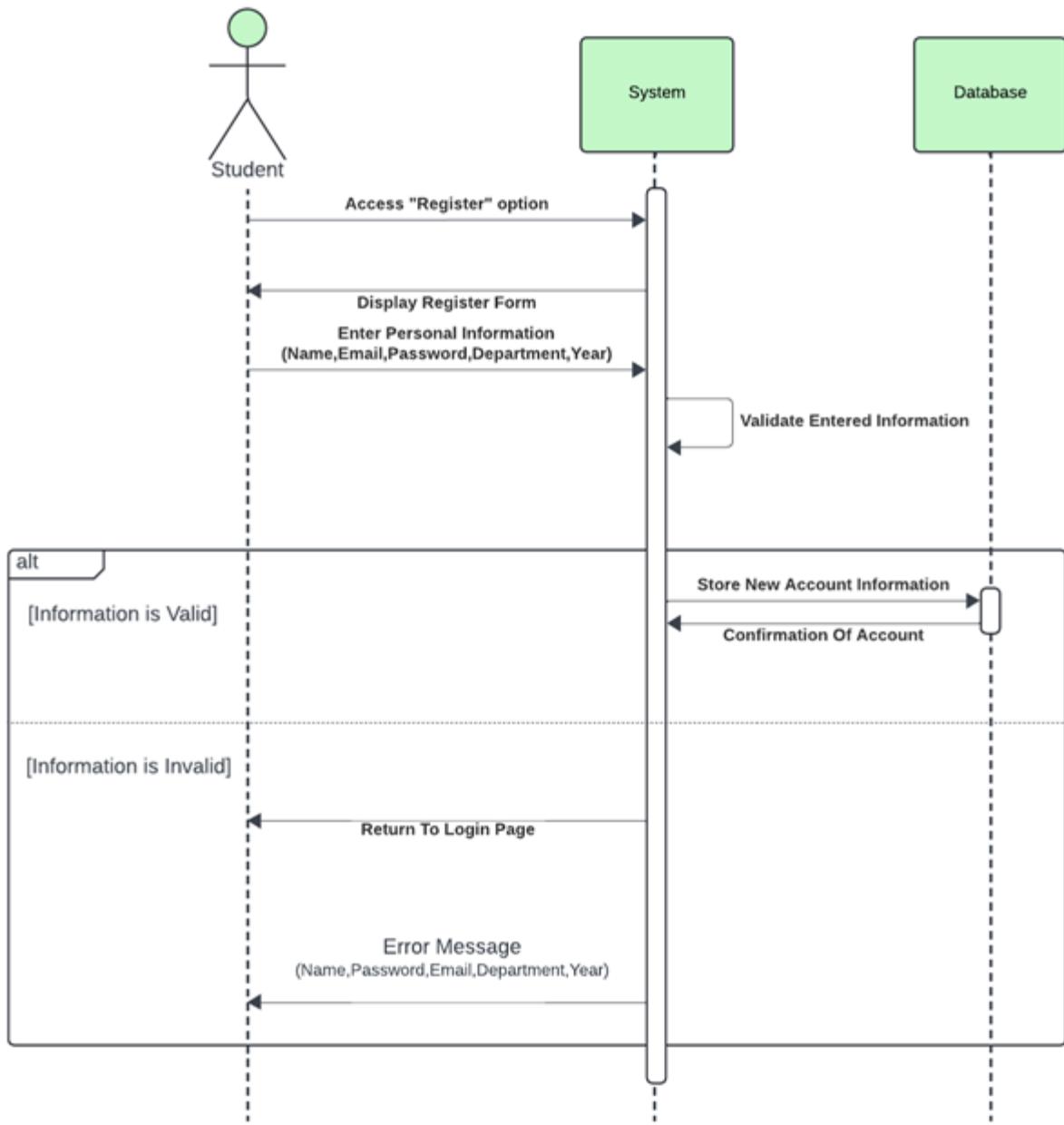


Figure 25: Sequence Diagram: Student Register

4.5.1 instructor sequence

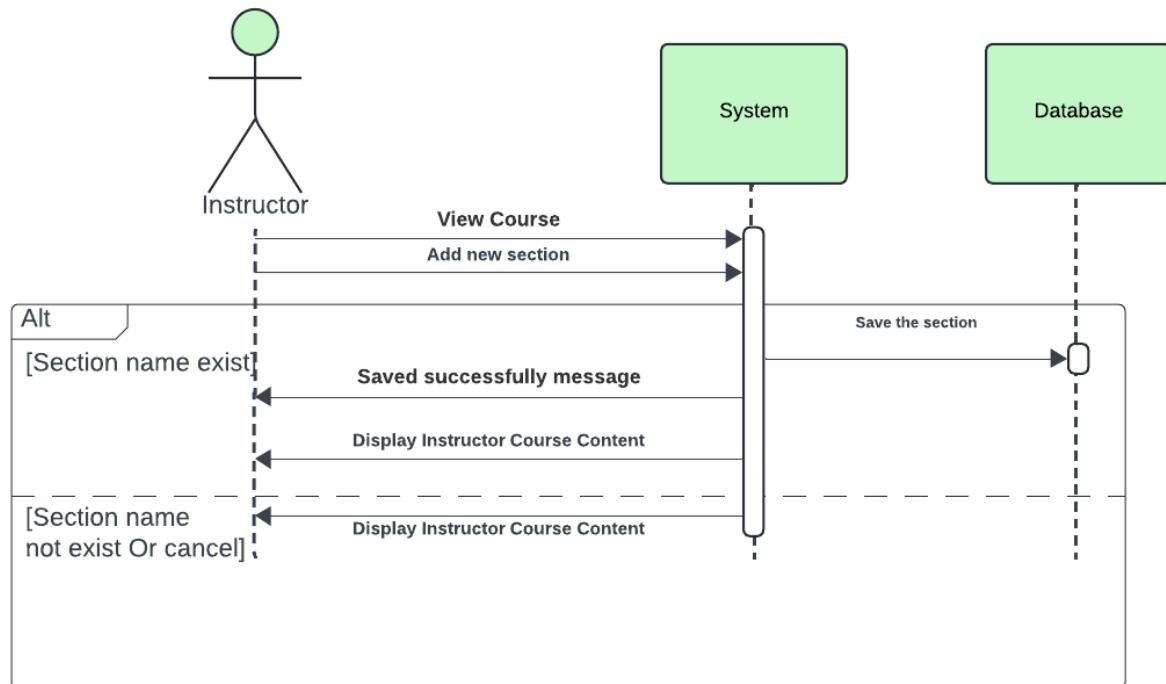


Figure 26: Sequence Diagram: Instructor Add New Section

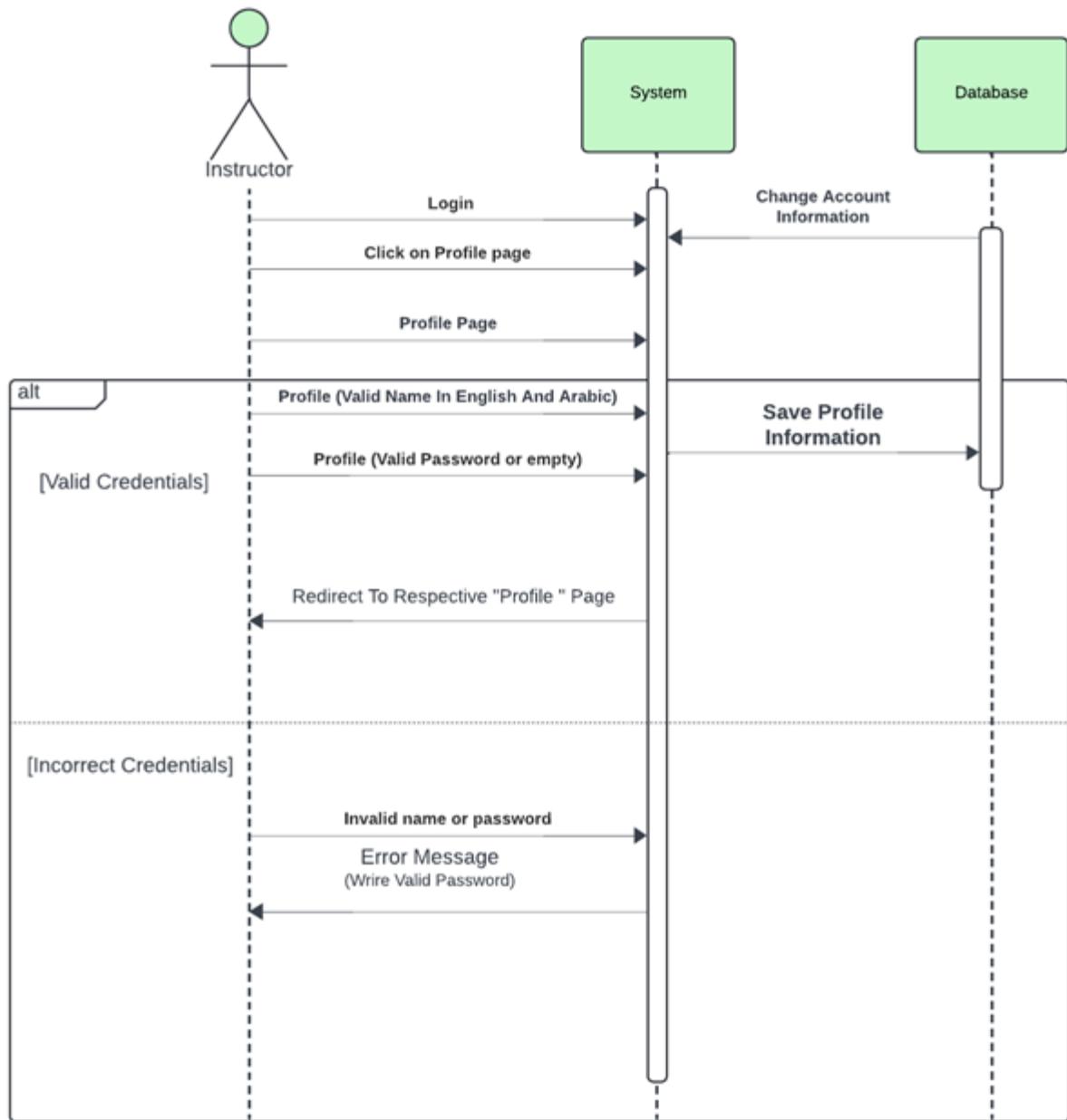


Figure 27::Sequence Diagram: Instructor Edit Profile

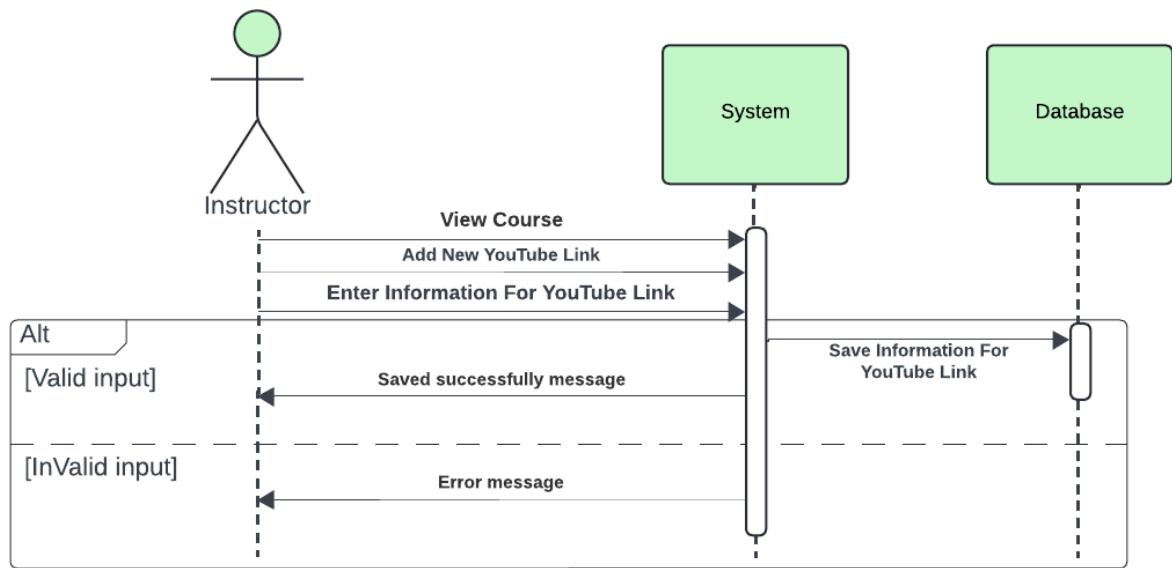


Figure 28: Sequence Diagram: Instructor Add New YouTube Link

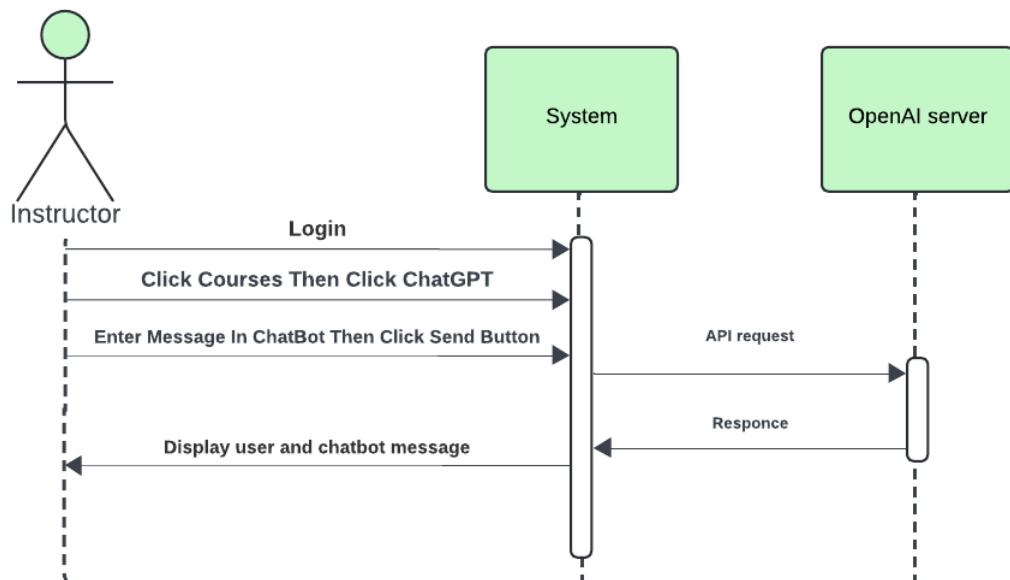


Figure 29: Sequence Diagram: Instructor Chat with GPT

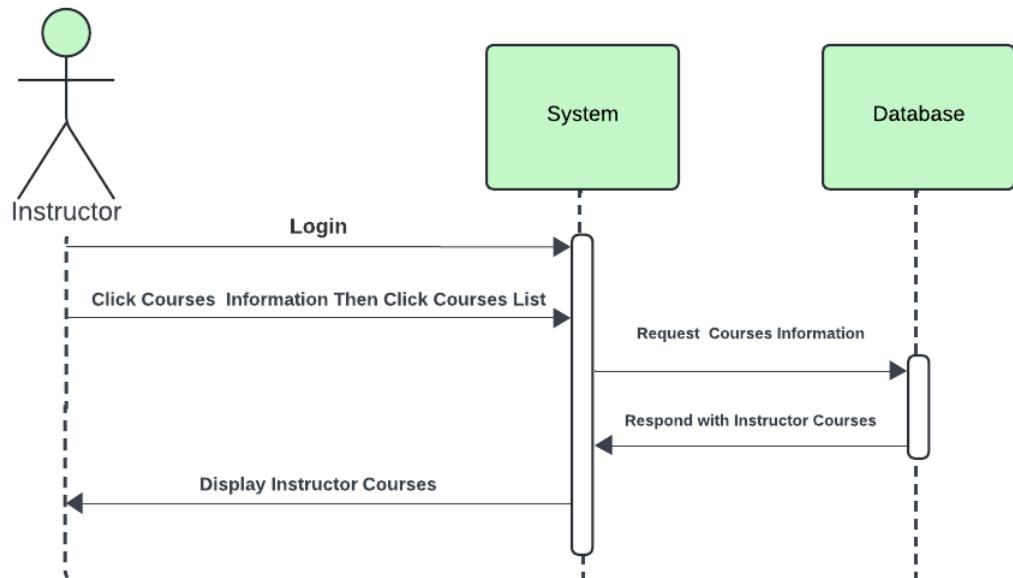


Figure 30: Sequence Diagram: Instructor Display Courses

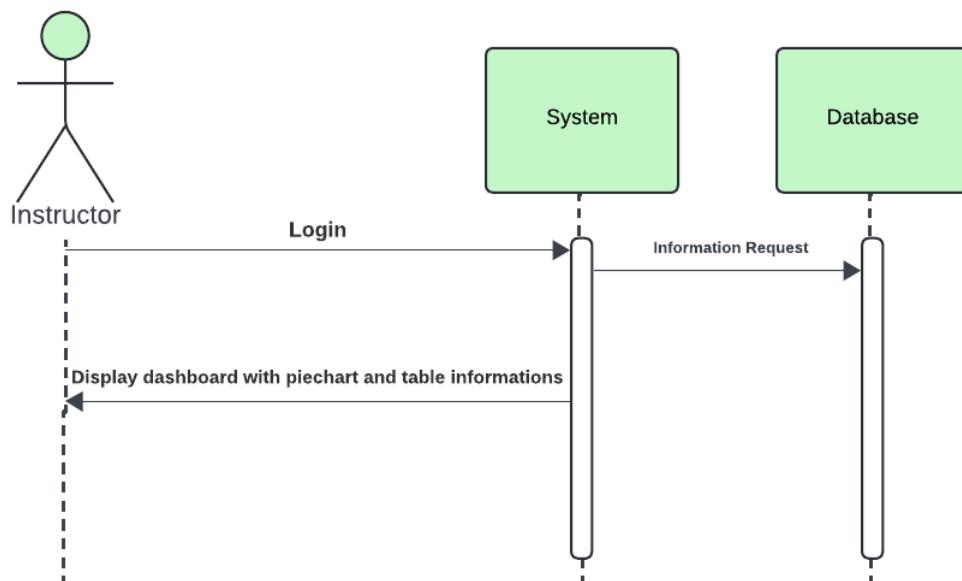


Figure 31: Sequence Diagram: Instructor Dashboard

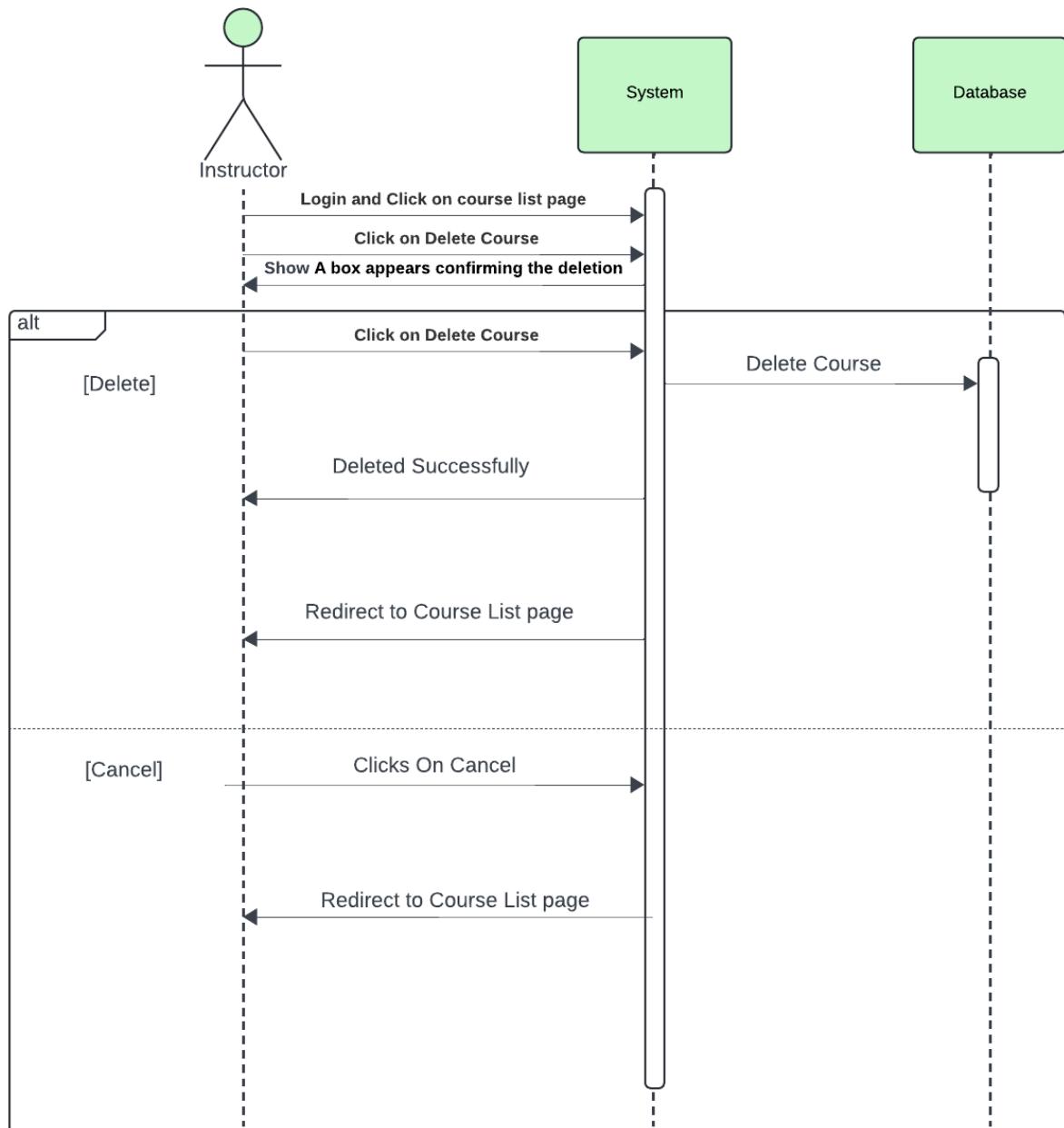


Figure 32: Sequence Diagram: Instructor Delete Course

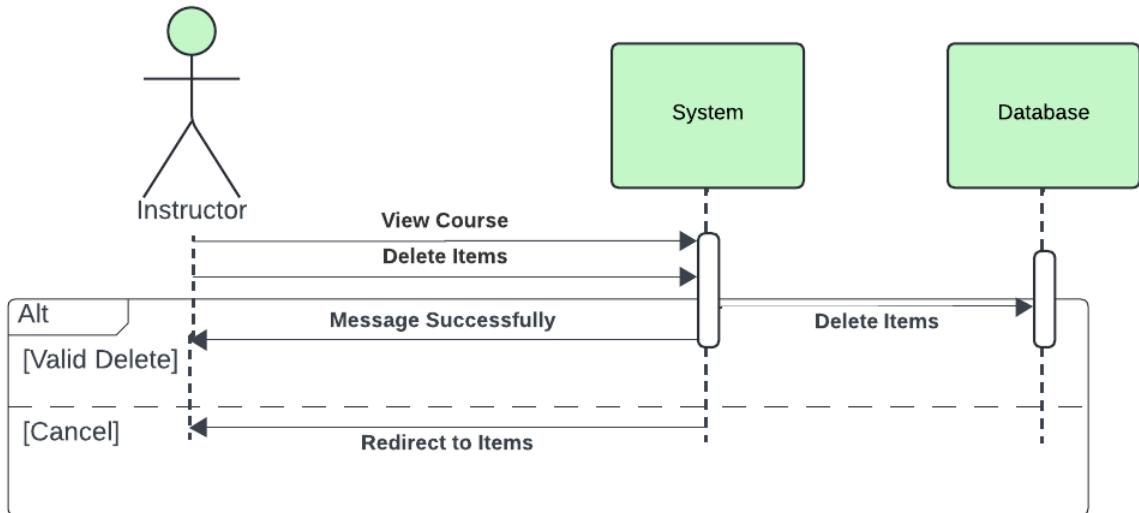


Figure 33: Sequence Diagram: Instructor Delete Item

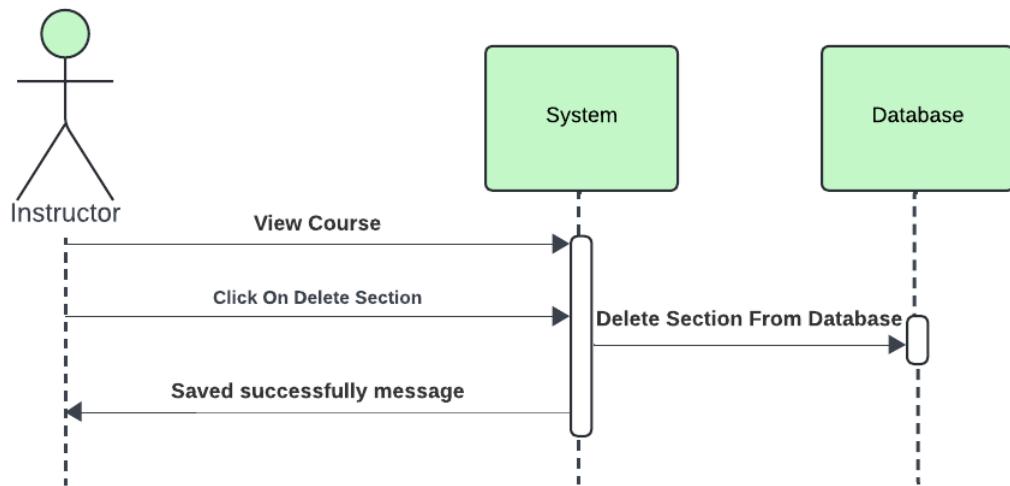


Figure 34: Sequence Diagram: Instructor Delete Section

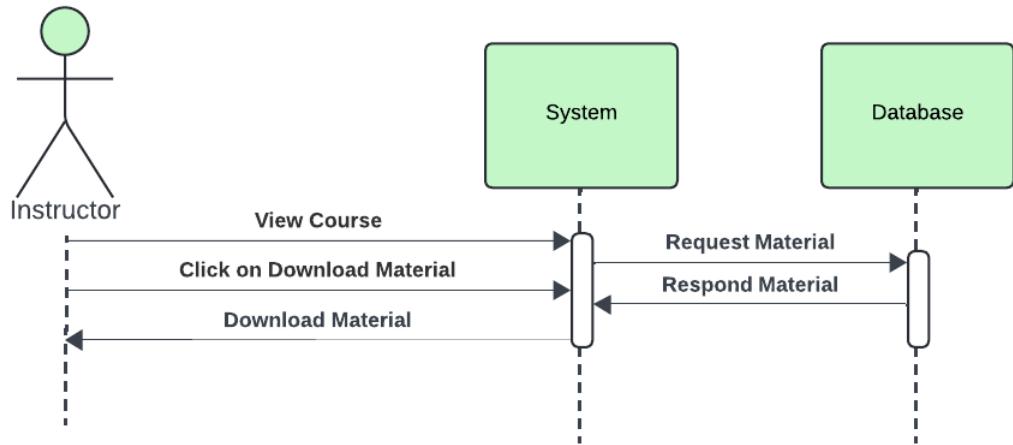


Figure 35: Sequence Diagram: Instructor Download Material

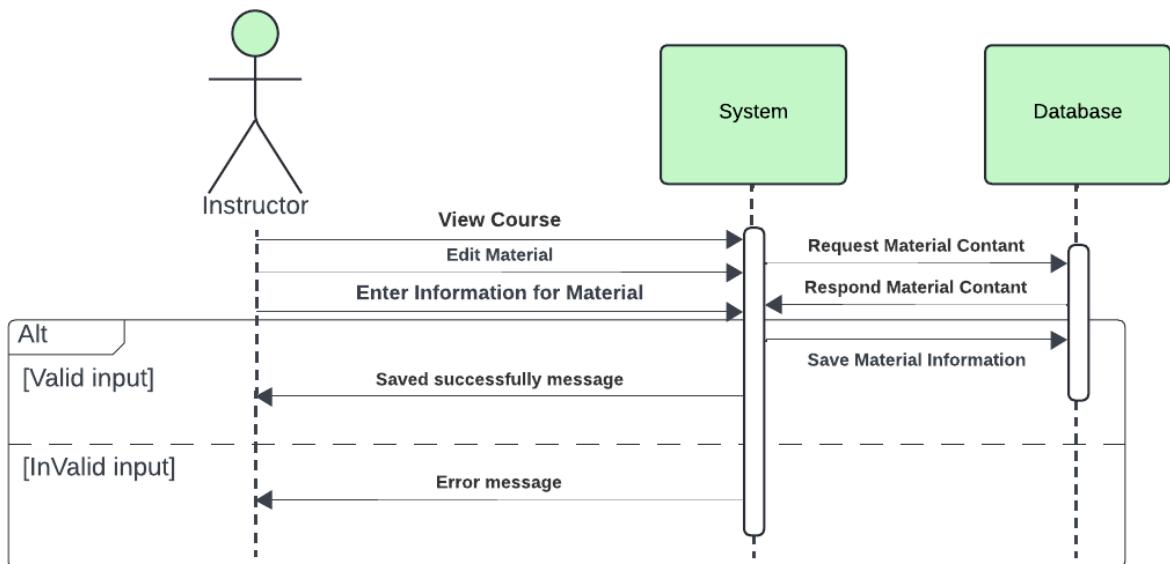


Figure 36: Sequence Diagram: Instructor Edit Material

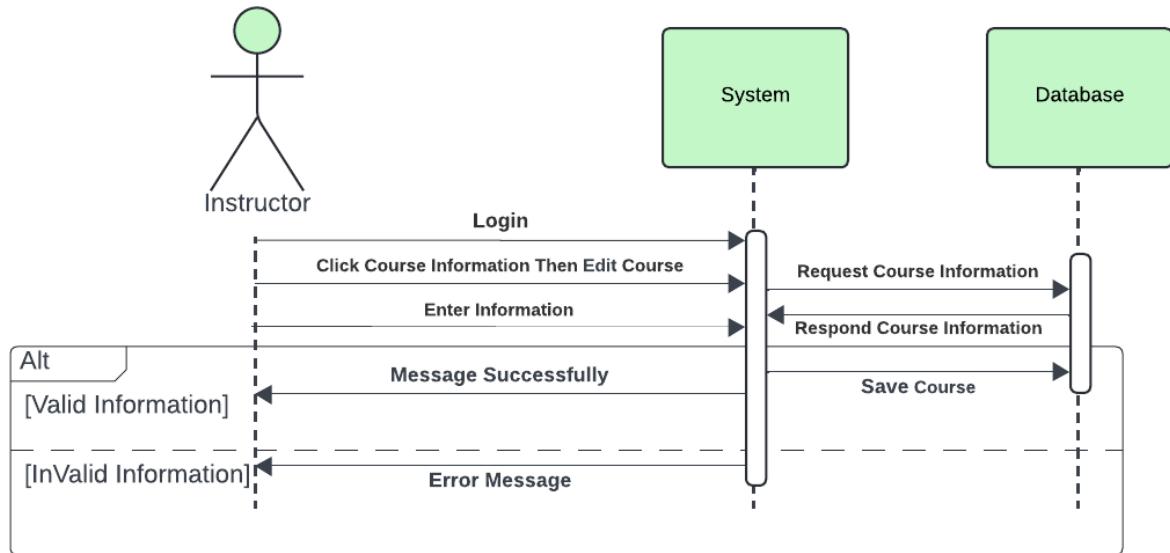


Figure 37: Sequence Diagram: Instructor Edit Course

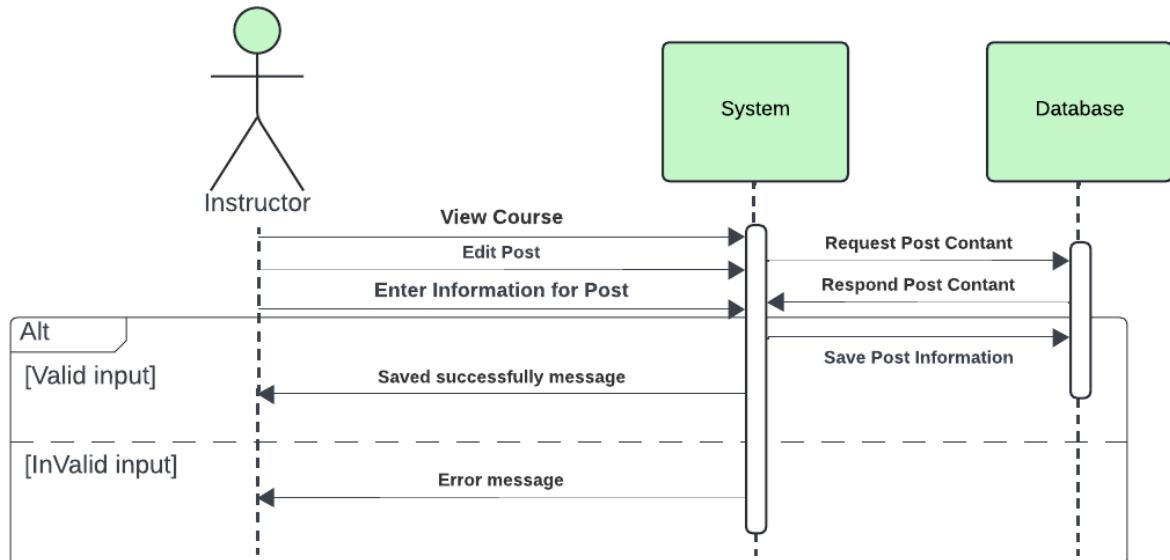


Figure 38: Sequence Diagram: Instructor Edit Post

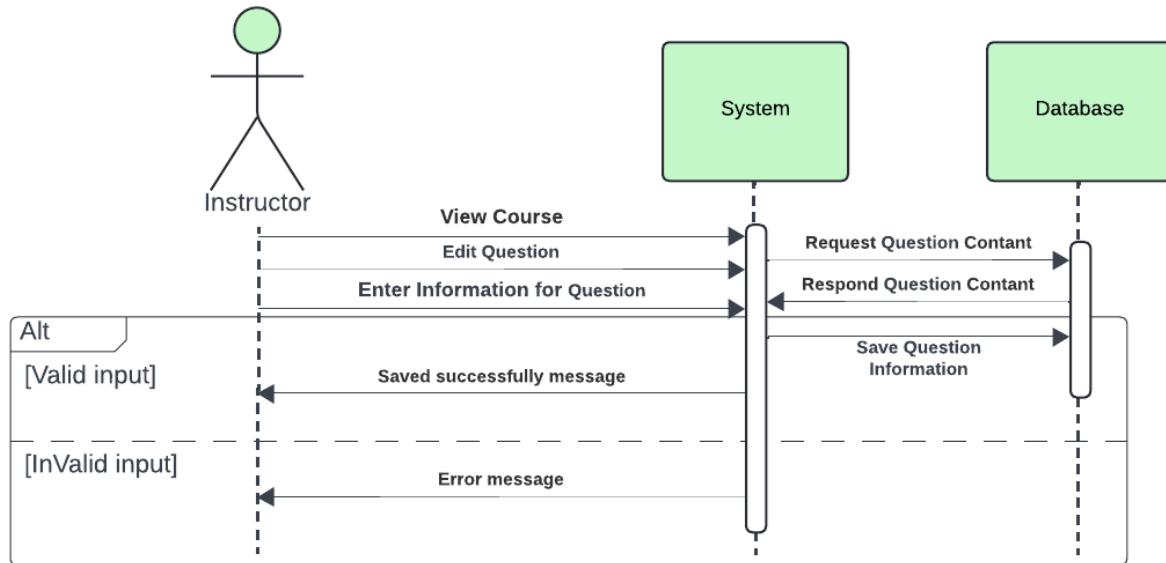


Figure 39: Sequence Diagram: Instructor Edit Question

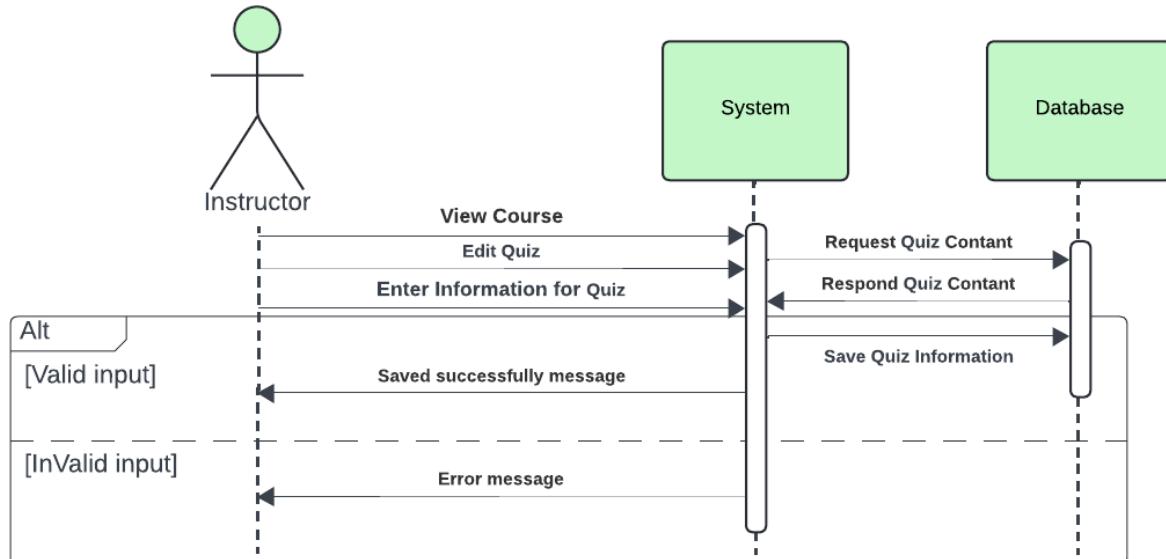


Figure 40: Sequence Diagram: Instructor Edit Quiz

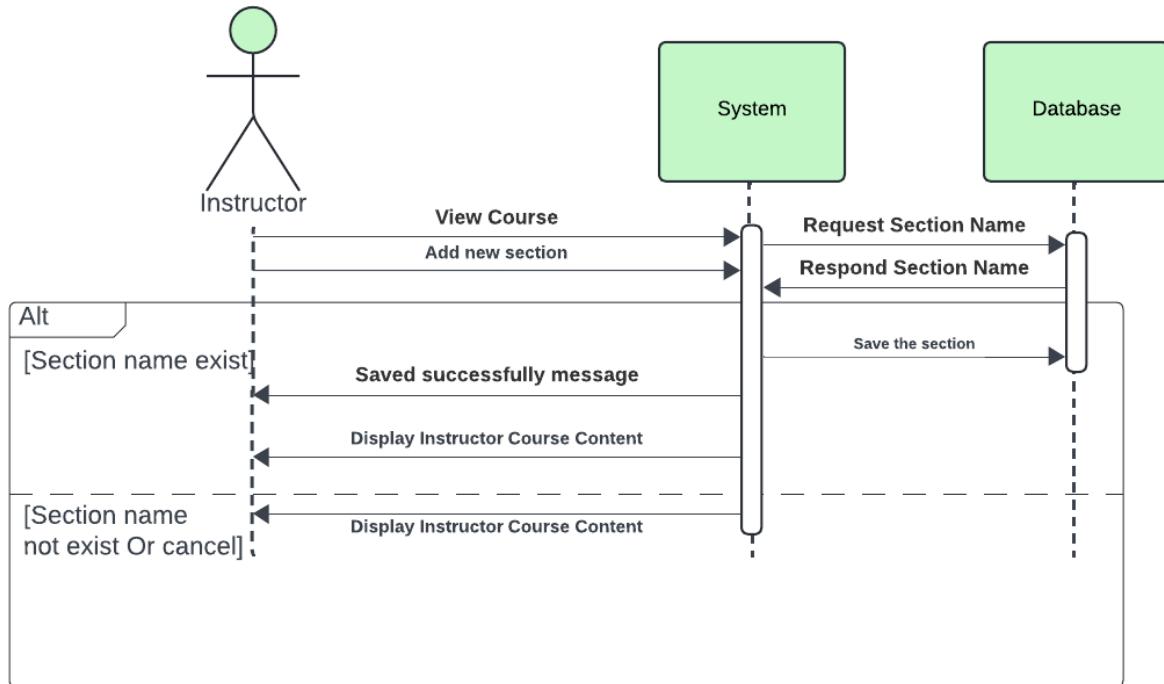


Figure 41: Sequence Diagram: Instructor Add New Section

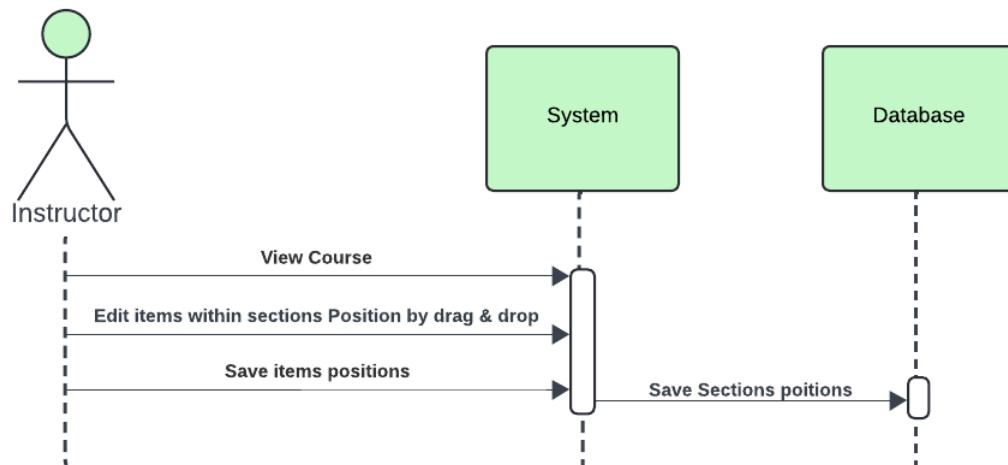


Figure 42: Sequence Diagram: Instructor Edit Item Position

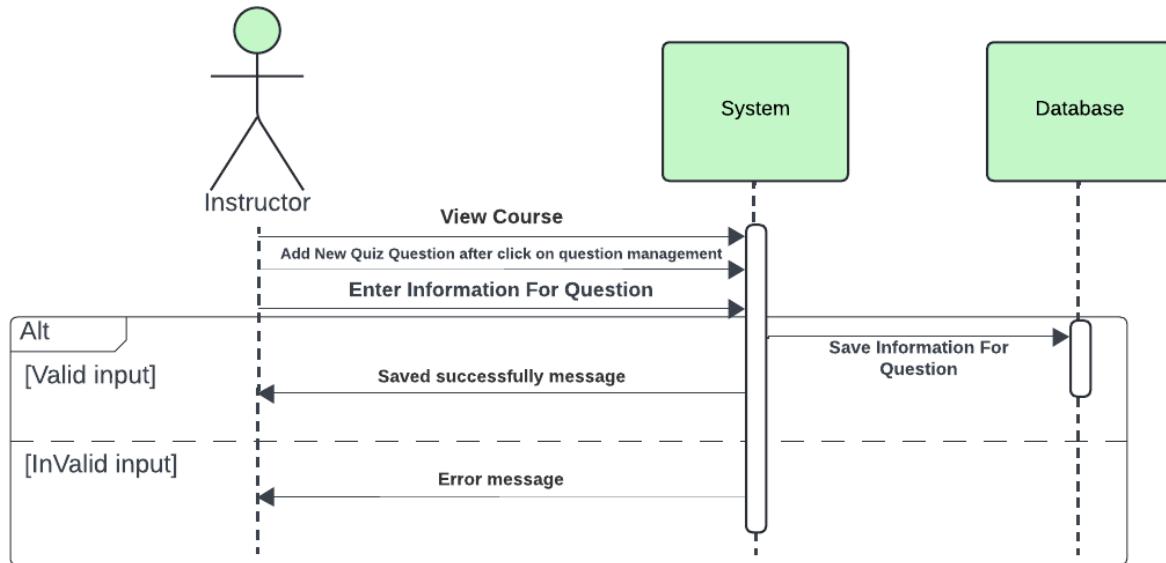


Figure 43: Sequence Diagram: Instructor Add Quiz Question

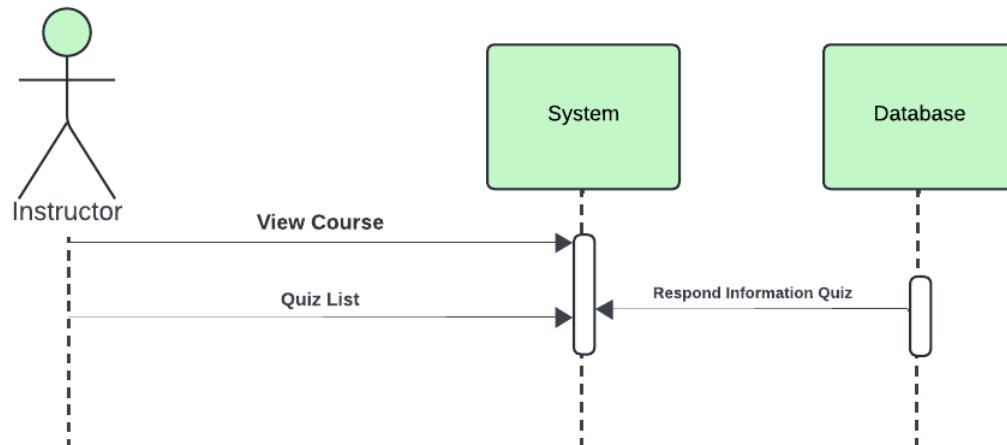


Figure 44: Sequence Diagram: Instructor Display Quizzes List

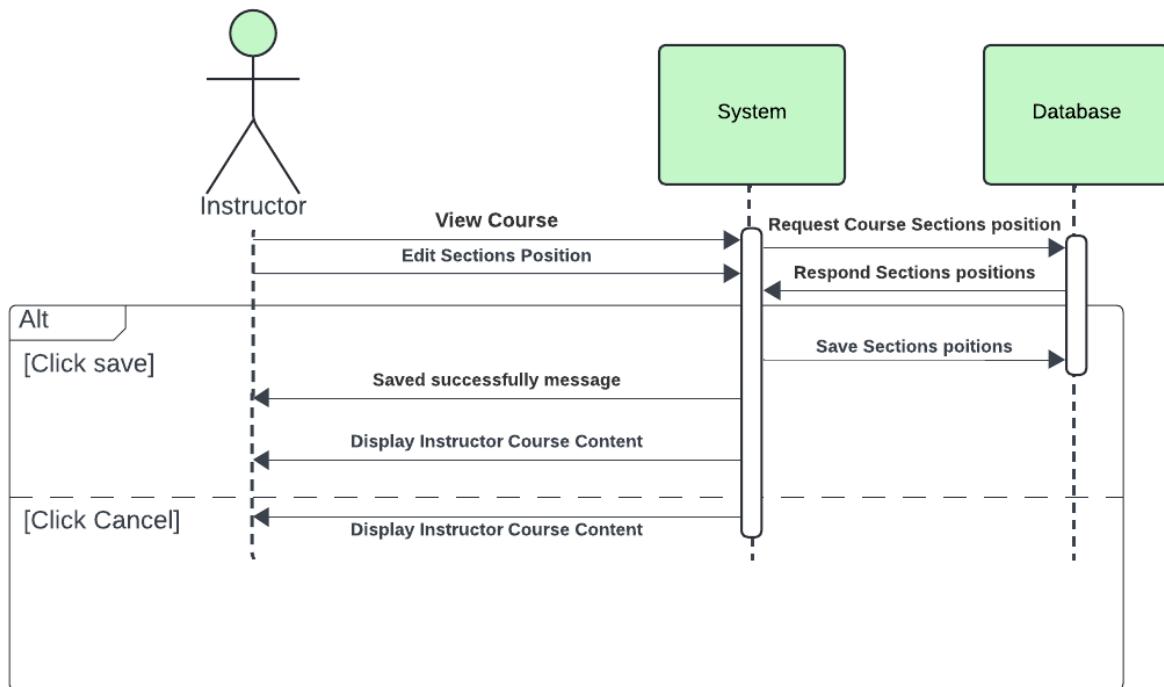


Figure 45: Sequence Diagram: Instructor Edit Section Position

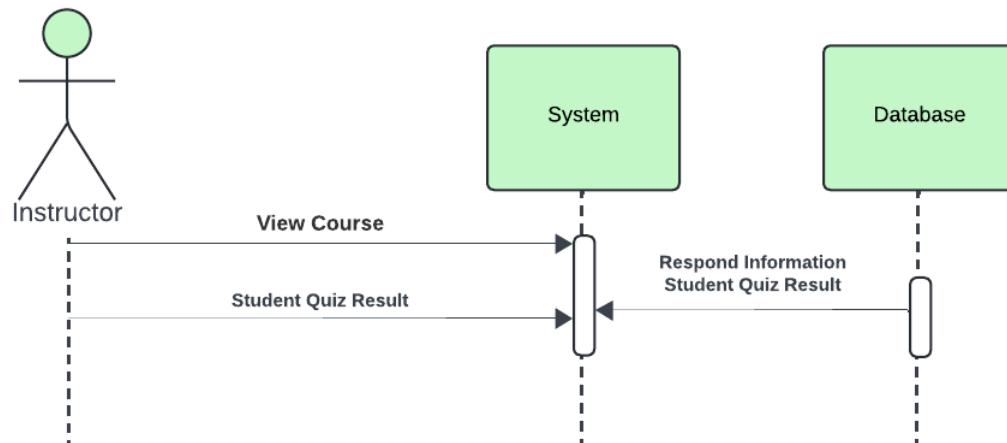


Figure 46: Sequence Diagram: Instructor: Display Student Quiz Result

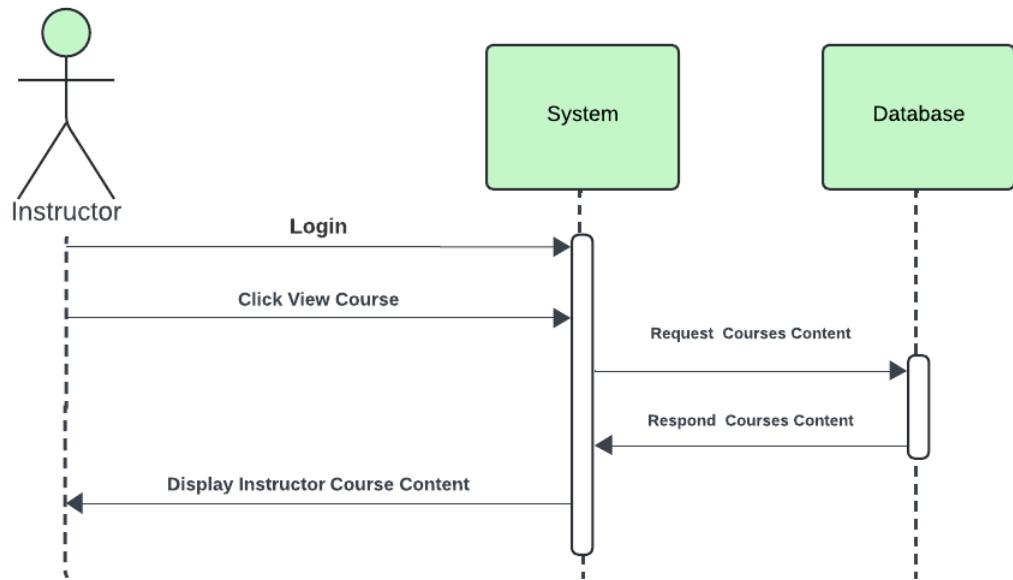


Figure 47: Sequence Diagram: Instructor Display Courses Content

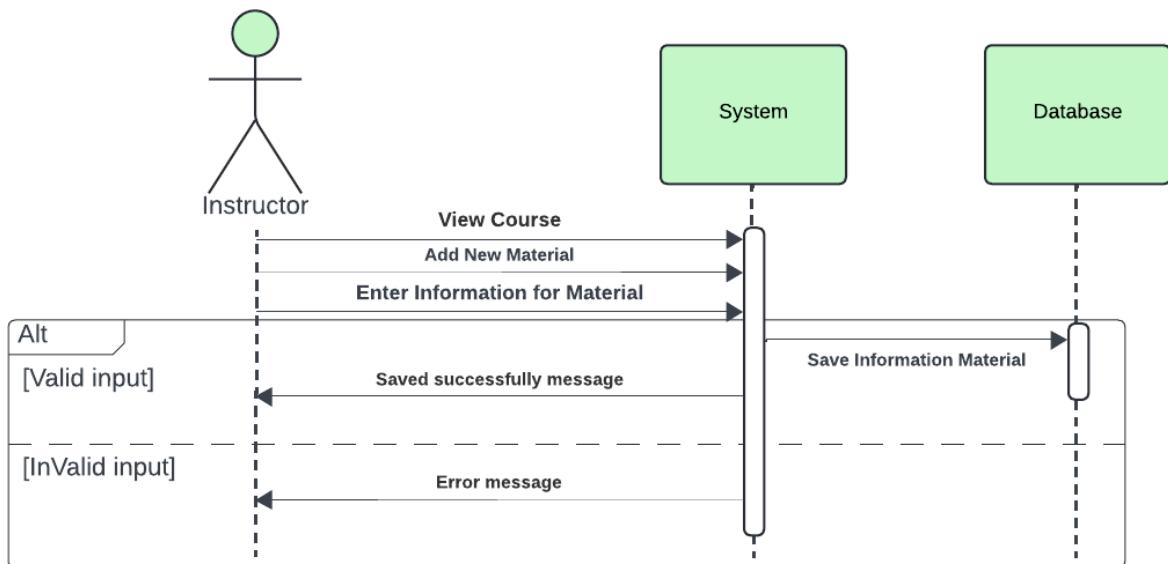


Figure 48: Sequence Diagram: Instructor Add Material

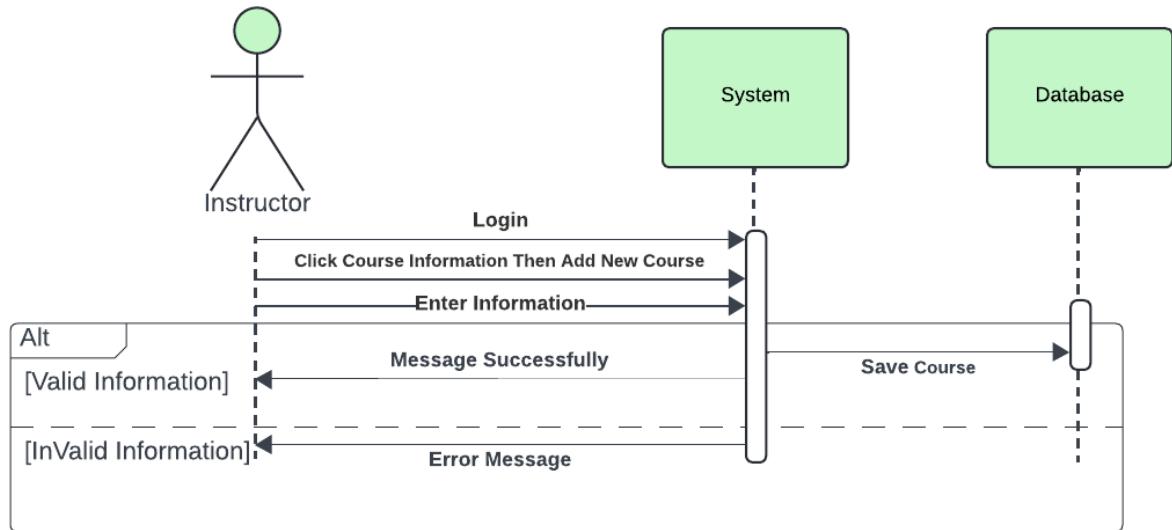


Figure 49: Sequence Diagram: Instructor Add New Course

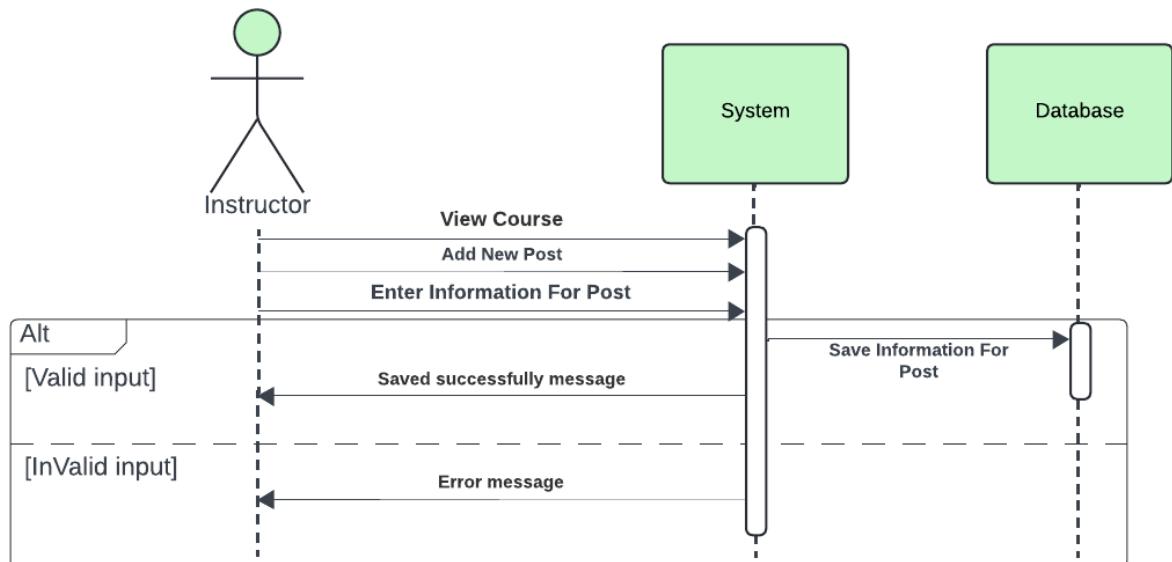


Figure 50: Sequence Diagram: Instructor Add New Post

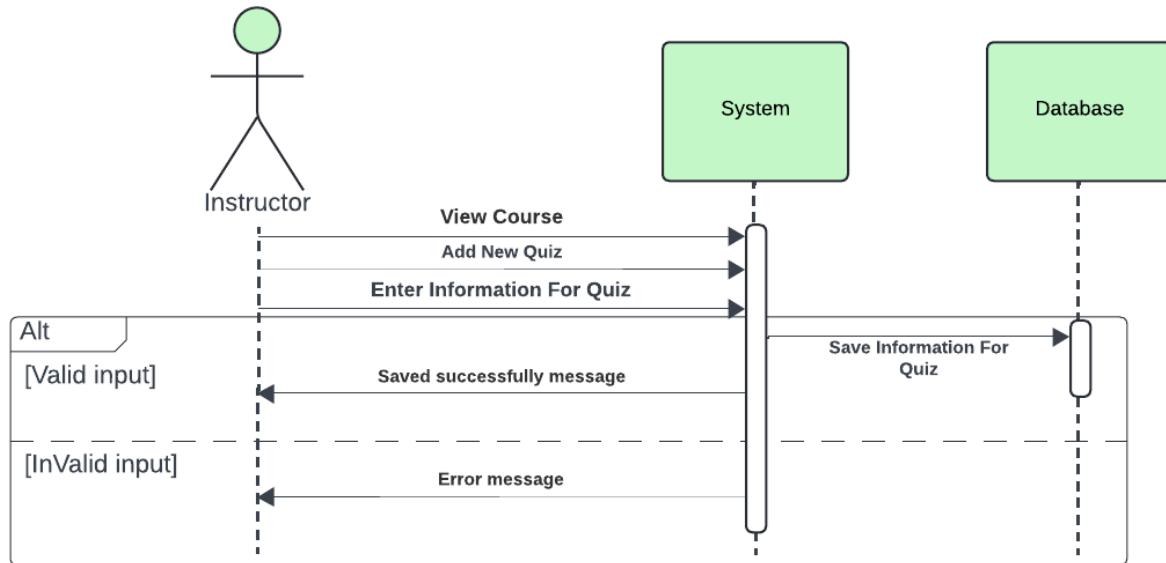


Figure 51: Sequence Diagram: Instructor Add New Quiz

4.5.1 admin sequence

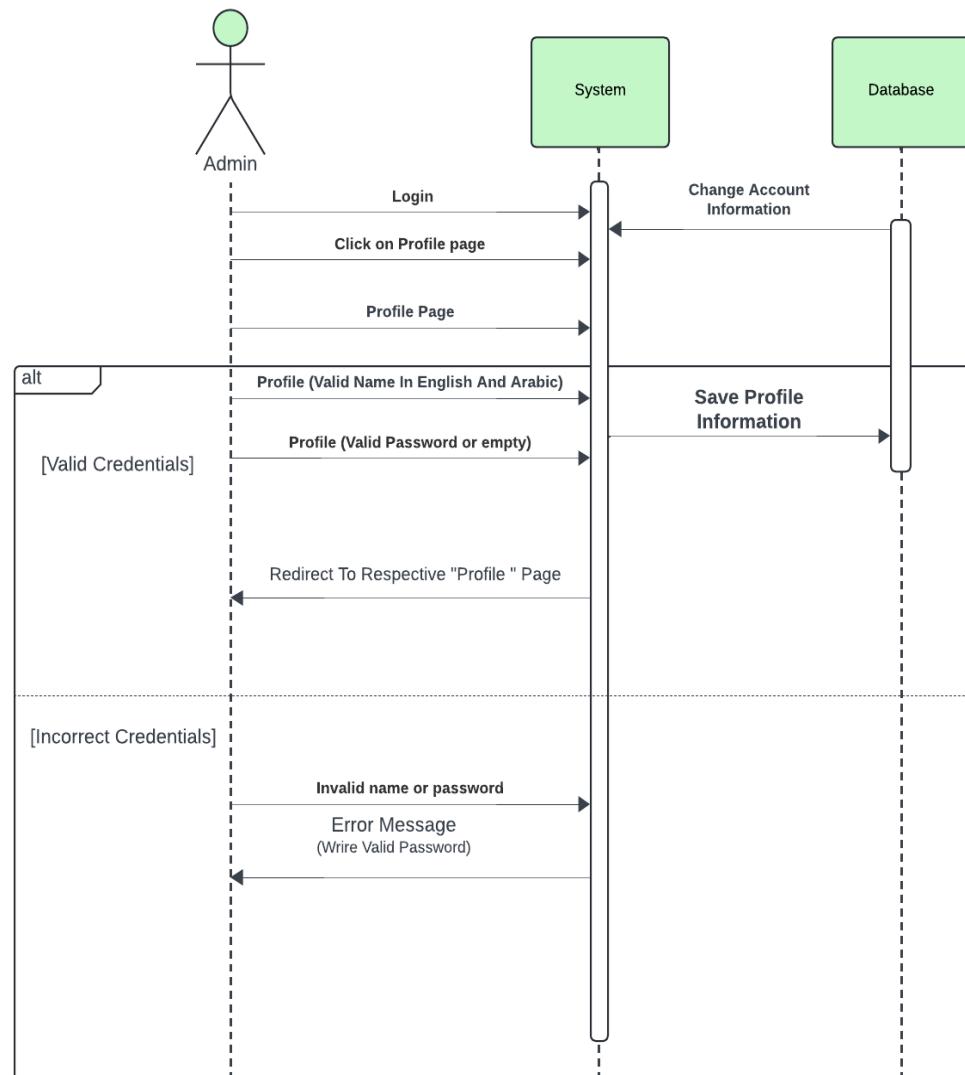


Figure 52: Sequence Diagram: Admin Edit Profile

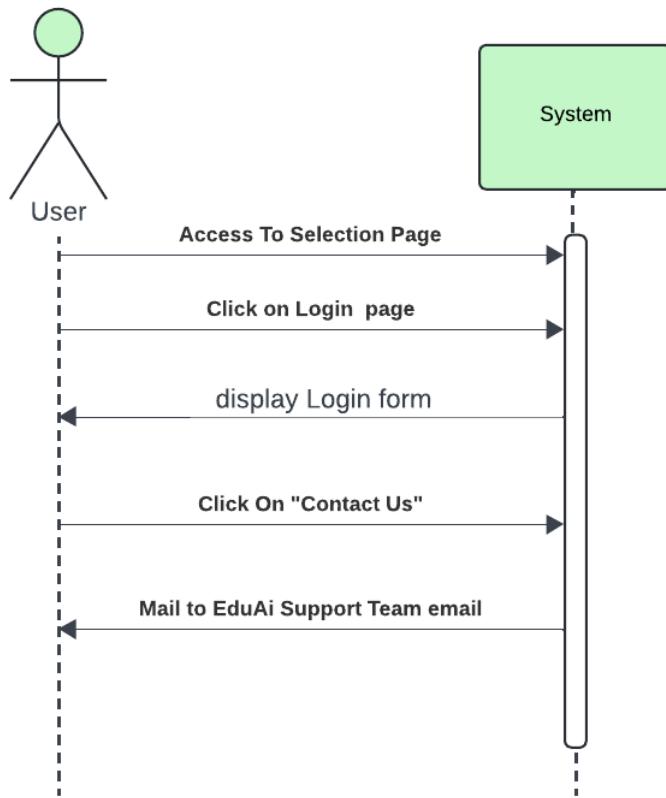


Figure 53: Sequence Diagram: User Contact us

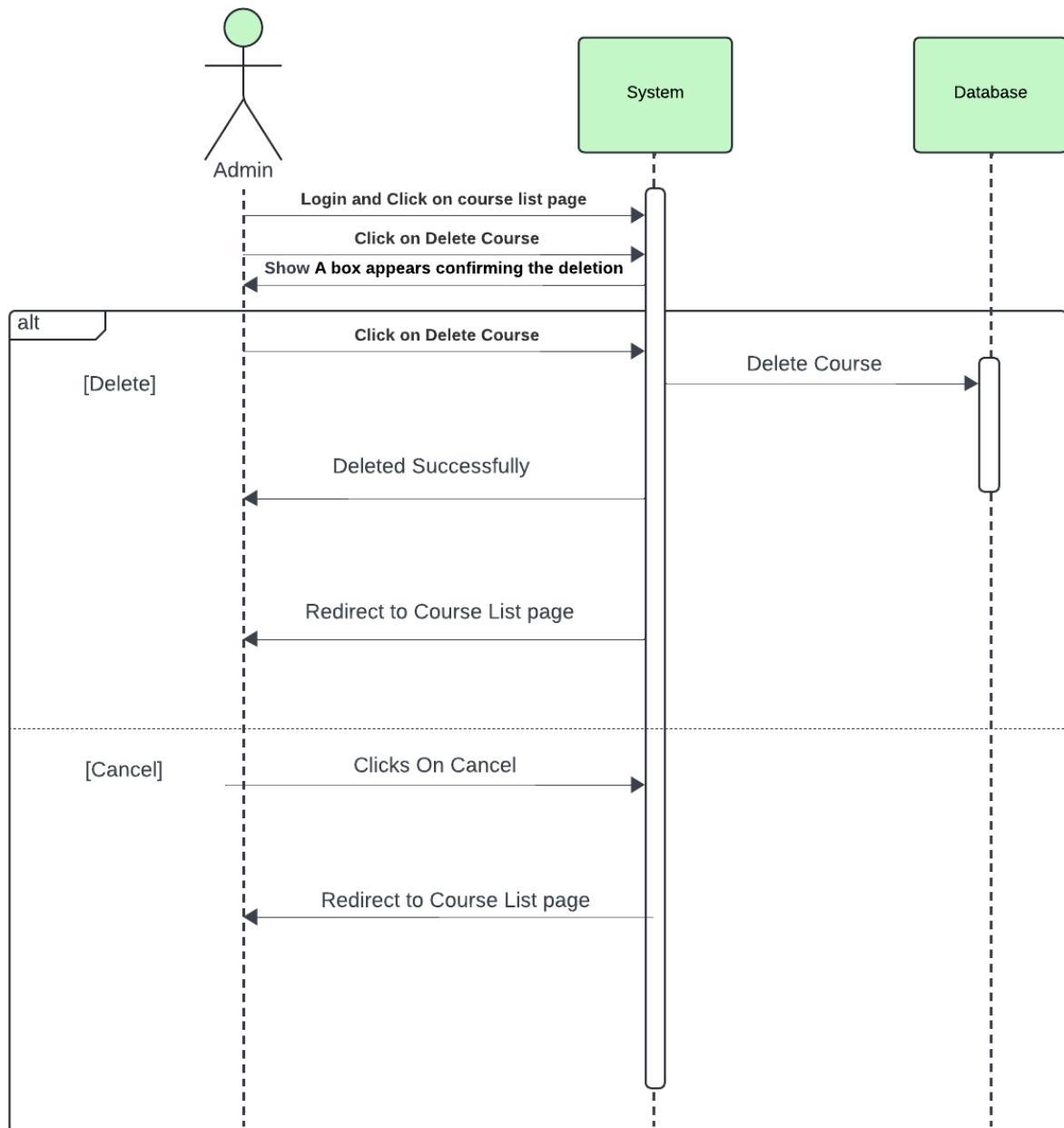


Figure 54: Sequence Diagram: Admin Delete Course

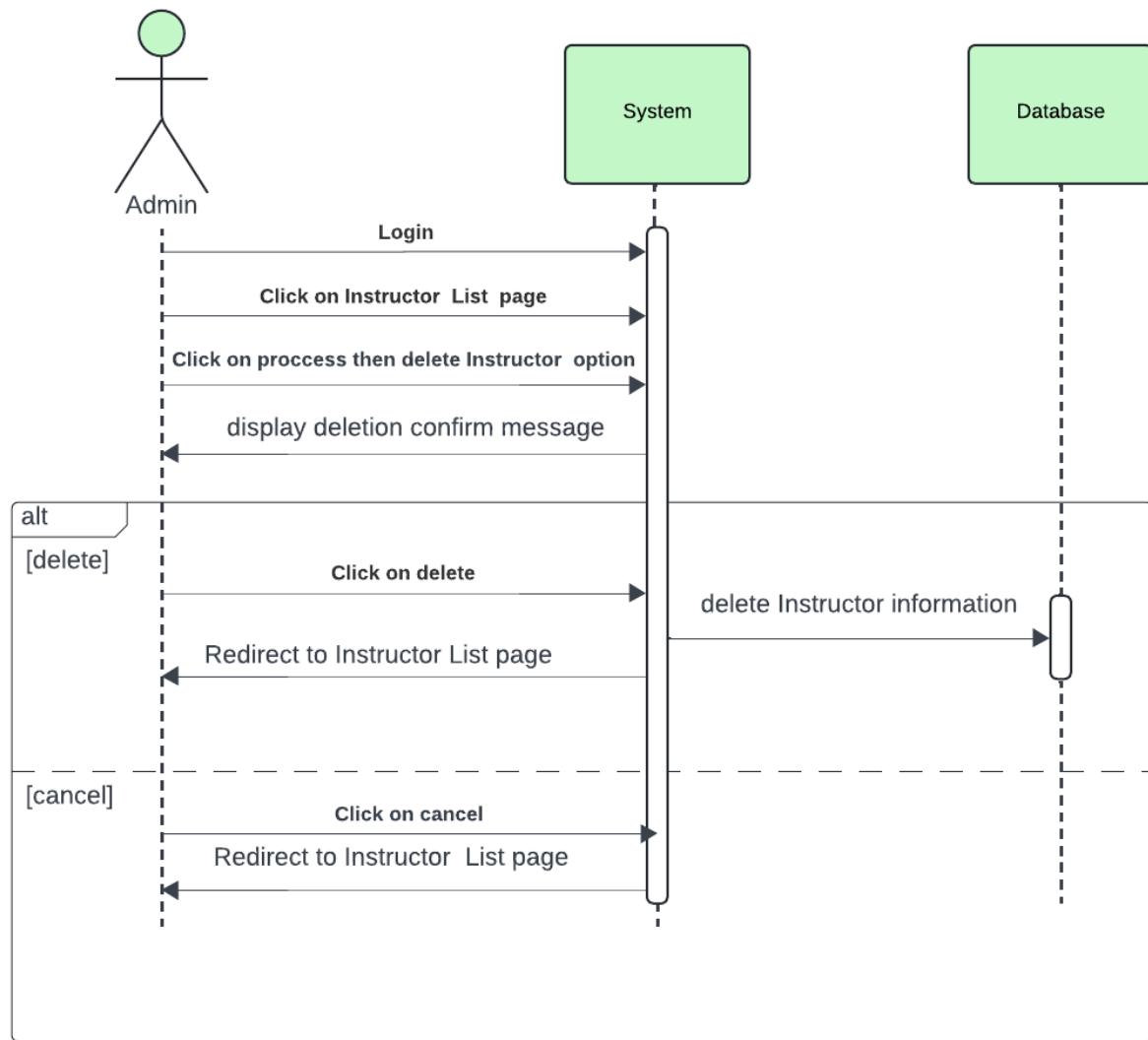


Figure 55: Sequence Diagram: Admin Delete Instructor

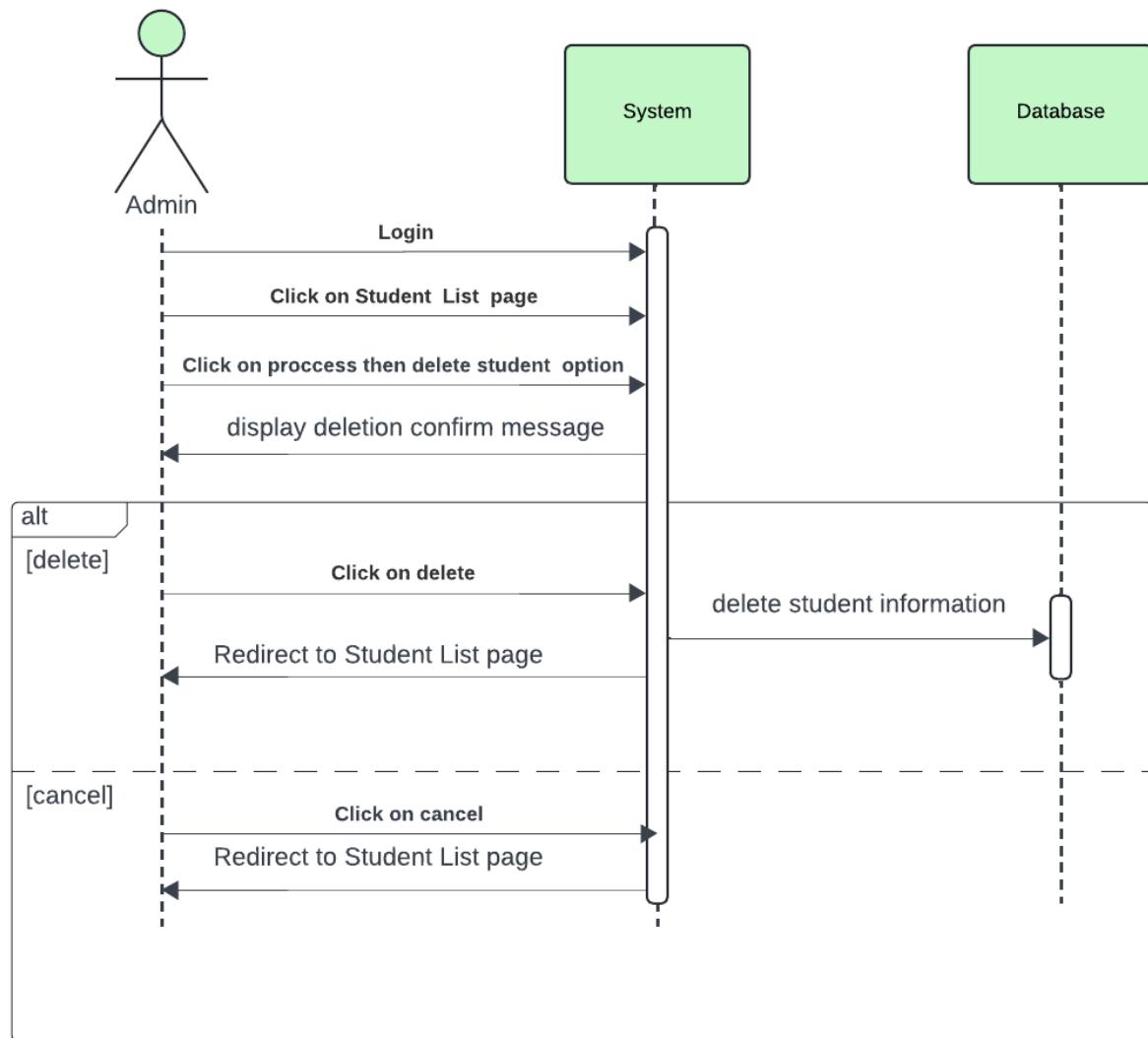


Figure 56: Sequence Diagram: Admin Delete Student

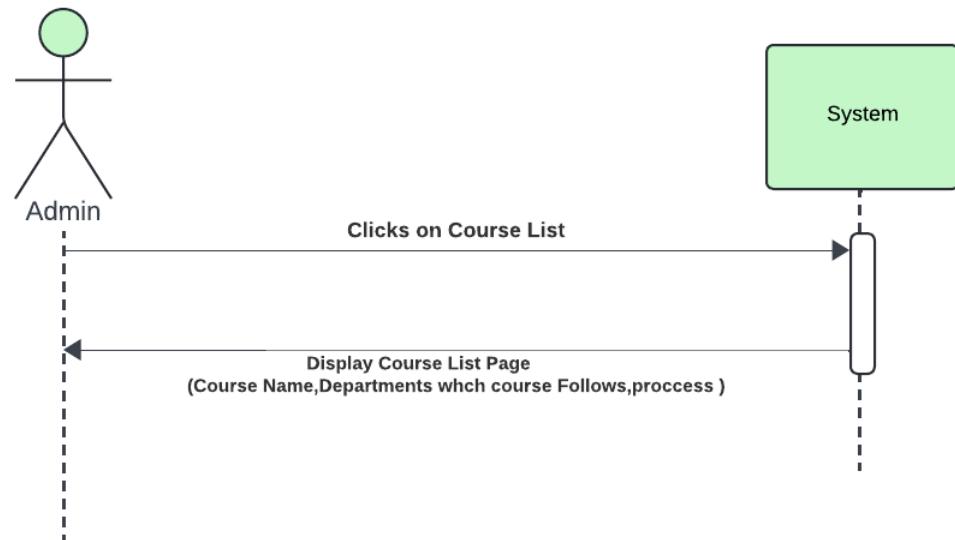


Figure 57: Sequence Diagram: Admin Display Courses List

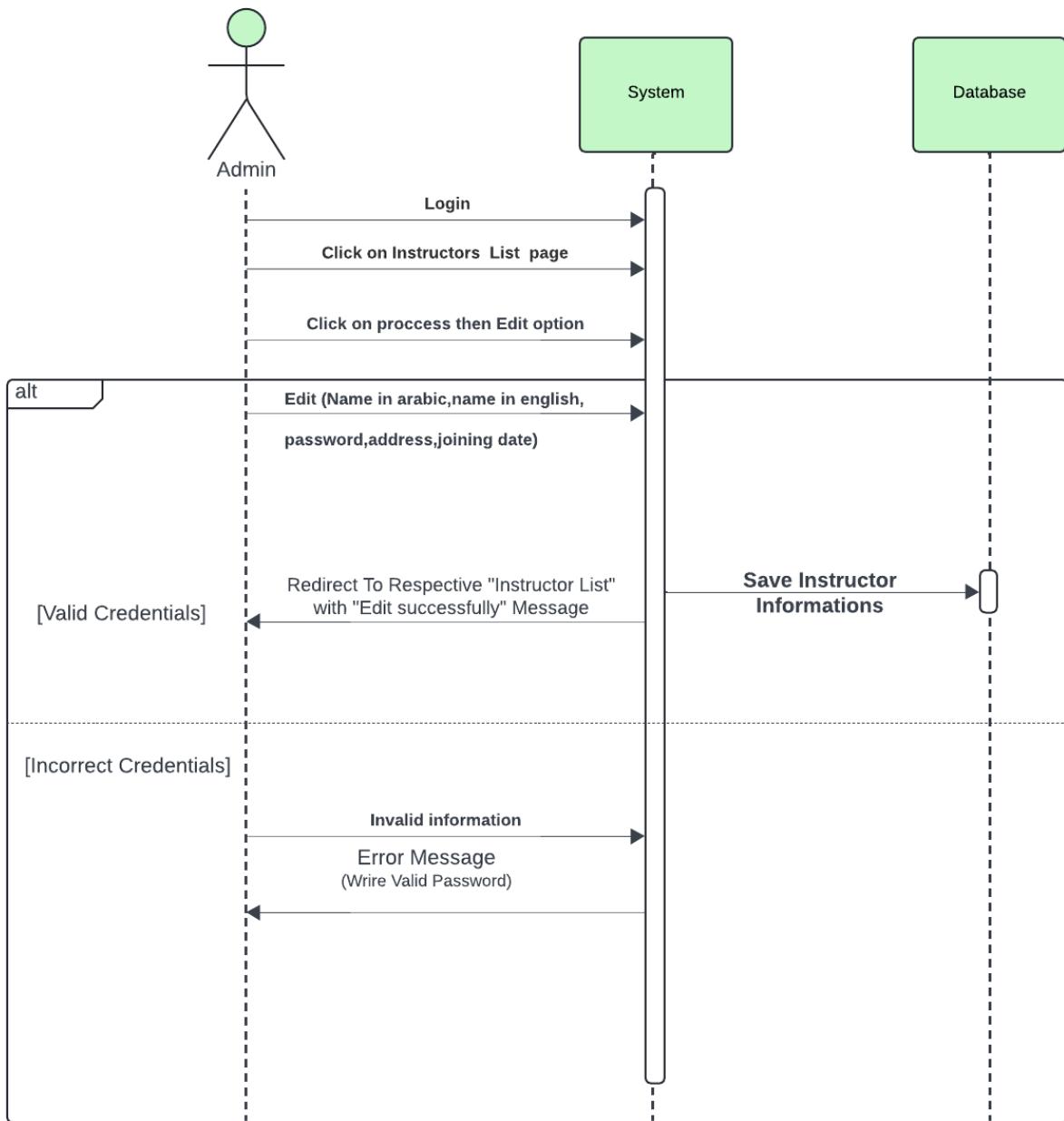


Figure 58: Sequence Diagram: Admin Edit Instructor

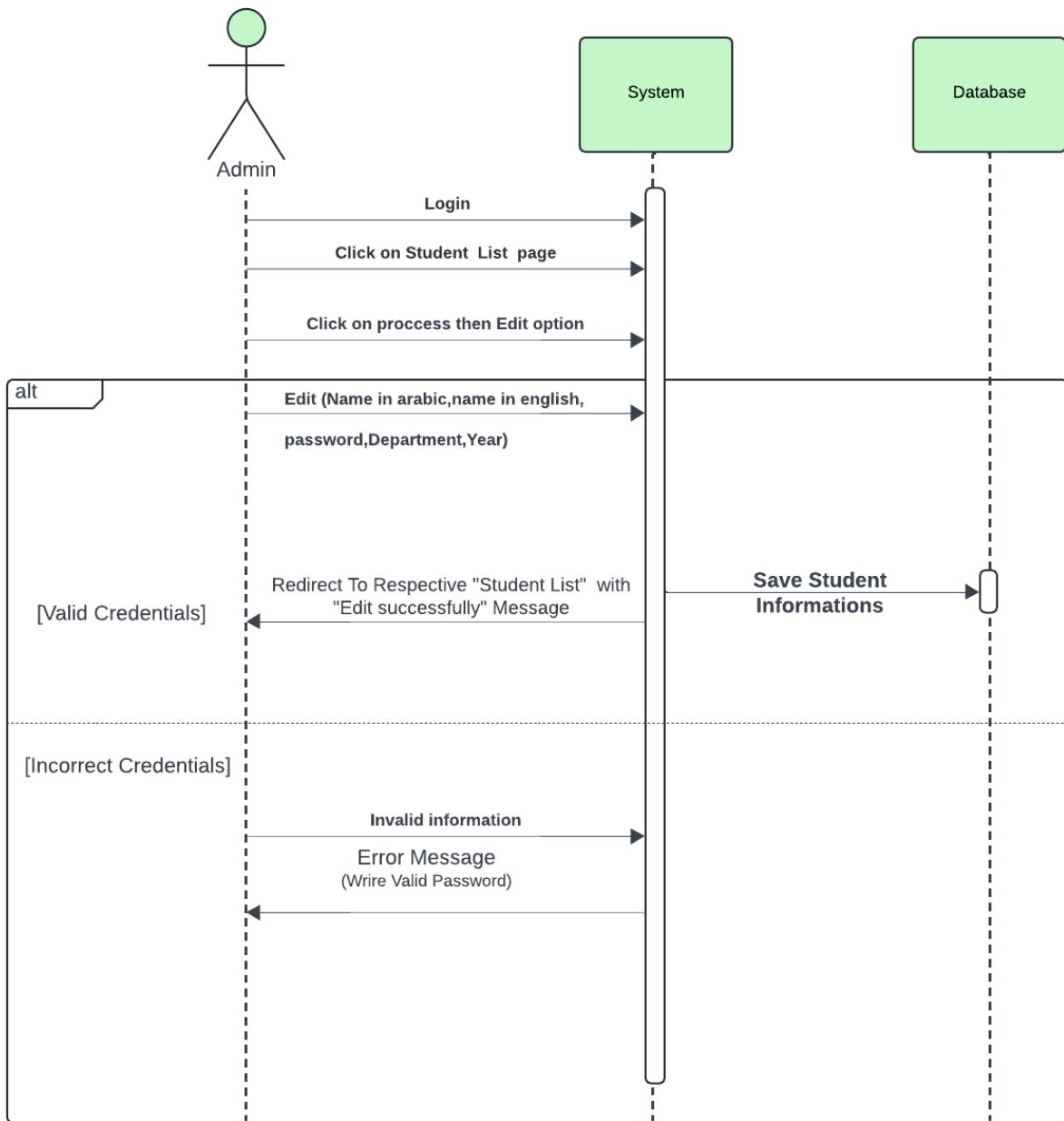


Figure 59: Sequence Diagram: Admin Edit Student List

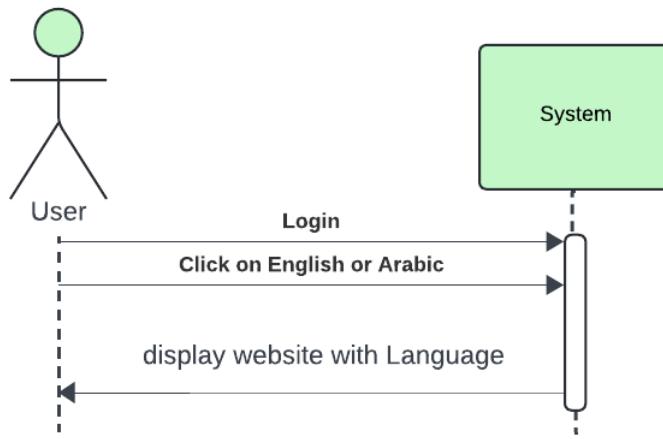


Figure 60::Sequence Diagram: User Change Language

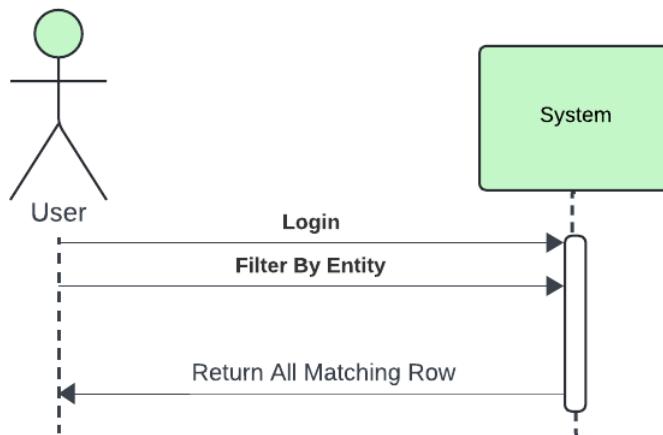


Figure 61:Sequence Diagram: User Filter By Entity

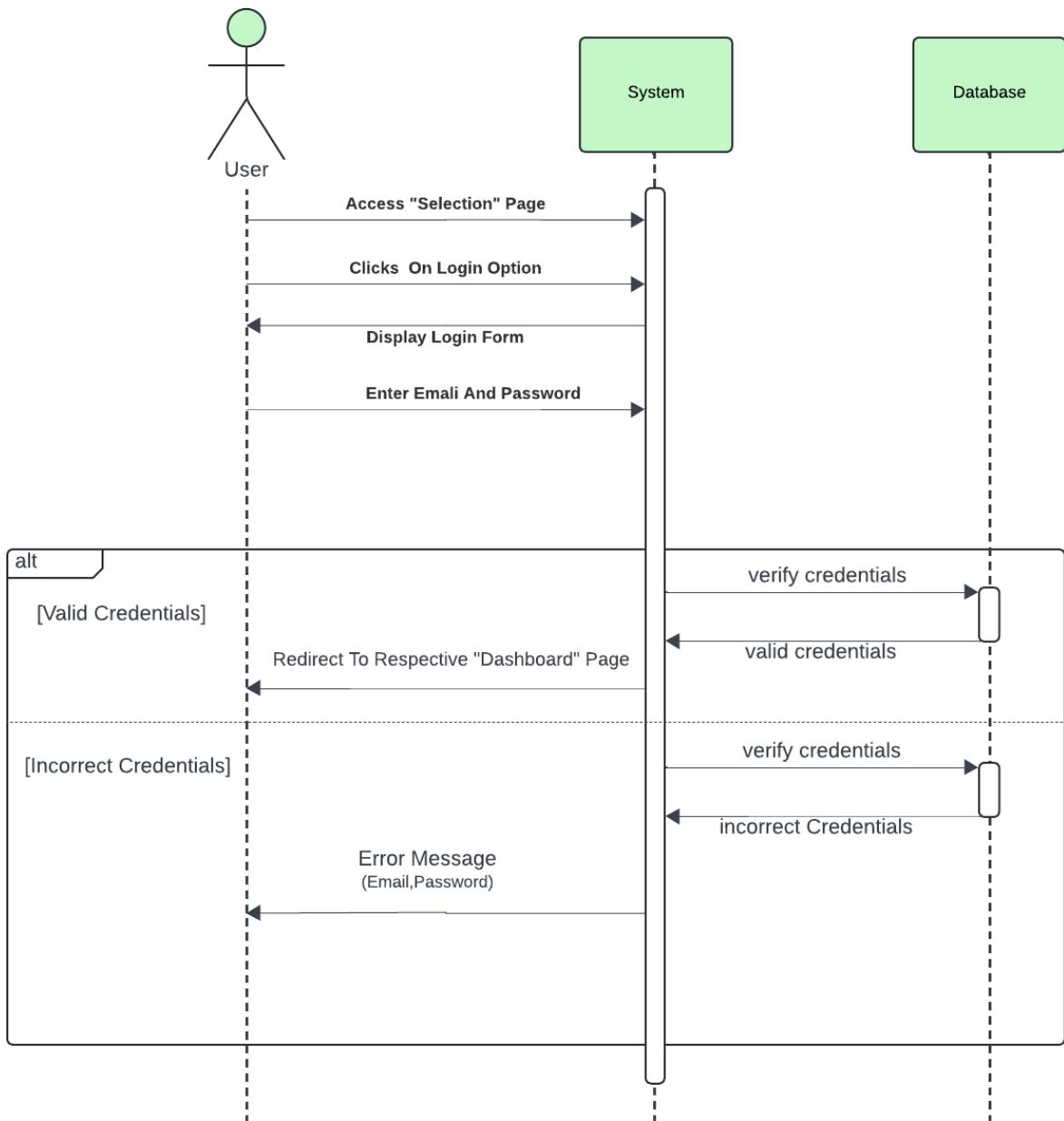


Figure 62: Sequence Diagram: User Login

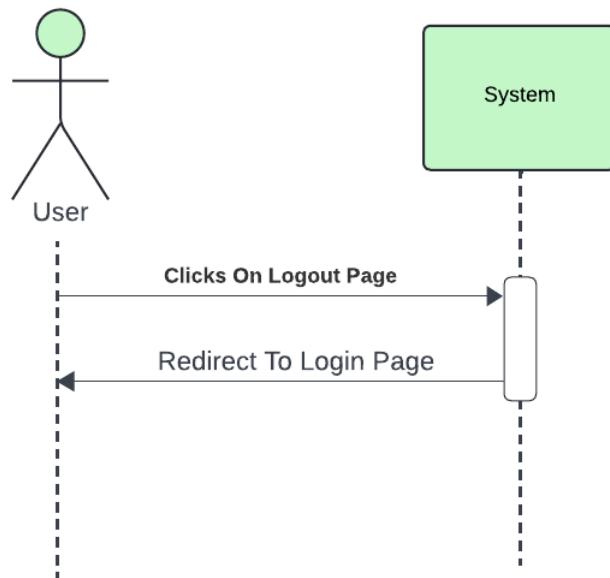


Figure 63::Sequence Diagram: User Logout

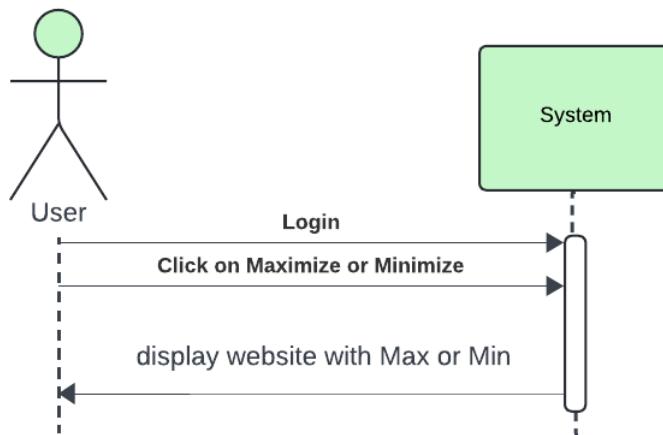


Figure 64:Sequence Diagram: User Maximize Or Minimize Website

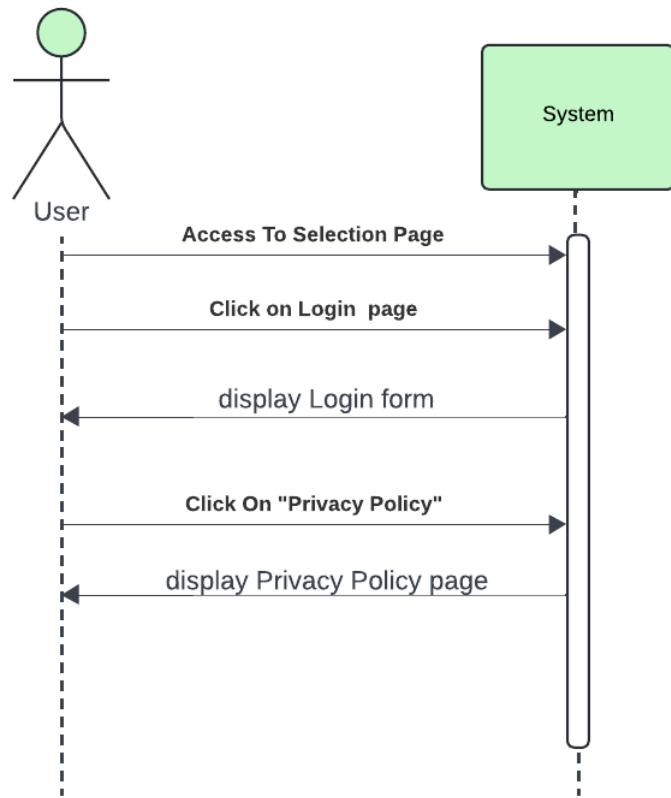


Figure 65::Sequence Diagram: User Display privacy and policy

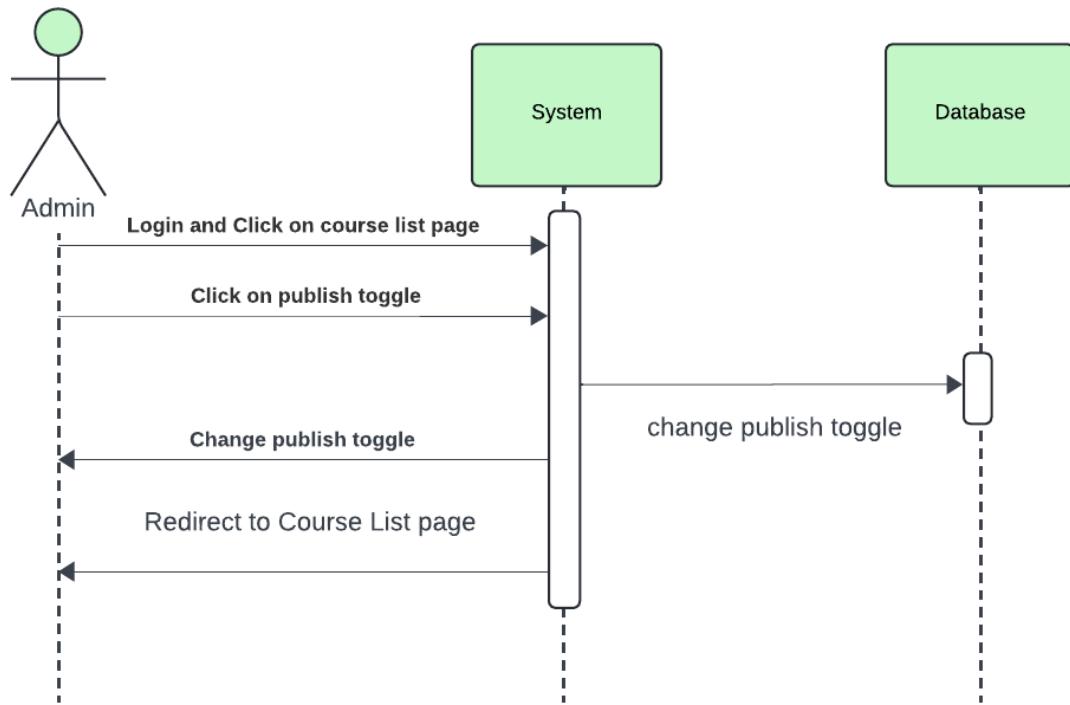


Figure 66::Sequence Diagram: Admin Publish Toggle

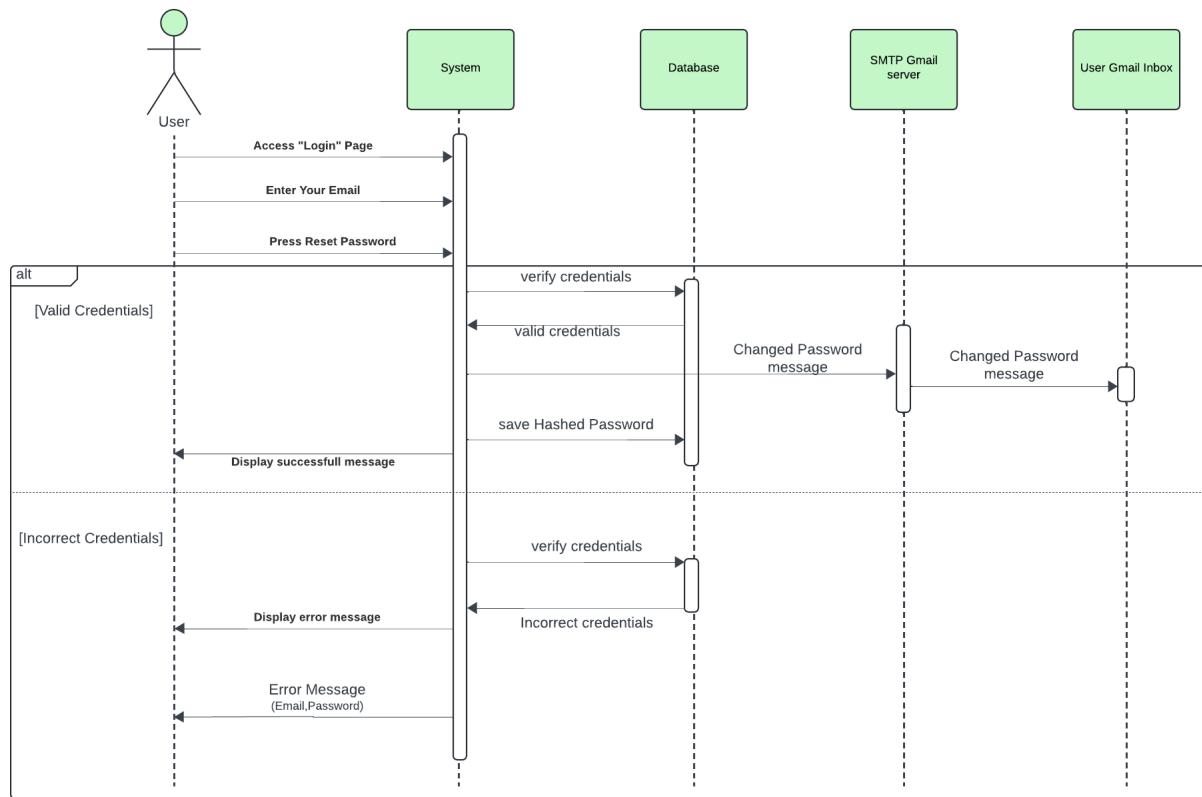


Figure 67::Sequence Diagram: User Reset Password

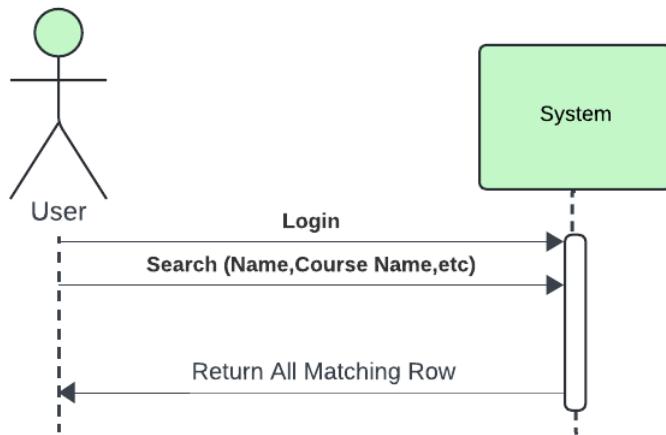


Figure 68: Sequence Diagram: User Search

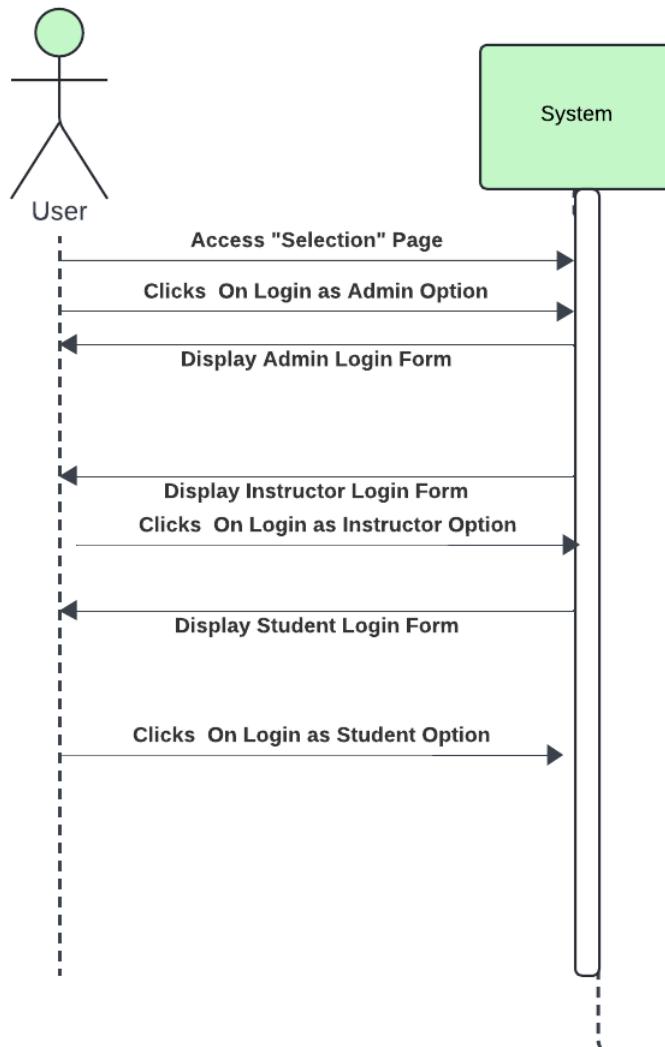


Figure 69: Sequence Diagram: User Selection Page

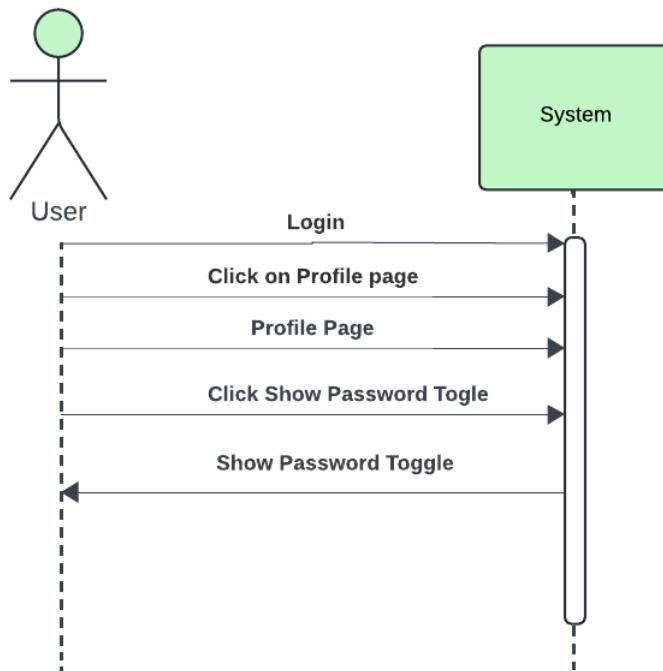


Figure 70: Sequence Diagram: User Show Password

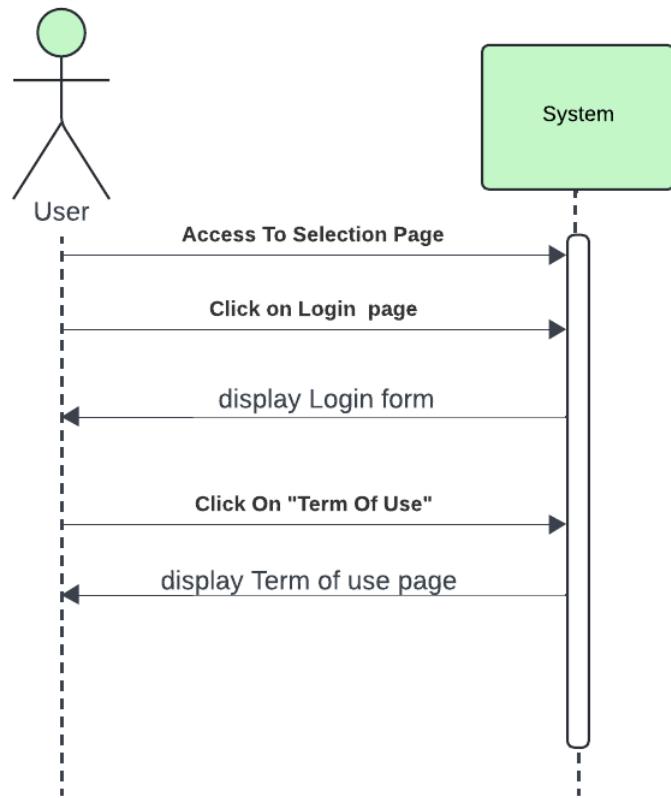


Figure 71: Sequence Diagram: User Term Of Use

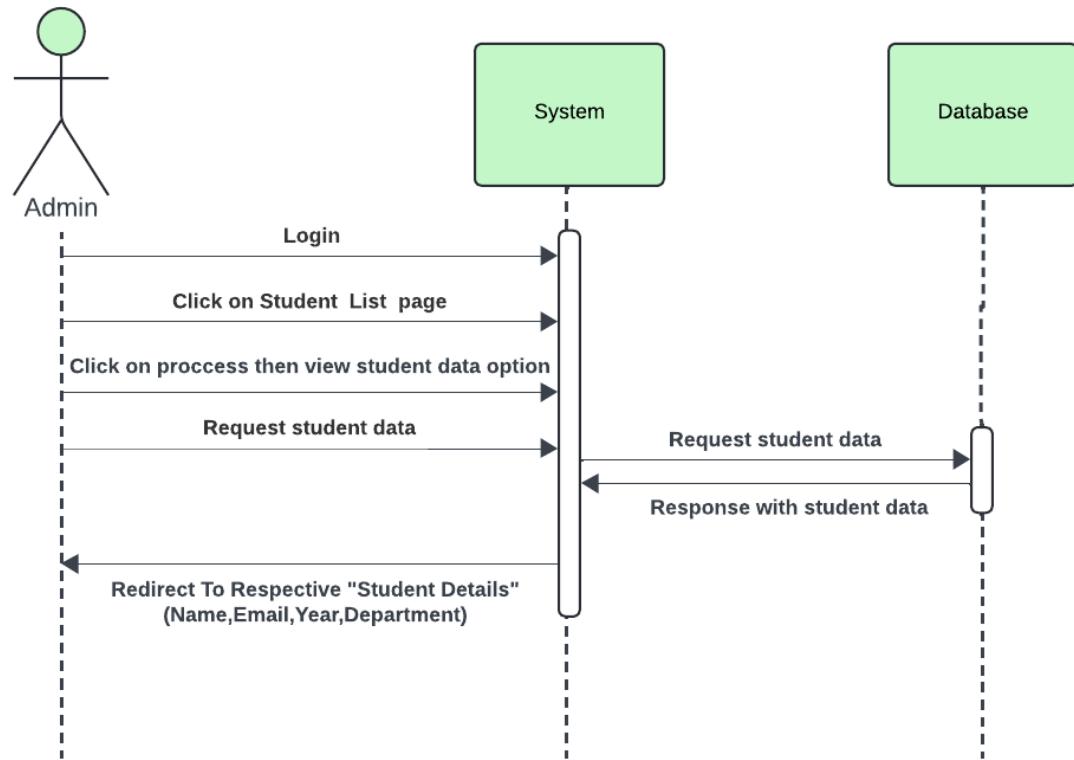


Figure 72: Sequence Diagram: Admin view Student

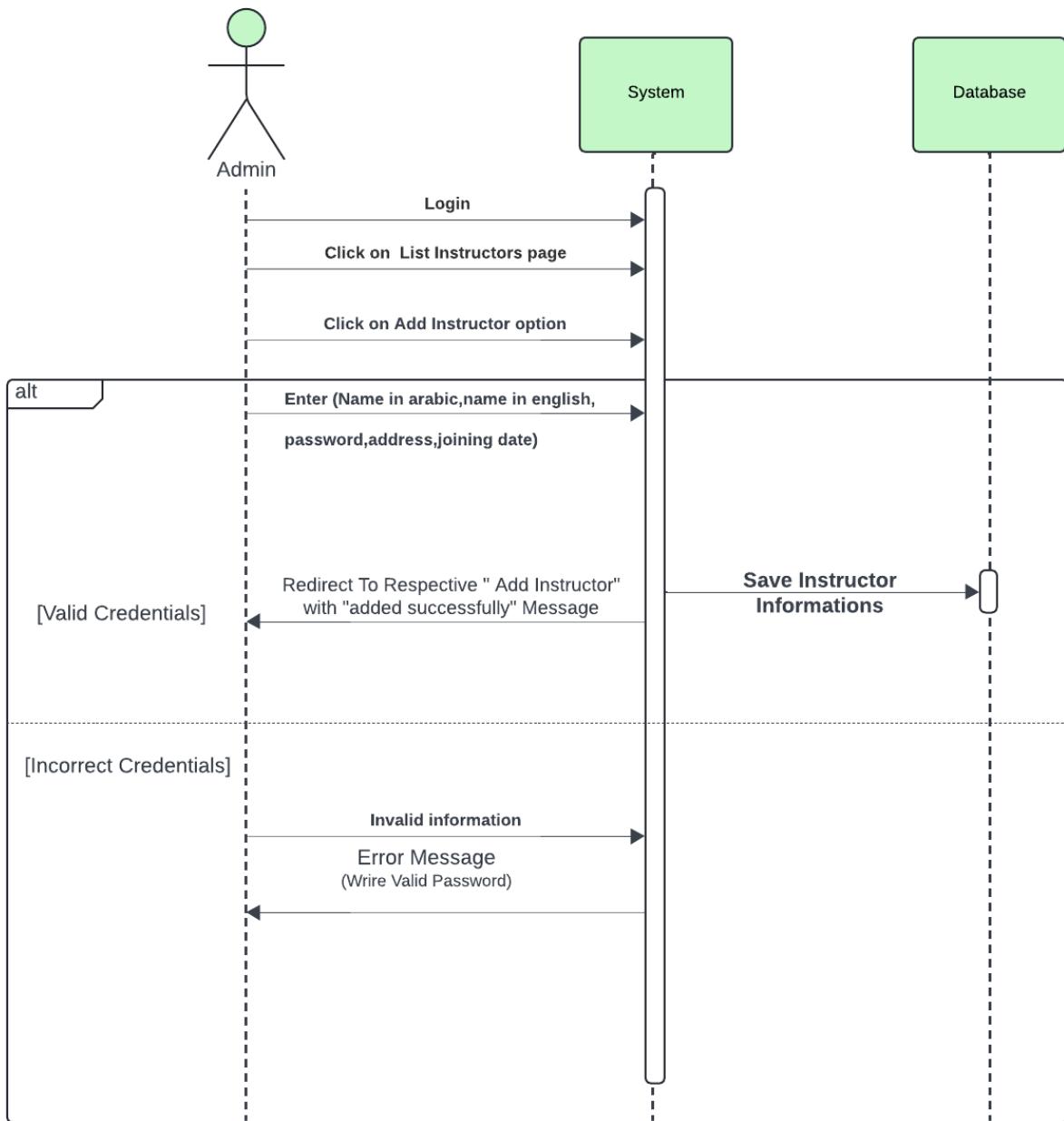


Figure 73: Sequence Diagram: Admin Add Instructor

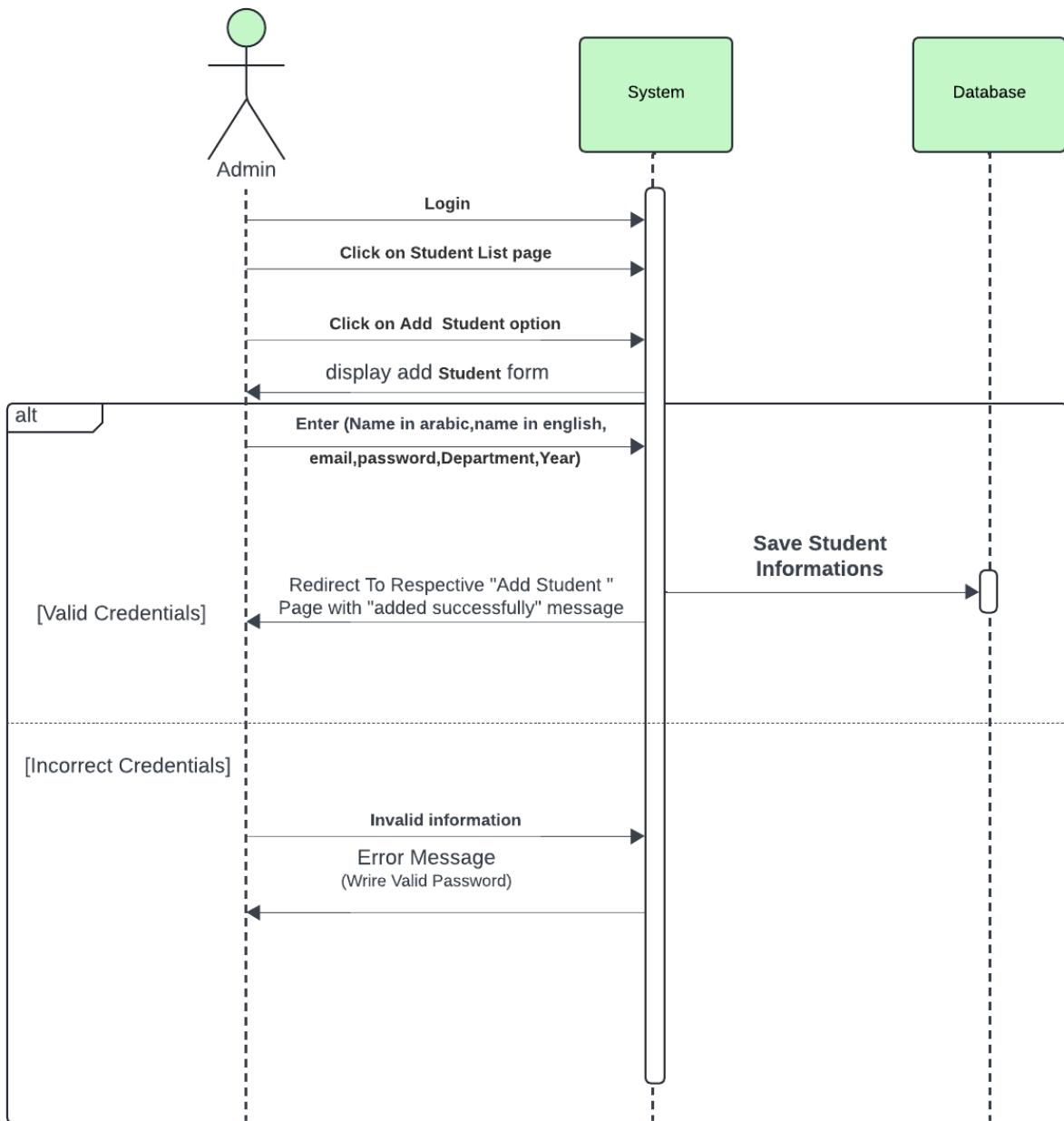


Figure 74: Sequence Diagram: Admin Add Student

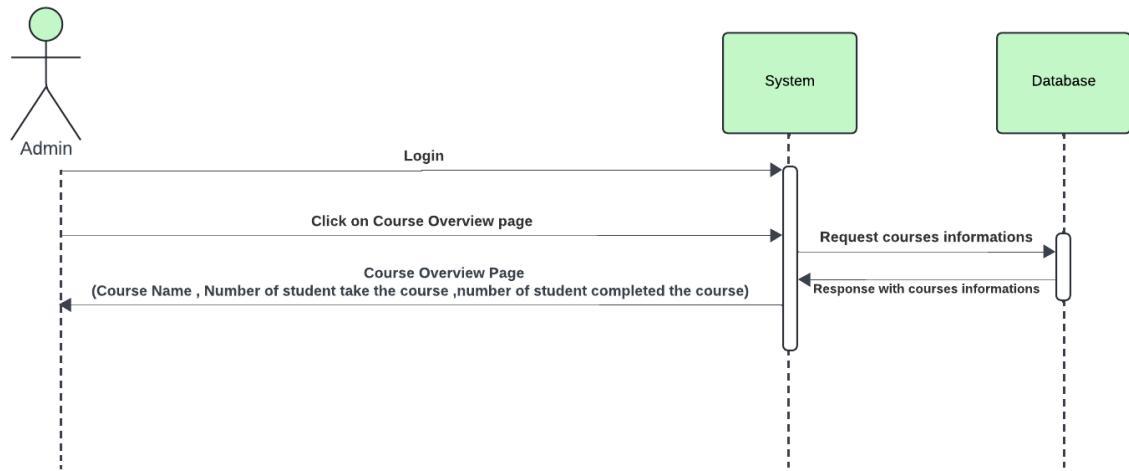


Figure 75: Sequence Diagram: Admin Course Overview

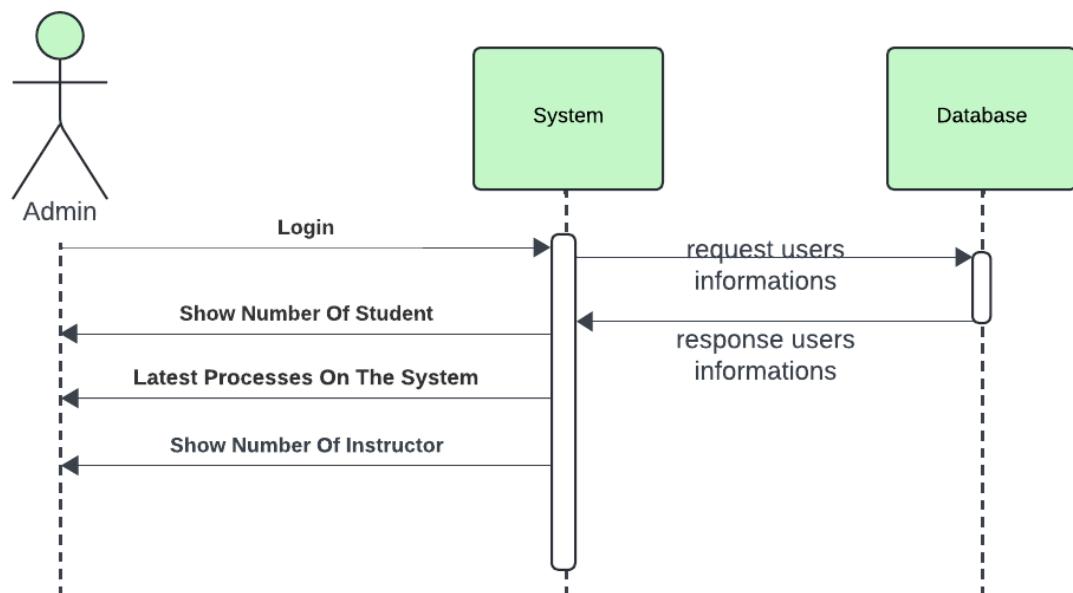


Figure 76: Sequence Diagram: Admin Dashboard

4.6 UML Class Diagram

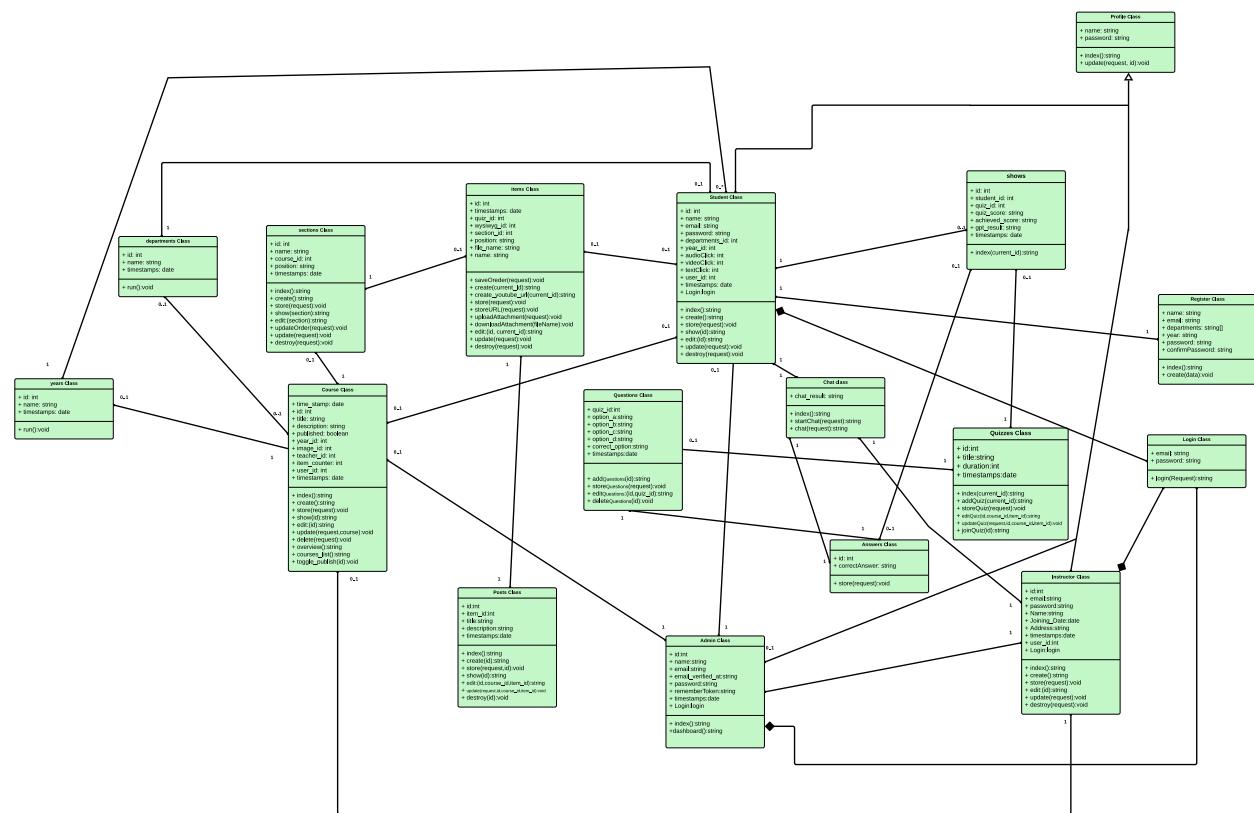


Figure 77: UML Class Diagram

4.7 graphical user interface design / low fidelity prototype

Before Login Page

1 / 6

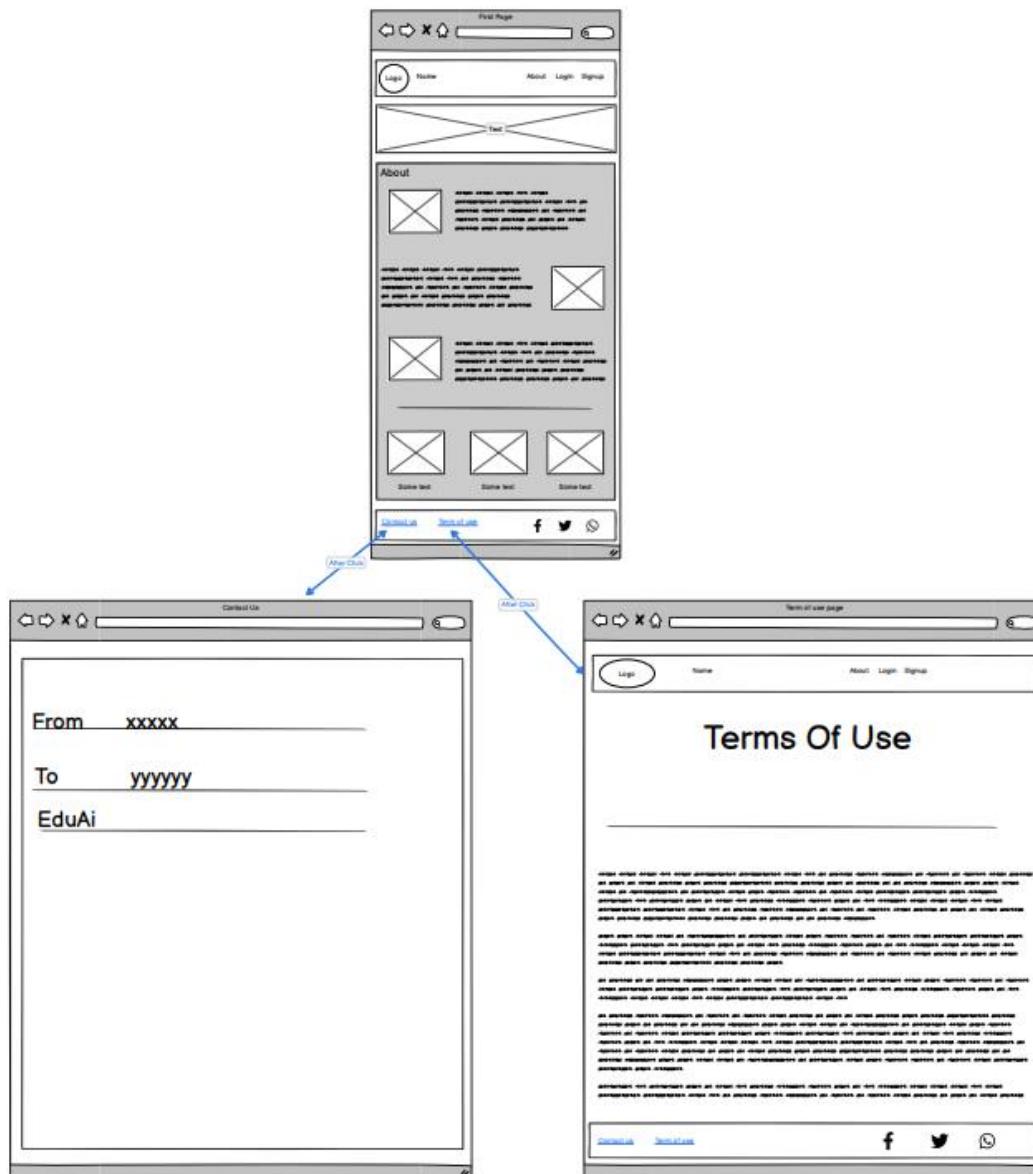


Figure 78: low fidelity: Home Page

The wireframe depicts a registration form titled "Sign Up". It includes fields for "Full Name", "Email", "Password", "Department" (with a dropdown arrow), and "Current Year" (with a dropdown arrow). A checkbox labeled "You agree to our Terms and Privacy Policy" is present, followed by a blue "Sign Up" button. Below the button, text reads "Already a member? [Login](#)". At the bottom, there are links for "Contact us" and "Term of use", along with social media icons for Facebook, Twitter, and a speech bubble. A search bar is located at the top right.

SignUp

Logo Name About Login Signup

Sign Up

Full Name

Email

Password

Department

Current Year

You agree to our Terms and Privacy Policy

Sign Up

Already a member? [Login](#)

Contact us Term of use

f t <#>

Figure 79: low fidelity: Register Page

Login

2 / 6

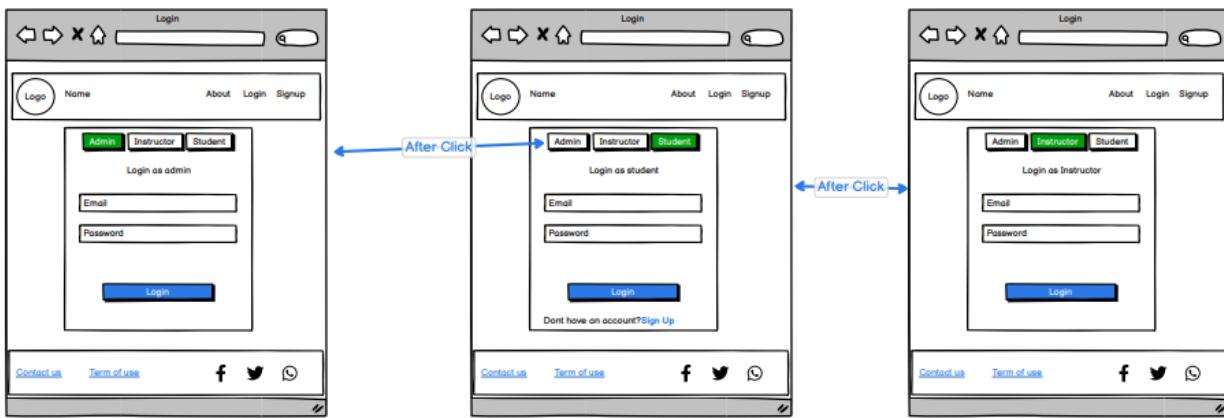


Figure 80: low fidelity: Login Page

Students Page

4 / 6

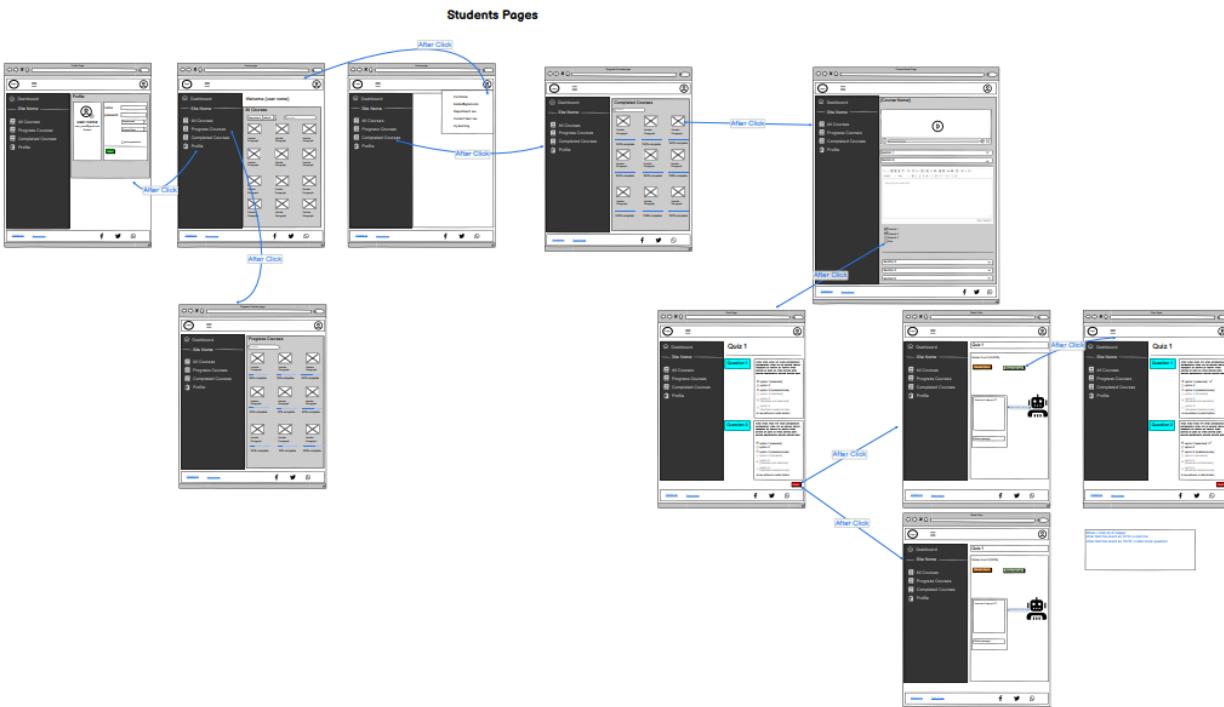


Figure 81: low fidelity: Students Pages

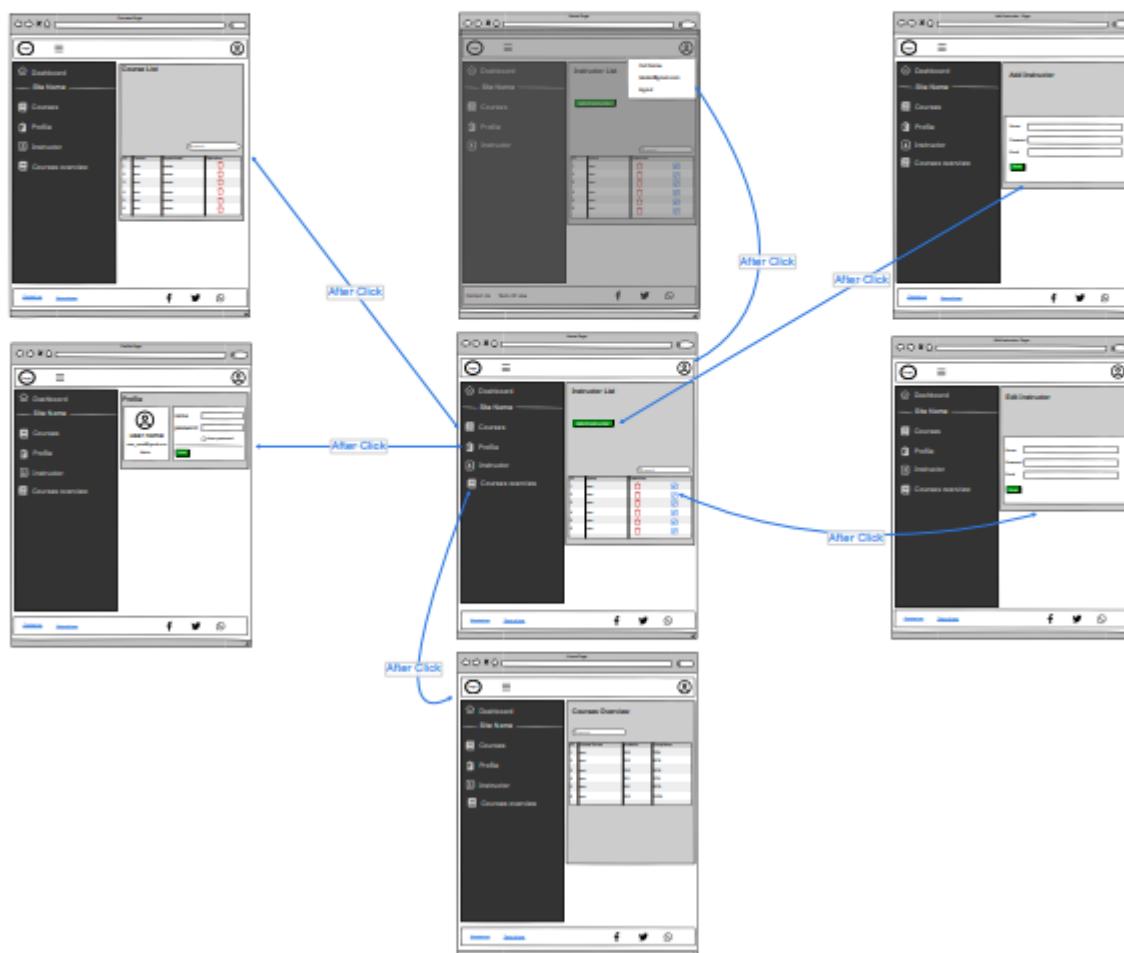
Admin Pages

Figure 82:low fidelity: Admin Pages

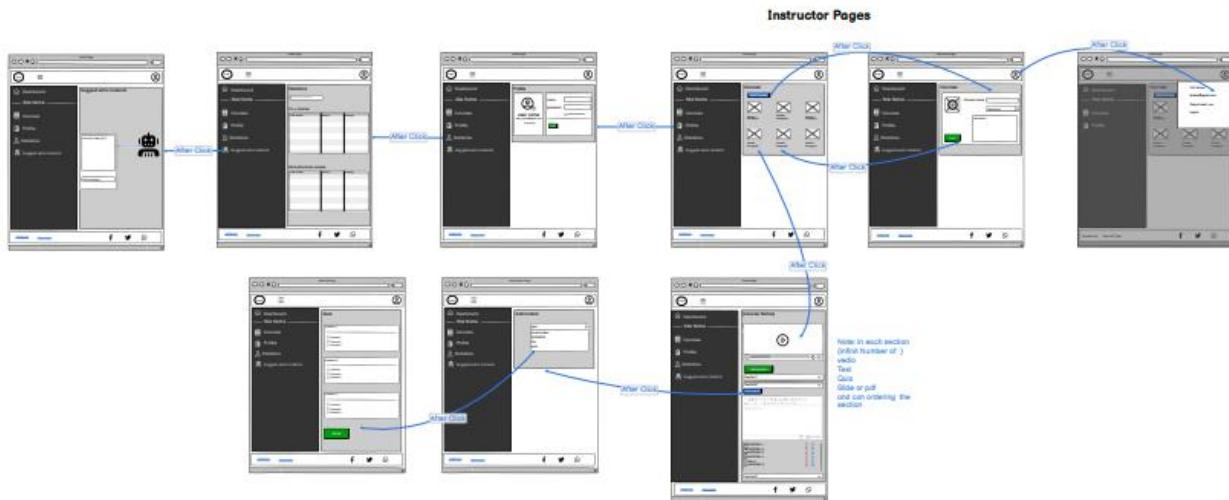


Figure 83: low fidelity: Instructor Pages

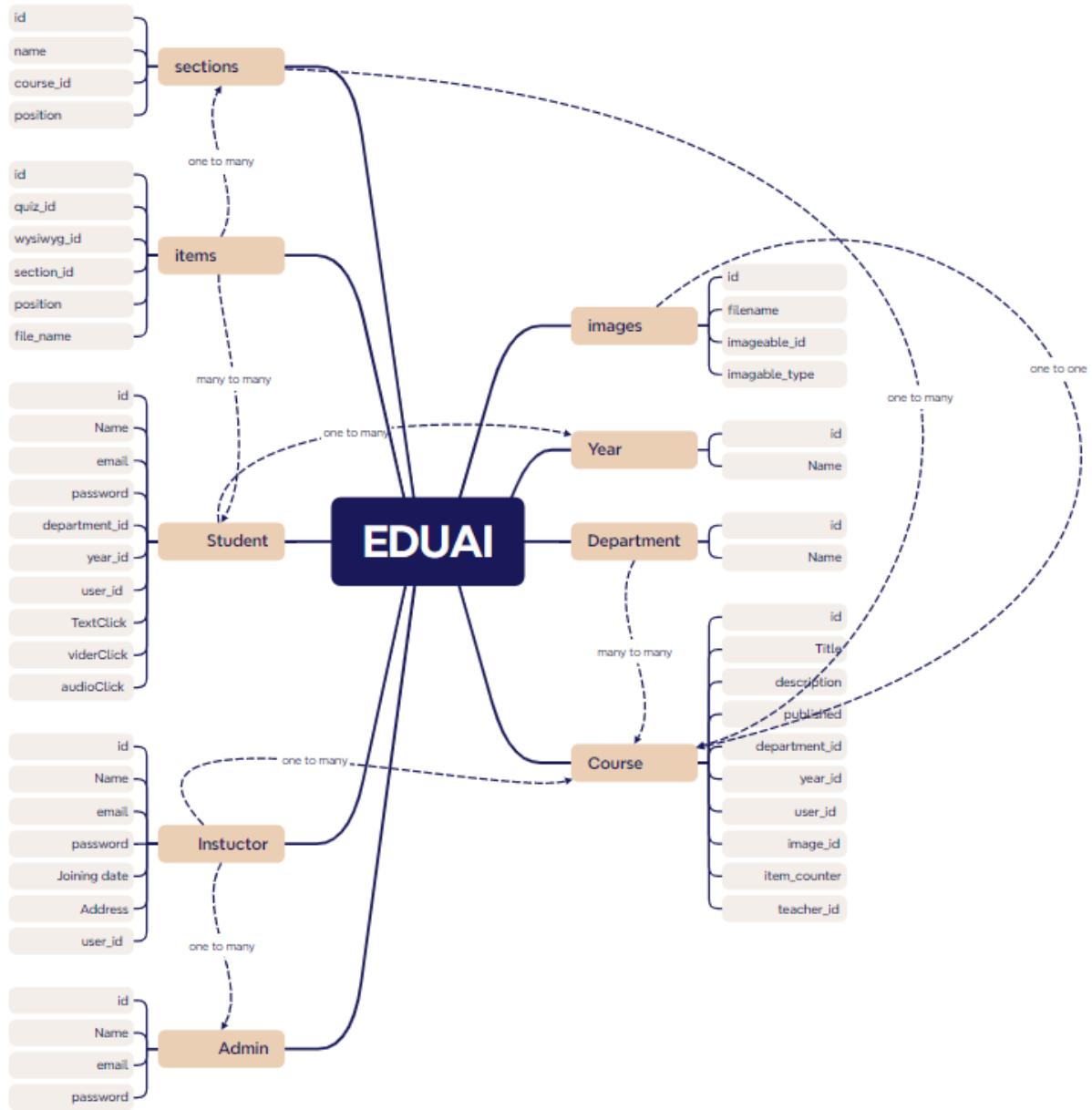


Figure 84: Database Design

Chapter five: System Implementation

5.1 database implementation

```
SELECT * FROM `posts`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id	title	description	created_at	updated_at
-----------	--------------	--------------------	-------------------	-------------------

Figure 85:Database Implementation 1

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

```
SELECT * FROM `questions`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id	quiz_id	question	option_a	option_b	option_c	option_d	correct_option	created_at	updated_at
-----------	----------------	-----------------	-----------------	-----------------	-----------------	-----------------	-----------------------	-------------------	-------------------

Query results operations

Figure 86:Database Implementation 2

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0099 seconds.)

```
SELECT * FROM `quizzes`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id	title	duration	from_time	to_time	created_at	updated_at
-----------	--------------	-----------------	------------------	----------------	-------------------	-------------------

Figure 87:Database Implementation 3

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0102 seconds.)

```
SELECT * FROM `results`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

id	student_id	quiz_id	quiz_score	achieved_score	created_at	updated_at
-----------	-------------------	----------------	-------------------	-----------------------	-------------------	-------------------

Figure 88:Database Implementation 4

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0100 seconds.)

`SELECT * FROM `sections``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

`id name course_id position created_at updated_at`

Figure 89:Database Implementation 5

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0001 seconds.)

`SELECT * FROM `shows``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

`id student_id quiz_id quiz_score achieved_score gpt_result created_at updated_at`

Figure 90:Database Implementation 6

Show all | Number of rows: 25 Search this table

Extra options

-T→	id	name	email	password	department_id	year_id	textClic
Edit Copy Delete	1	{"ar":"\u0645\u062du0645\u062f","en":"mohammd"}	mohammed@gmail.com	\$2y\$10\$QT3lw3WqahFhcXFV/NXIOEsHPXakJBpHPNfu4fNRm...	1	1	

Figure 91:Database Implementation 7

`SELECT * FROM `student_course``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

`student_id course_id checked_items_counter created_at updated_at`

Figure 92:Database Implementation 8

`SELECT * FROM `student_item``

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

`student_id item_id checked`

Figure 93:Database Implementation 9

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>										
	T		id		email		password		Name		Joining Date		Address		created_at	updated_at
1	<input type="checkbox"/>	adnan@gmail.com	\$2y\$10\$EnxU9iFSqqJyAnw9s0FsegmQmxJlAF73CUKCVk1est... {"ar": "\u0639\u062f\u0646\u0627\u0646 \u0633\u0627", "en": "Mohanned"}	2020-03-12	Amman	2024-05-10 15:10:14	2024-05-10 15:10:14									

Figure 94:Database Implementation 10

Extra options

	<input type="checkbox"/>															
	T		id		name		email		email_verified_at		password		remember_token		created_at	updated_at
1	<input type="checkbox"/>	2024-05-10 15:10:14	2024-05-10 15:10:14													

Figure 95:Database Implementation 11

Extra options

	<input type="checkbox"/>														
	T		id		Name								created_at	updated_at	
1	<input type="checkbox"/>	2024-05-10 15:10:14	2024-05-10 15:10:14												
2	<input type="checkbox"/>	2024-05-10 15:10:14	2024-05-10 15:10:14												
3	<input type="checkbox"/>	2024-05-10 15:10:14	2024-05-10 15:10:14												
4	<input type="checkbox"/>	2024-05-10 15:10:14	2024-05-10 15:10:14												

Figure 96:Database Implementation 12

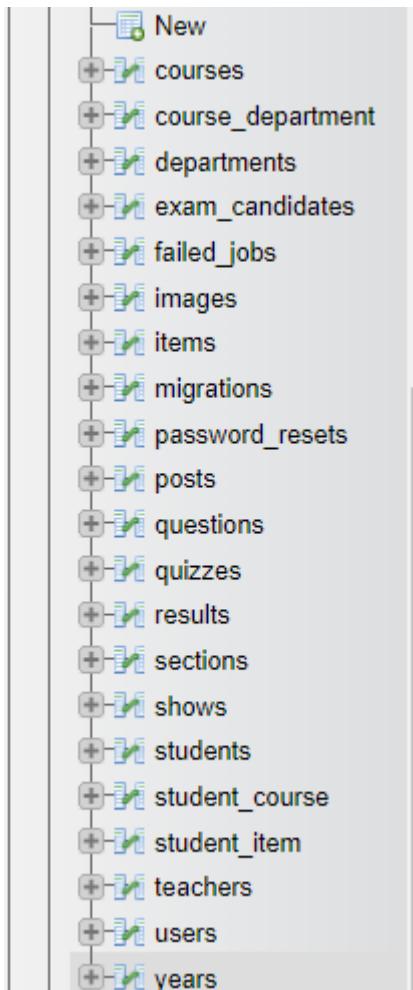


Figure 97:Database Implementation 13

	<input type="checkbox"/>																		
			<input type="checkbox"/>																
			<input type="checkbox"/>																
			<input type="checkbox"/>																
	<input type="checkbox"/>																		

Figure 98:Database Implementation 14

	<input type="checkbox"/>																		
			<input type="checkbox"/>																
			<input type="checkbox"/>																
			<input type="checkbox"/>																
	<input type="checkbox"/>																		

Figure 99:Database Implementation 15

	<input type="checkbox"/>																		
			<input type="checkbox"/>																
			<input type="checkbox"/>																
			<input type="checkbox"/>																
	<input type="checkbox"/>																		

Figure 100:Database Implementation 16

<pre>SELECT * FROM `exam_candidates`</pre>
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
id student_id quiz_id created_at updated_at
Query results operations

Figure 101:Database Implementation 17

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	<input type="checkbox"/> id	<input type="checkbox"/> filename	<input type="checkbox"/> created_at	<input type="checkbox"/> updated_at
	<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	1715343136java.png	2024-05-10 15:12:17	2024-05-10 15:12:17
	<input type="checkbox"/>	<input type="checkbox"/> Check all	With selected:	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	<input type="button" value="Export"/>	

Figure 102:Database Implementation 18

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

```
SELECT * FROM `items`
```

<input type="checkbox"/> id	<input type="checkbox"/> name	<input type="checkbox"/> file_name	<input type="checkbox"/> position	<input type="checkbox"/> section_id	<input type="checkbox"/> wysiwyg_id	<input type="checkbox"/> quiz_id	<input type="checkbox"/> created_at	<input type="checkbox"/> updated_at
-----------------------------	-------------------------------	------------------------------------	-----------------------------------	-------------------------------------	-------------------------------------	----------------------------------	-------------------------------------	-------------------------------------

Figure 103:Database Implementation 19

5.2 Graphical user interface implementation

```
<div class="section-field mb-20">
    <label class="mb-10" for="Password"> {{ trans('login_page.password') }} </label>
    <input id="password" type="password"
        class="form-control @error('password') is-invalid @enderror" name="password"
        autocomplete="current-password"
        @error('password')
            <span class="invalid-feedback" role="alert">
                <strong>{{ $message }}</strong>
            </span>
        @enderror
    </div>

    <div class="form-group mb-0">
        <button class="button"><span>{{ trans('login_page.Login') }}</span><i class="fa fa-check"></i></button>
    </div>
</form> You, 6 days ago • Uncommitted changes

<form method="POST" action="{{ route('forgetPassword') }}" style="display:inline;">
    @csrf
    <input id="user_email" type="hidden" name="user_email" value="{{ old('email') }}>
    <input type="hidden" name="user_type" value="{{ $type }}>
    <button type="submit" class="btn btn-link float-right">
        {{ trans('login_page.Forgot Your Password') }}
    </button>
</form>
</div>
@if ($type == 'student')
<div class="mt-3">
    <p>{{ trans('login_page.Dont have an account?') }} <a class="text-success" href="{{ route('register') }}>{{ trans('login_page.Register here') }}</a></p>
</div>
@endif
@if ($session()->has('msg'))
<br>
<div id="message" style="color:{{ $isError == 2 ? 'red' : 'green' }}">
    {{ $session()->get('msg') }}
</div>
@endif
</div>
```

Figure 104:Graphical user interface implementation 1

```

@section('content')
<!-- row -->


<!-- Filter input field -->
    <div class="row mb-3">
        <div class="col-auto">
            <label for="filterTitle" class="col-form-label">{{trans('student_courses.Filter by Course Title')}}:</label>
        </div>
        <div class="col">
            <input type="text" class="form-control" id="filterTitle">
        </div>
    </div>

    <div class="col-md-4 mb-3">
        <label for="filterDepartment" class="form-label">{{trans('student_courses.Filter by Department')}}:</label>
        <select class="form-select" id="filterDepartment">
            <option value="">{{trans('student_courses.All Departments')}}</option>
            @foreach($departments as $department)
                <option value="{{ $department->id }}>{{ $department->Name }}</option>
            @endforeach
        </select>
    </div>

    <div class="col-md-4 mb-3">
        <label for="filterYear" class="form-label">{{trans('student_courses.Filter by Year')}}:</label>
        <select class="form-select" id="filterYear">
            <option value="">{{trans('student_courses.All Years')}}</option>
            @foreach($years as $year)
                <option value="{{ $year->id }}>{{ $year->Name }}</option>
            @endforeach
        </select>
    </div>

    <div class="col-md-12 mb-30">
        <div class="card card-statistics h-100">
            @php
                $studentId = auth()->user()->id;
                $counter=0;
            @endphp
            <div id="cardContainer" class="card-body" style="display: flex; flex-wrap: wrap;">
                @foreach($courses as $key => $course)

                    @if($course->published=="Active")
                    @php
                        $studentCourse = DB::table('student_course')
                            ->where('student_id', $studentId)
                            ->where('course_id', $course->id)
                            ->first();
                    @endphp
                    $checkedItemsCounter = $studentCourse ? $studentCourse->checked_items_counter : 0;
                    $totalItems = $course->item_counter;
                    $percent = ($totalItems > 0) ? ($checkedItemsCounter / $totalItems) * 100 : 0;
                    @endphp
                @endif
            @endforeach
        </div>
    </div>


```

Figure 105: Graphical user interface implementation 2

```

<div class="col-md-12 mb-30">
    <div class="card card-statistics h-100">
        @php
            $studentId = auth()->user()->id;
            $counter=0;
        @endphp
        <div id="cardContainer" class="card-body" style="display: flex; flex-wrap: wrap;">
            @foreach($courses as $key => $course)
                @if($course->published=="Active")
                    @php
                        $studentCourse = DB::table('student_course')
                            ->where('student_id', $studentId)
                            ->where('course_id', $course->id)
                            ->first();
                    @endphp
                    $checkedItemsCounter = $studentCourse ? $studentCourse->checked_items_counter : 0;
                    $totalItems = $course->item_counter;
                    $percent = ($totalItems > 0) ? ($checkedItemsCounter / $totalItems) * 100 : 0;
                    @endphp
                    @if($percent==100)
                        @php
                            $counter++;
                        @endphp
                    <div class="card" style="width: 18rem; margin-right: 10px; margin-bottom: 10px;" data-department="{{ implode(',', $course->Department->pluck('id')->toArray()) }}" data-year="{{ $course->year_id }}"
                        
                        <div class="card-body">
                            @php
                                $title_length = mb_strlen($course->title);
                                $description_length = mb_strlen($course->description);
                            @endphp
                            @if ($title_length > 18)
                                <h5 class="card-title">
                                    <span id="titleShort{{$key}}">{{ Illuminate\Support\Str::limit($course->title, 18)}}</span>
                                    <span id="titleFull{{$key}}" style="display: none;">{{ $course->title }}</span>
                                    <a href="#" onclick="toggleTitle({{$key}});">{{trans('student_courses.Show More')}}{{ $course->description }}</span>
                                <a href="#" onclick="toggleDescription({{$key}});"
                                    @if ($description_length > 30)

```

Figure 106: Graphical user interface implementation 3

```

@extends('layouts.master')          You, 2 months ago • Initial commit
@section('css')
    @toastr_css

@section('title')
{{trans('admin_student.add_student')}}
@stop
@endsection
@section('page-header')
<!-- breadcrumb -->
@section('PageTitle')
{{trans('admin_student.add_student')}}
@stop
<!-- breadcrumb -->
@endsection
@section('content')
<!-- row -->


@if ($errors->any())


@foreach ($errors->all() as $error)
- {{ $error }}

@endforeach


@endif

<form method="post" action="{{ route('Students.store') }}" autocomplete="off"
enctype="multipart/form-data">
@csrf
<h6 style="font-family: 'Cairo'>{{trans('admin_student.personal_information')}}</h6>

<br>


<label>{{trans('admin_student.name_ar')}} <span
class="text-danger"></span></label>
<input type="text" name="name_ar" class="form-control">



<div class="form-group">
<label>{{trans('admin_student.name_en')}} <span
class="text-danger"></span></label>
<input class="form-control" name="name_en" type="text">
</div>


```

Figure 107:Graphical user interface implementation 4

```

<div class="col-md-6">
    <div class="form-group">
        <label>{{trans('admin_student.name_en')}} <span
            class="text-danger"></span></label>
        <input class="form-control" name="name_en" type="text">
    </div>
</div>

<div class="row">
    <div class="col-md-6">
        <div class="form-group">
            <label>{{trans('admin_student.email')}} </label>
            <input type="email" name="email" class="form-control">
        </div>
    </div>

    <div class="col-md-6">
        <div class="form-group">
            <label>{{trans('admin_student.password')}}</label>
            <input type="password" name="password" class="form-control">
        </div>
    </div>

    <div class="col-md-3">
        <div class="form-group">
            <label for="department">{{trans('admin_student.department')}} <span
                class="text-danger"></span></label>
            <select class="custom-select mr-sm-2" name="department_id">
                <option selected disabled>{{ trans('Parent_trans.Choose') }}...</option>
                @foreach ($departments as $department)
                    <option value="{{ $department->id }}>{{ $department->Name }}</option>
                @endforeach
            </select>
        </div>
    </div>

    <div class="col-md-3">
        <div class="form-group">
            <label for="year">{{trans('admin_student.year')}} <span
                class="text-danger"></span></label>
            <select class="custom-select mr-sm-2" name="year_id">
                <option selected disabled>{{ trans('Parent_trans.Choose') }}...</option>
                @foreach ($years as $year)
                    <option value="{{ $year->id }}>{{ $year->Name }}</option>
                @endforeach
            </select>
        </div>
    </div>

```

Figure 108:Graphical user interface implementation 5

```

You, 6 days ago | 1 author (You)
<!DOCTYPE html>
<html lang="en" dir="rtl">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="keywords" content="HTML5 Template" />
    <meta name="description" content="Webmin - Bootstrap 4" />
    <meta name="author" content="potenzaglobalsolutions.co" />
    <meta name="viewport" content="width=device-width, ini
    <title>{{ trans('login_page.EduAi') }} </title>

    <!-- Favicon -->
    <link rel="shortcut icon" href="images/favicon.ico" /> You, 2 months ago • Initial commit

    <!-- Font -->
    <link rel="stylesheet"
        href="https://fonts.googleapis.com/css?family=Poppins:200,300,300i,400,400i,500,500i,600,600i,700,700i,800,800i,
        <!-- css -->
        <link href="{{ URL::asset('assets/css/rtl.css') }}" rel="stylesheet">
        <link rel="shortcut icon" href="{{ URL::asset('assets/images/fiv.png') }}" type="image/x-icon" />
</head>

<body>
@php
if (!isset($isError)) {
    $isError=0;
}

```

Figure 109: Graphical user interface implementation 6

```

<body>
@endphp
<div class="wrapper">
    <!-- Preloader -->
    <div id="pre-loader">
        
    </div>
    <!-- Preloader -->

    <!-- Login Section -->
    <section class="height-100vh d-flex align-items-center page-section-ptb login"
        style="background-image: url('{{ asset('assets/images/bg2.png') }}');">
        <div class="container">
            <div class="row justify-content-center no-gutters vertical-align">
                <div class="col-lg-4 col-md-6 login-fancy-bg bg">
                    >
                    <div class="login-fancy text-center">
                        <h2 class="text-white mb-4" style="font-size: 40px;">{{ trans('login_page.Welcome') }}<br>{{ trans('login_page.To') }} <br>!{{ trans('login_page.EduAi') }}</h2>
                        <p class="mb-4 text-white" style="font-size: 14px;">{{ trans('login_page.content') }}</p>
                        <ul class="list-unstyled pos-bot pb-3">
                            <li class="list-inline-item mb-2"><a class="text-white mr-4 p-5"
                                href="{{ route('term') }}>{{ trans('login_page.Terms of Use') }}</a></li>
                            <li class="list-inline-item mb-2"><a class="text-white"
                                href="{{ route('privacy') }}>{{ trans('login_page.Privacy Policy') }}</a></li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </section>
</body>

```

Figure 110: Graphical user interface implementation 7

```
<div class="wrapper">
    <section class="height-100vh d-flex align-items-center page-section-ptb login">
        <div class="container">
            </div>
            <div class="col-lg-4 col-md-6 bg-white">
                <div class="login-fancy pb-40 clearfix">
                    @if ($type == 'student')
                        <h3 style="font-family: 'Cairo', sans-serif" class="mb-30">
                            {{ trans('login_page.Student') }} </h3>
                    @elseif($type == 'teacher')
                        <h3 style="font-family: 'Cairo', sans-serif" class="mb-30">
                            {{ trans('login_page.Instructor') }}</h3>
                    @else
                        <h3 style="font-family: 'Cairo', sans-serif" class="mb-30">
                            {{ trans('login_page.Admin') }}</h3>
                    @endif

                    @if (\Session::has('message'))
                        <div class="alert alert-danger">
                            <li>{!! \Session::get('message') !!}</li>
                        </div>
                    @endif

                    <form method="POST" action="{{ route('login') }}" style="display:inline;">
                        @csrf

                        <div class="section-field mb-20">
                            <label class="mb-10" for="name">{{ trans('login_page.Email') }}</label>
                            <input id="email" type="email" name="email" value="{{ old('email') }}">
                        </div>
                </div>
            </div>
        </div>
    </section>
</div>
```

Figure 111: Graphical user interface implementation 8

5.3 Graphical user interface high fidelity prototype

5.3.1 Student interface

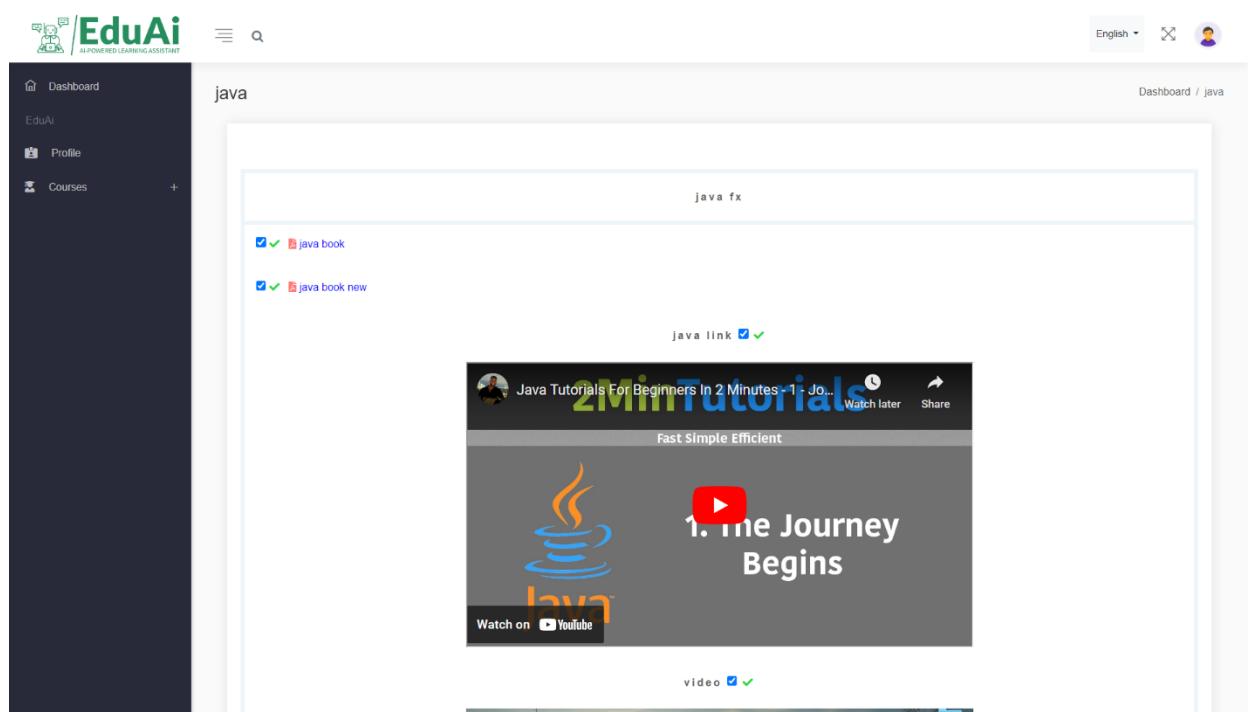


Figure 112: Student GUI 1

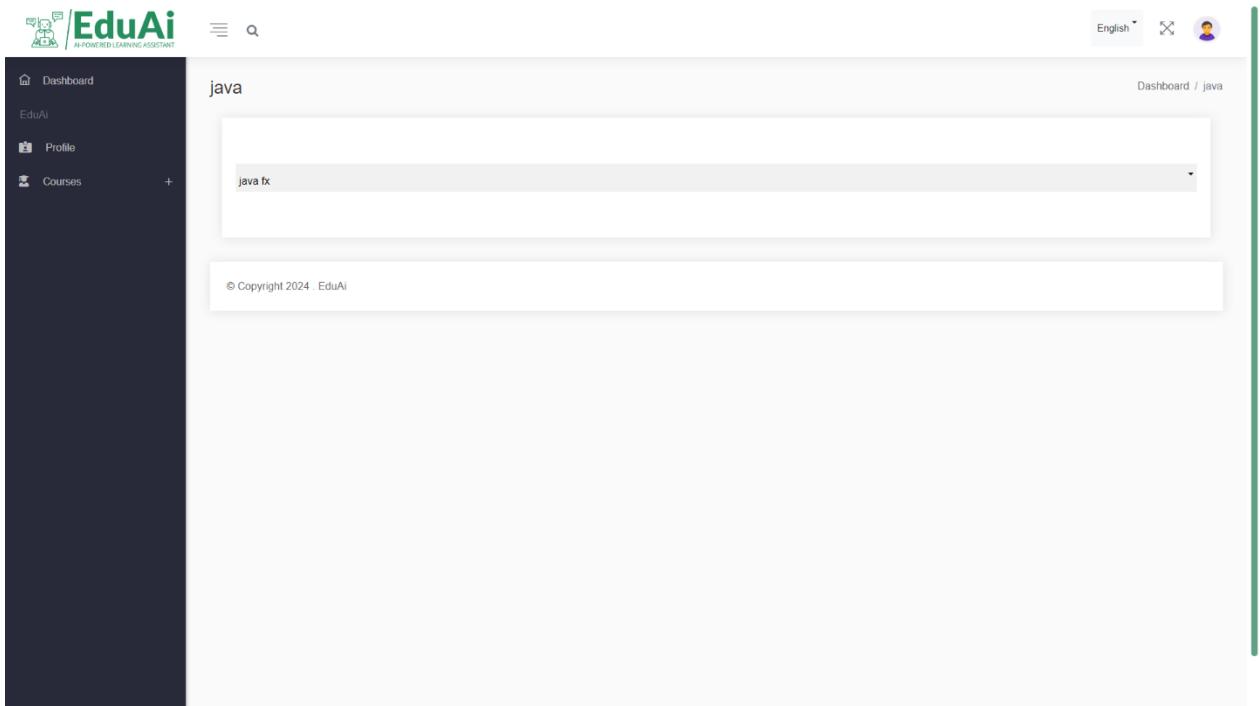


Figure 113: Student GUI 2

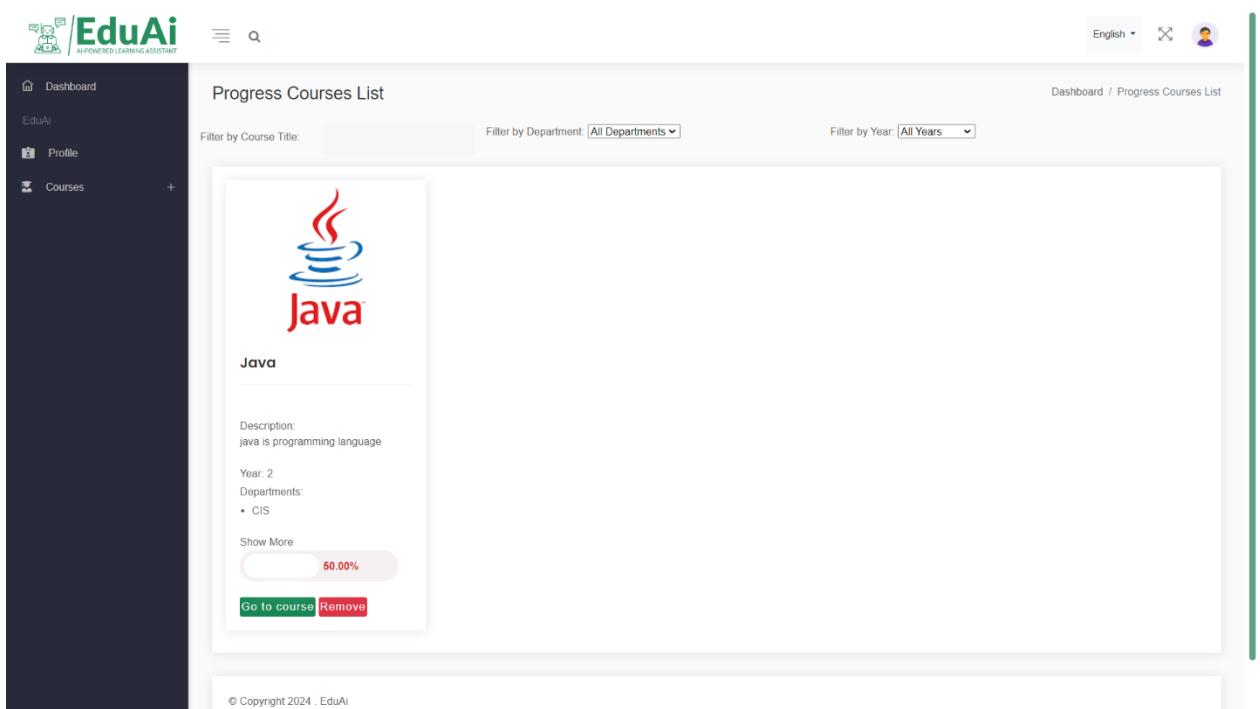


Figure 114: Student GUI 3

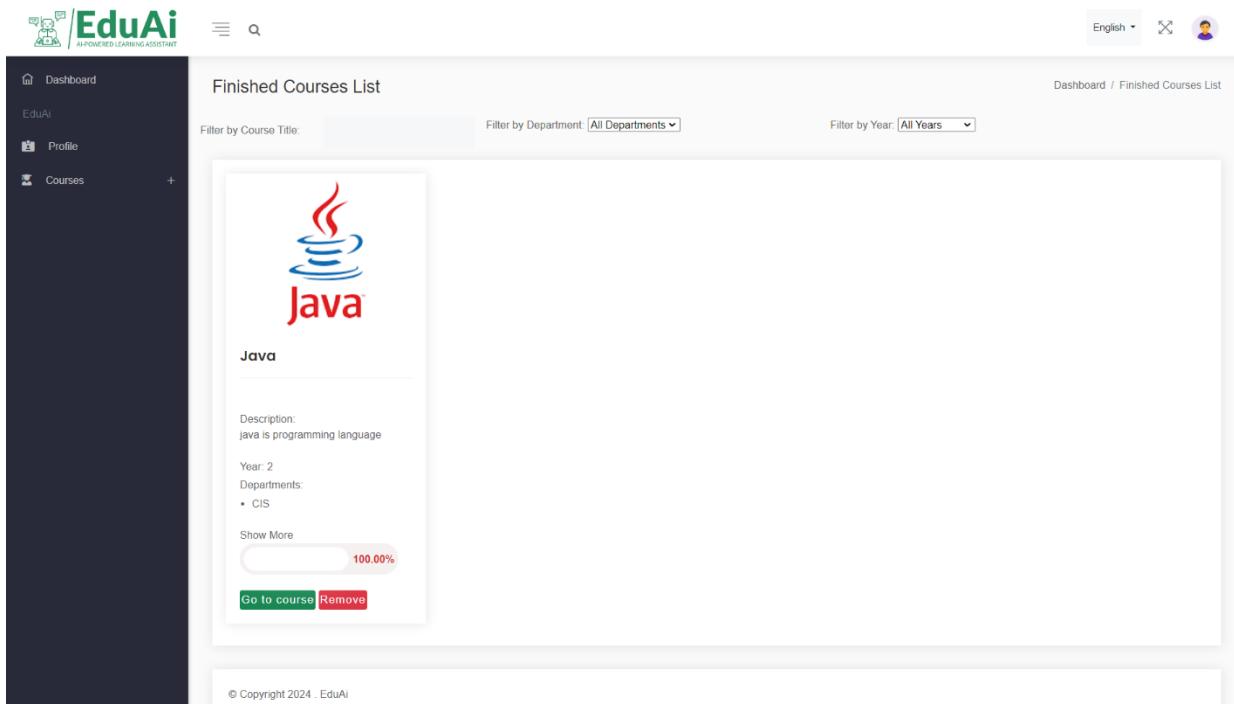


Figure 115: Student GUI 4

5.3.1 Instructor interface

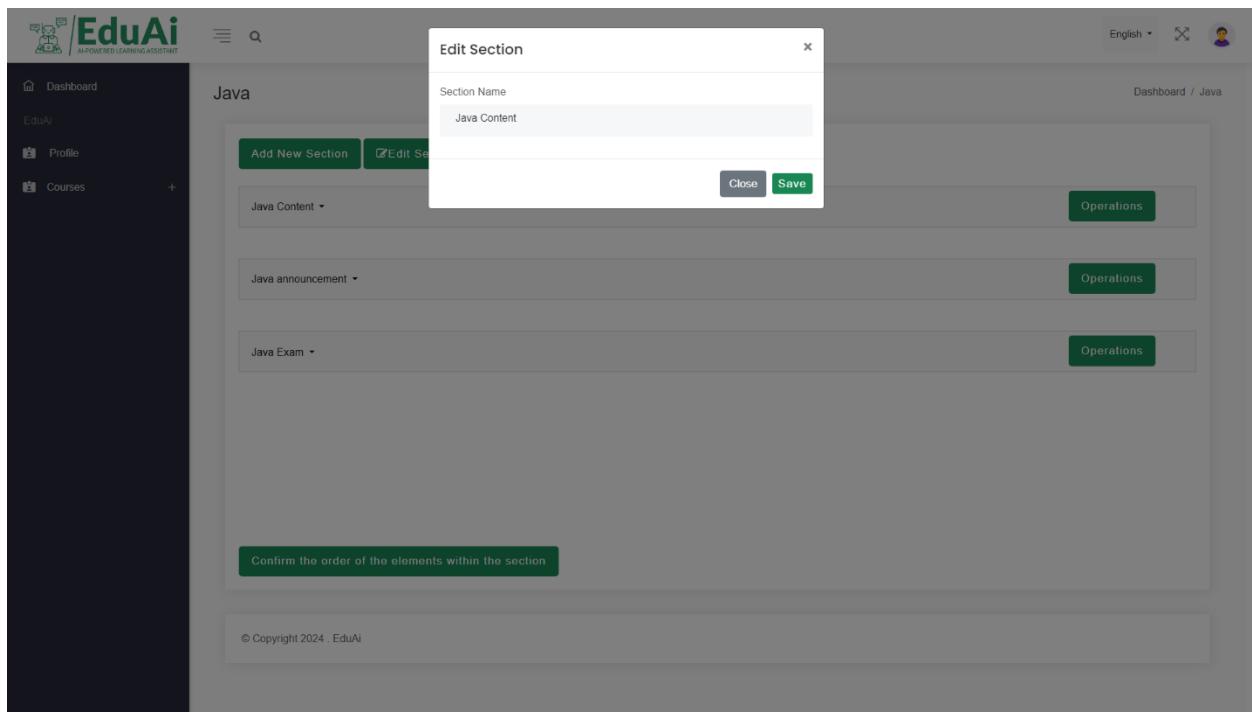


Figure 116: Instructor GUI 1

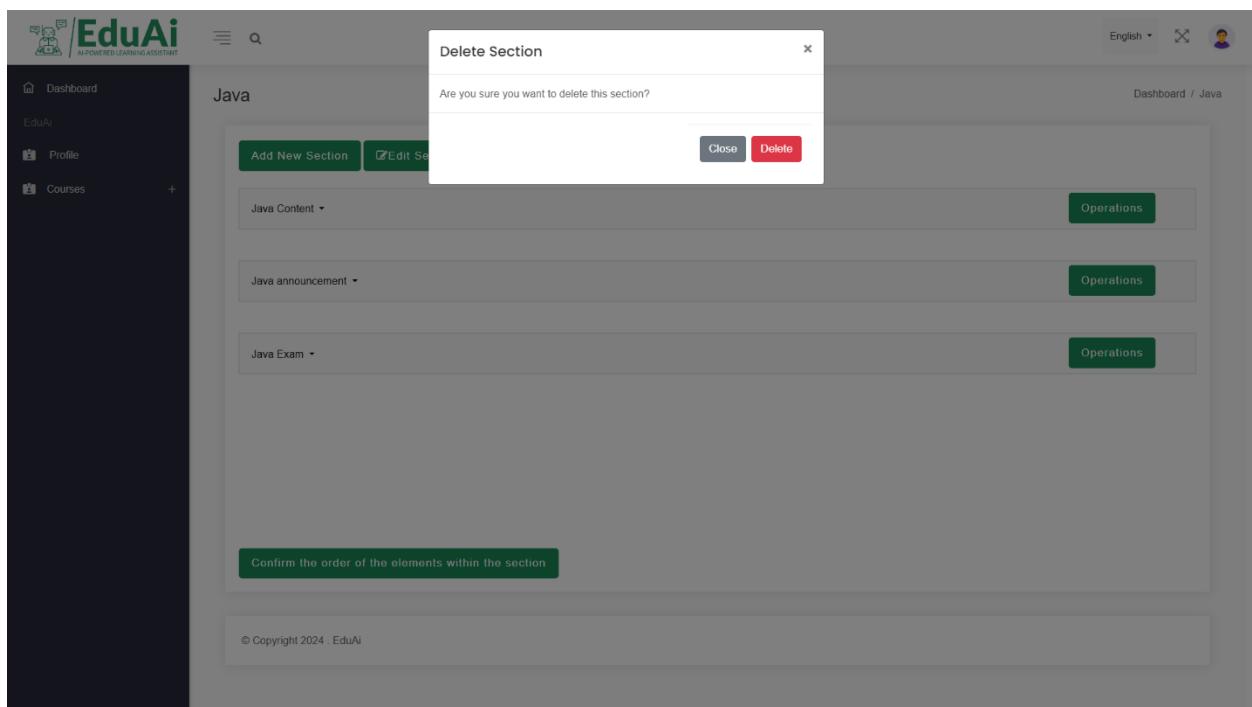


Figure 117: Instructor GUI 2

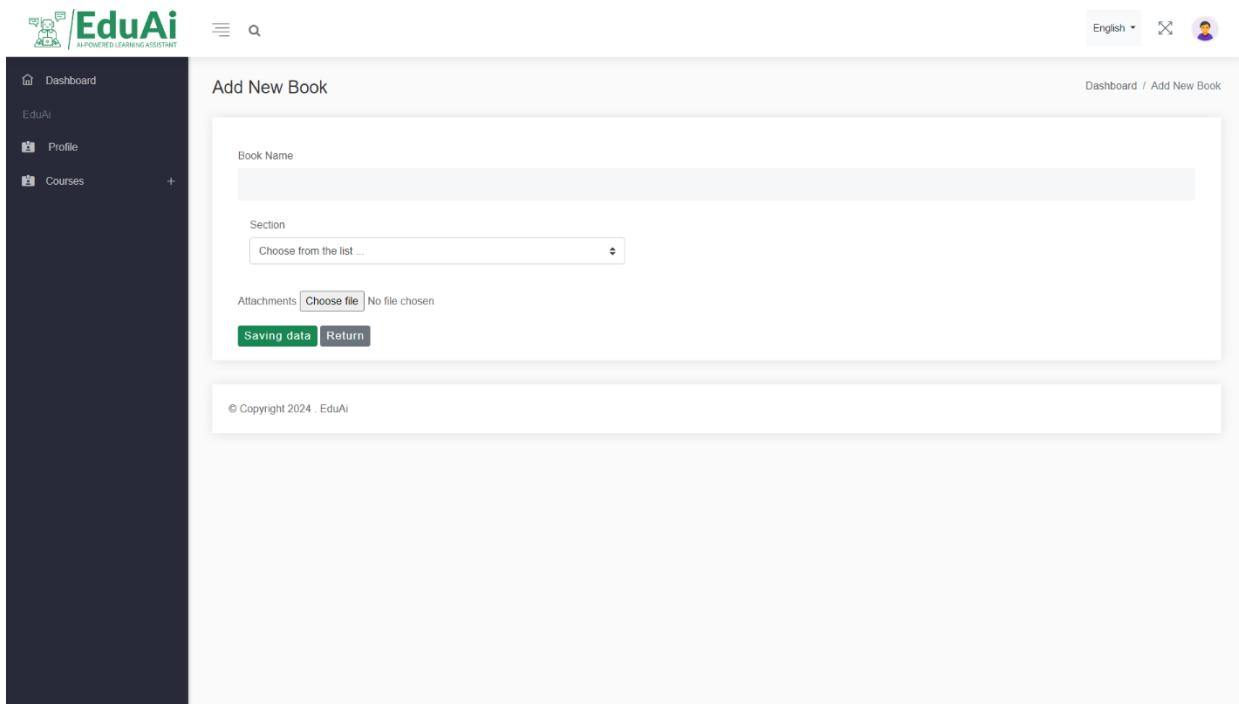


Figure 118: Instructor GUI 3

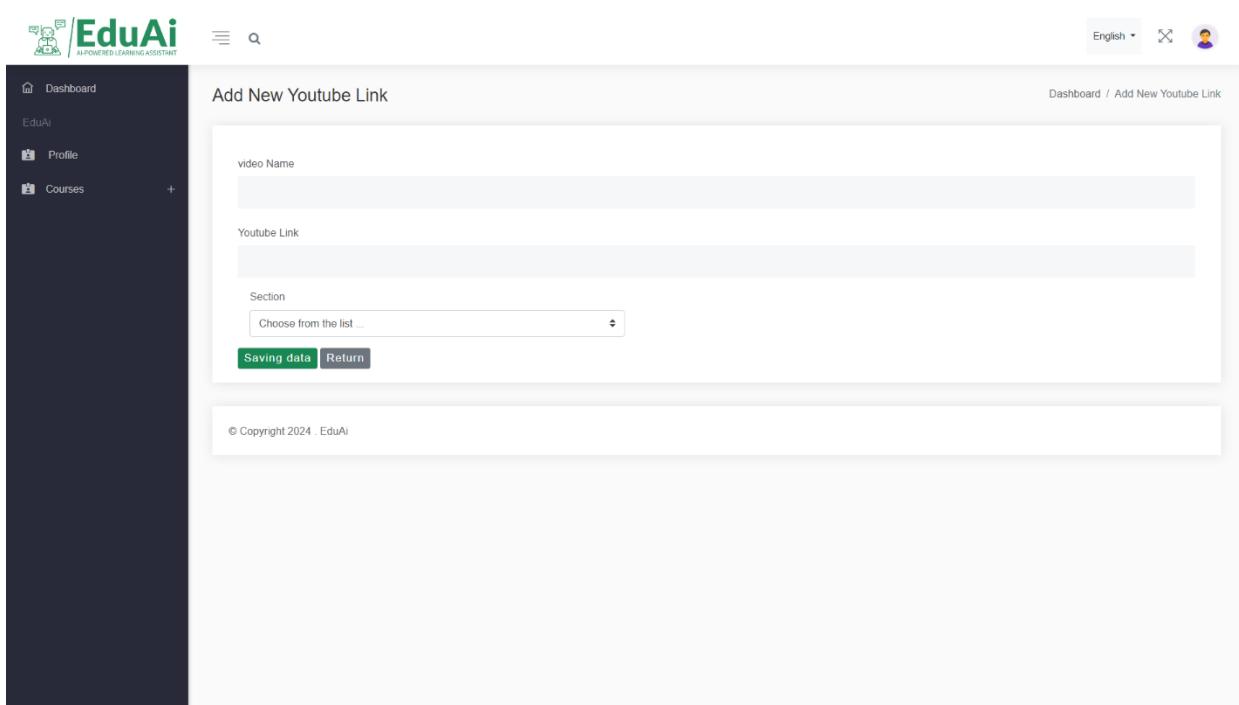


Figure 119: Instructor GUI 4

Add new post

Title

Description

Enter your description...

Section

Choose from the list

Add

Cancel

Figure 120: Instructor GUI 5

The screenshot shows the EduAi Instructor GUI. On the left is a dark sidebar with icons for Dashboard, Profile, and Courses. The main area has a header with the EduAi logo, a search bar, and user settings. The current page is 'Add New Quiz'. It contains fields for 'Quiz Title' and 'Duration in Minute', a 'Section' dropdown set to 'Choose from the list...', and two buttons: 'Add' (green) and 'Return' (grey). At the bottom is a footer with the copyright notice: '© Copyright 2024 EduAi'.

Figure 121: Instructor GUI 6

The screenshot shows the EduAi Instructor GUI. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, Courses, and a plus sign for adding new items. The main content area has a header with a search bar and a user profile icon. Below the header is a breadcrumb navigation: Dashboard / Quiz List. The central part of the screen displays a table titled "Quiz List". The table has three columns: "#", "Title", and "Duration". There is one entry in the table:

#	Title	Duration
1	Java Fundamentals Quiz	60 minutes

Below the table is a "Return" button and a copyright notice: © Copyright 2024 EduAi.

Figure 122: Instructor GUI 7

The screenshot shows the EduAi Instructor GUI. The sidebar on the left includes: Dashboard, EduAi, Profile, Courses, and a plus sign for adding new items. The main content area features a search bar and a "Java Content" dropdown menu. A green "Operations" button is located in the top right corner of the content area. The content area displays a "Java Book" section with a "Download" button, an "Update" button, and a "Delete" button. Below it is a "Java Video" section showing a thumbnail for a video titled "Java Tutorials For Beginners In 2 Minutes - 1 - Java Fundamentals". The thumbnail includes a play button, a timer (0:00), and options to "Watch later" and "Share". The video is described as "Fast Simple Efficient". The video thumbnail has a "Delete" button at the bottom. At the bottom of the content area are two more sections: "Java announcement" and "Java Exam", each with its own "Operations" button.

Figure 123: Instructor GUI 8

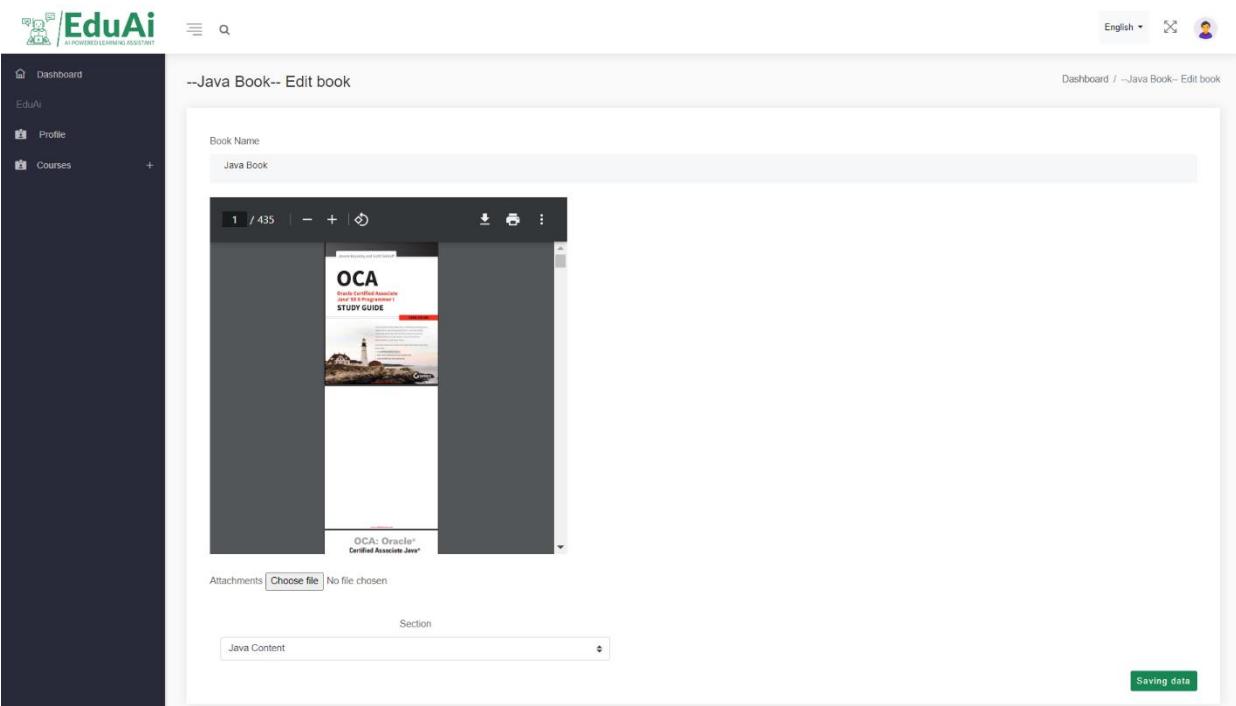


Figure 124: Instructor GUI 9

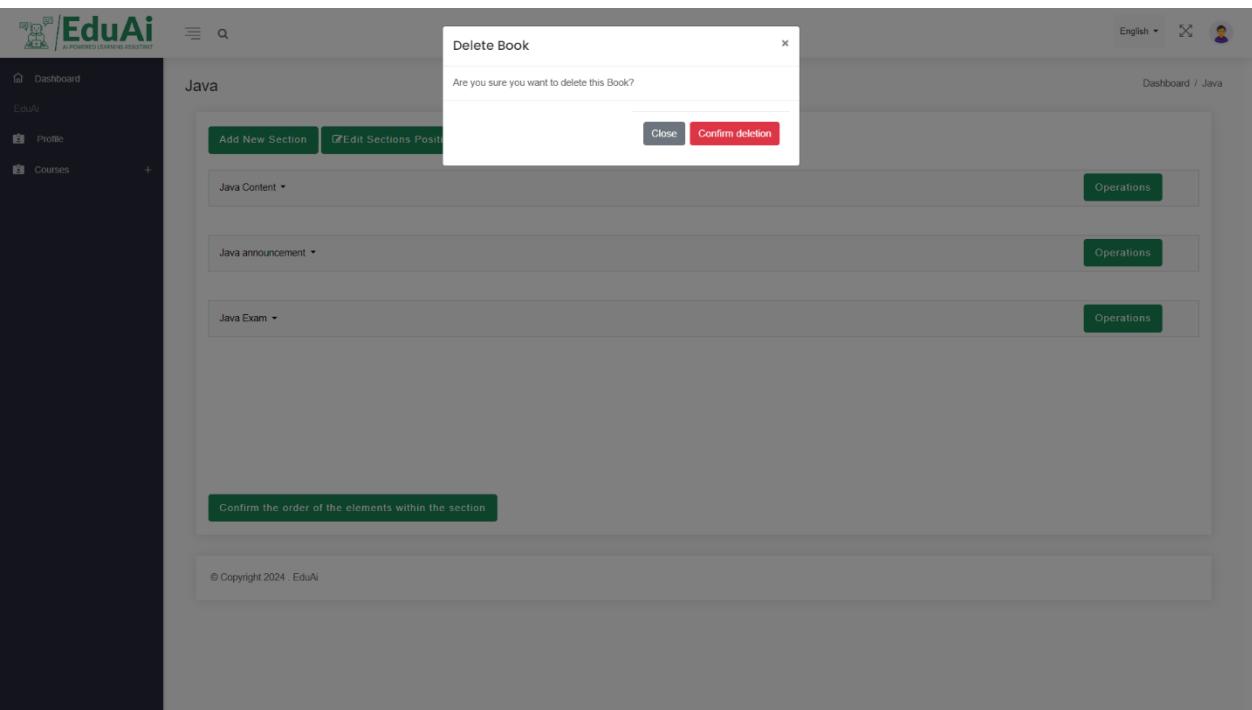


Figure 125: Instructor GUI 10

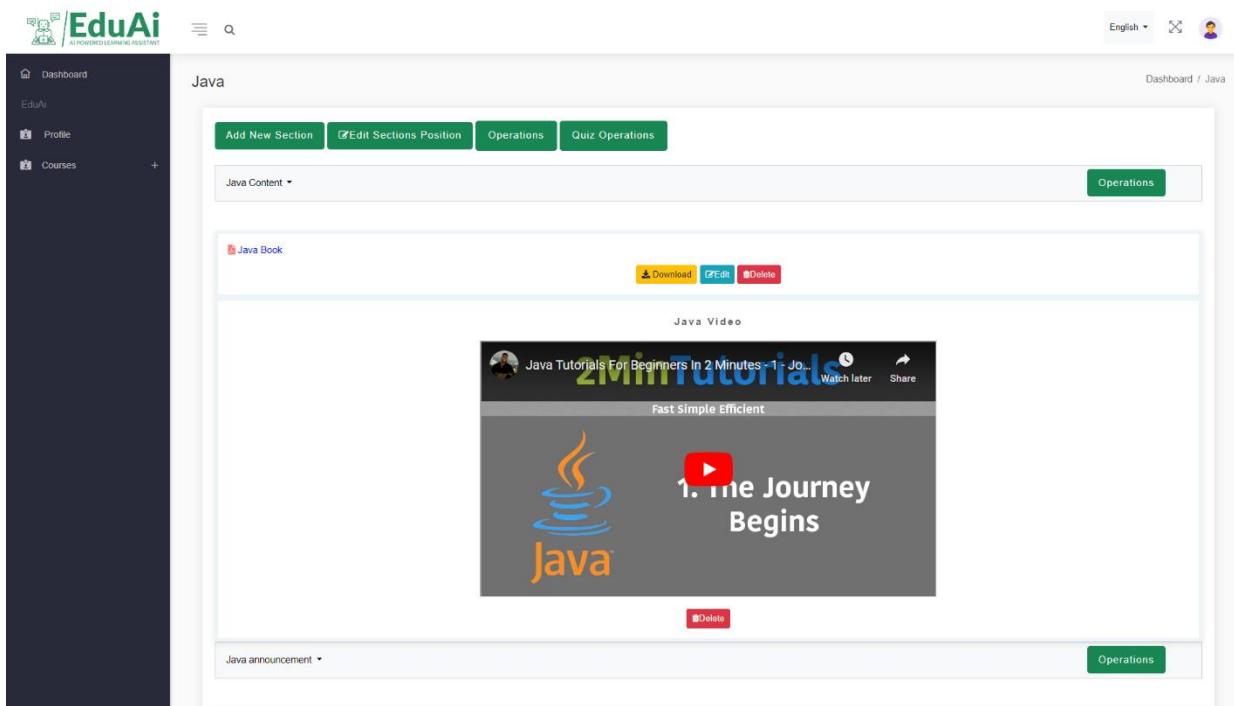


Figure 126: Instructor GUI 11

Edit Java Exam Details-Post

Title
Java Exam Details

Description

There is no makeup exam!!!!

Date: 5/5/2024

Location: online platform

Topics:

- Java concepts
- Java syntax
- Best practices

Section
Java announcement

Update **Cancel**

Figure 127: Instructor GUI 12

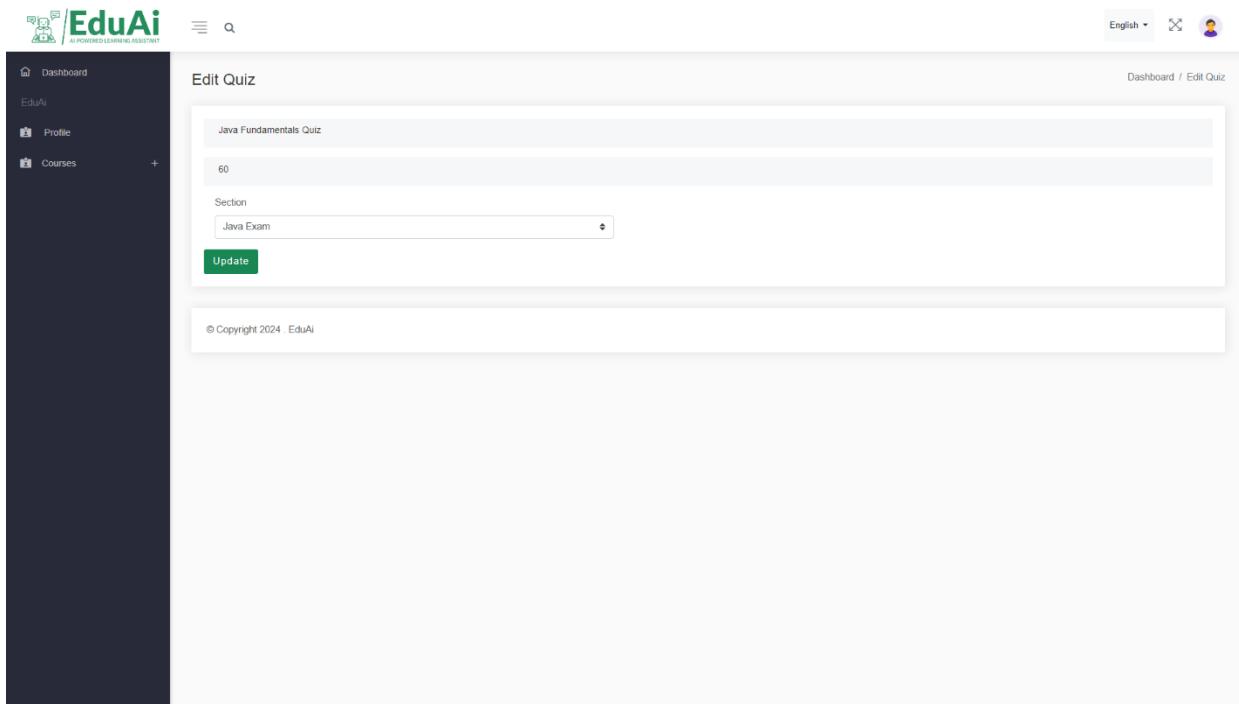


Figure 128: Instructor GUI 13

The screenshot shows the 'Add Question for the Quiz' interface. The title is 'Java Fundamentals Quiz'. The main area has sections for 'Question', 'Option A', 'Option B', 'Option C', 'Option D', and a dropdown for 'Select Correct Option'. Below this is a table with 5 rows of questions and their options. The table includes columns for '#', 'Question', 'A', 'B', 'C', 'D', 'Correct', and 'Operations'. Each row has edit and delete buttons. The footer shows the copyright notice '© Copyright 2024 EduAi'.

#	Question	A	B	C	D	Correct	Operations
1	What is the correct syntax for declaring a variable in Java?	variable x;	x = 5;	int x;	var x;	option_c	<button>Edit</button> <button>Delete</button>
2	Which of the following is NOT a primitive data type in Java?	boolean	string	float	int	option_b	<button>Edit</button> <button>Delete</button>
3	What does the "public static void main(String[] args)" method indicate in a Java program?	It is a constructor method	It is used to declare variables	It is a method to print output	It is the starting point of the program	option_d	<button>Edit</button> <button>Delete</button>
4	What does the "this" keyword refer to in Java?	All class objects	Next class object	Previous class object	Current class object	option_d	<button>Edit</button> <button>Delete</button>
5	What is the purpose of the "static" keyword in Java?	To prevent inheritance	To allow access to a method or variable without creating an instance of the class	To declare a variable as constant	To make a method non-static	option_b	<button>Edit</button> <button>Delete</button>

Figure 129: Instructor GUI 14

What is the correct syntax for declaring a variable in Java?

A: variable x;
B: x = 5;
C: int x;
D: var x;

#	Question	A	B	C	D	Correct
1	What is the correct syntax for declaring a variable in Java?	variable x;	x = 5;	int x;	var x;	option_c
2	Which of the following is NOT a primitive data type in Java?	boolean	string	float	int	option_b
3	What does the "public static void main(String[] args)" method indicate in a Java program?	It is a constructor method	It is used to declare variables	It is a method to print output	It is the starting point of the program	option_d
4	What does the "this" keyword refer to in Java?	All class objects	Next class object	Previous class object	Current class object	option_d
5	What is the purpose of the "static" keyword in Java?	To prevent inheritance	To allow access to a method or variable without creating an instance of the class	To declare a variable as constant	To make a method non-static	option_b

© Copyright 2024 EduAI

Figure 130: Instructor GUI 15

Are you sure you want to delete this Question?

Delete **Cancel**

#	Question	A	B	C	D	Correct	Operations
1	What is the correct syntax for declaring a variable in Java?	variable x;	x = 5;	int x;	var x;	option_c	Edit Delete
2	Which of the following is NOT a primitive data type in Java?	boolean	string	float	int	option_b	Edit Delete
3	What does the "public static void main(String[] args)" method indicate in a Java program?	It is a constructor method	It is used to declare variables	It is a method to print output	It is the starting point of the program	option_d	Edit Delete
4	What does the "this" keyword refer to in Java?	All class objects	Next class object	Previous class object	Current class object	option_d	Edit Delete
5	What is the purpose of the "static" keyword in Java?	To prevent inheritance	To allow access to a method or variable without creating an instance of the class	To declare a variable as constant	To make a method non-static	option_b	Edit Delete

Figure 131: Instructor GUI 16

Dashboard / Edit Question for the Quiz: Java Fundamentals Quiz

What is the correct syntax for declaring a variable in Java?

variable x;

x = 5;

int x;

var x;

C

Update

#	Question	A	B	C	D	Correct
1	What is the correct syntax for declaring a variable in Java?	variable x;	x = 5;	int x;	var x;	option_c
2	Which of the following is NOT a primitive data type in Java?	boolean	string	float	int	option_b
3	What does the "public static void main(String[] args)" method indicate in a Java program?	It is a constructor method	It is used to declare variables	It is a method to print output	It is the starting point of the program	option_d
4	What does the "this" keyword refer to in Java?	All class objects	Next class object	Previous class object	Current class object	option_d
5	What is the purpose of the "static" keyword in Java?	To prevent inheritance	To allow access to a method or variable without creating an instance of the class	To declare a variable as constant	To make a method non-static	option_b

© Copyright 2024 . EduAi

Figure 132: Instructor GUI 17

Welcome Adnan Samer

Welcome, Instructor! Need assistance or have any questions? Click [here](#) to chat with our helpful chatbot.

NO DATA!!

Latest On System Operations

#	Instructor Name	Date of hiring	Date added
1	Adnan Samer	2020-03-12	2024-05-04 10:52:34

Figure 133: Instructor GUI 18

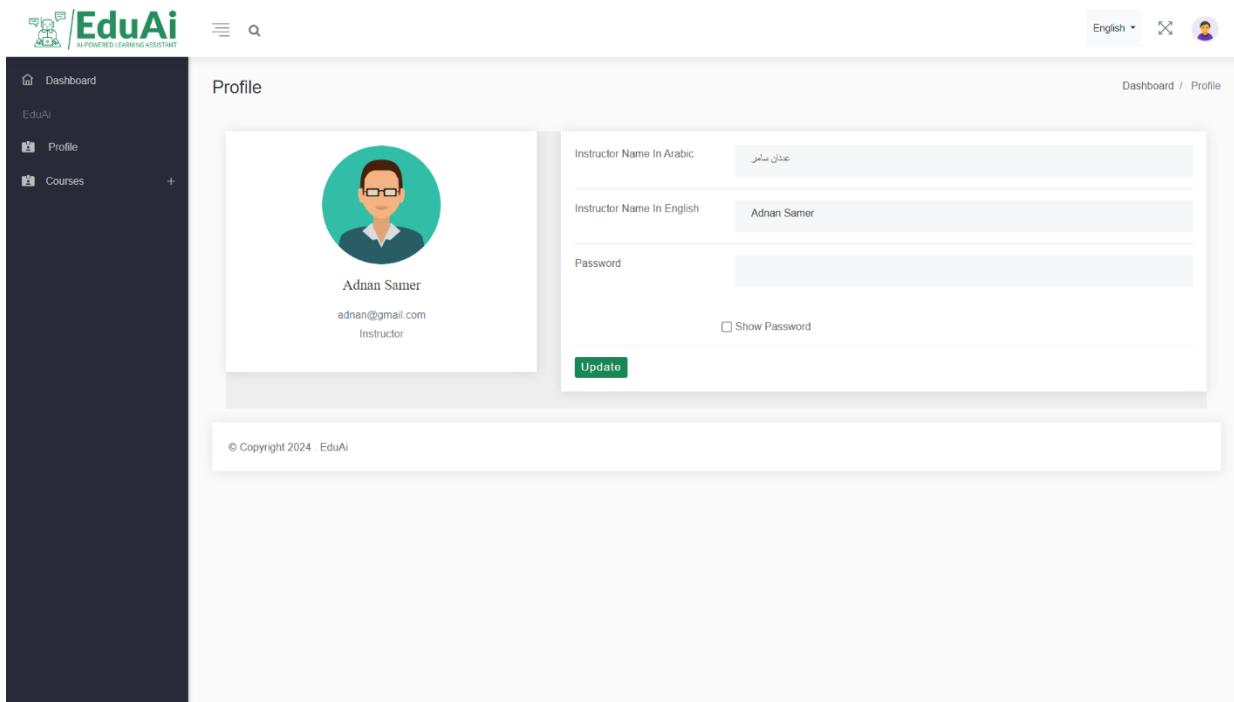


Figure 134: Instructor GUI 19

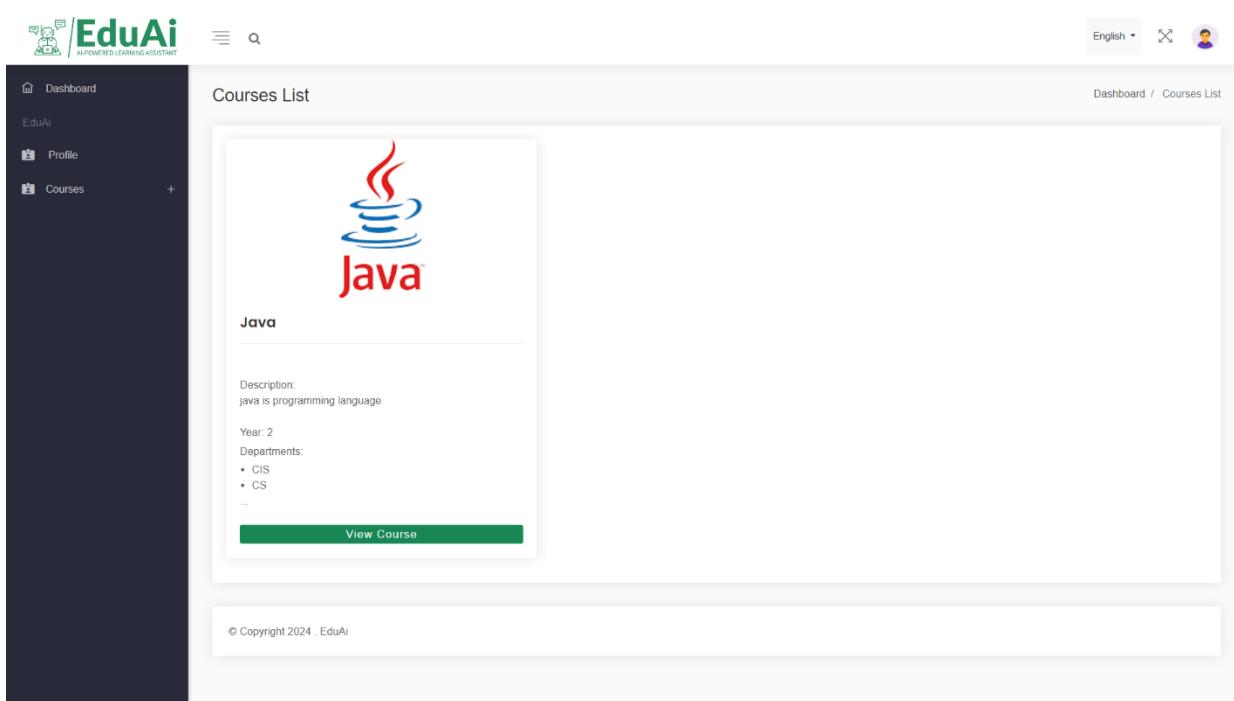


Figure 135: Instructor GUI 20

The screenshot shows the 'Courses List' page of the EduAi application. The left sidebar has a dark theme with icons for Dashboard, Profile, and Courses. The main area has a light theme with a header bar for English language and user profile. The title 'Courses List' is at the top, followed by a green button 'Add New Course'. A search bar and a dropdown for 'Show 50 entries' are present. A table lists one course: 'Java' with the description 'java is programming language', a course image icon, and status 'Active'. Action buttons 'Edit' and 'Delete' are available. Navigation buttons 'Previous', '1', and 'Next' are at the bottom.

Figure 136: Instructor GUI 22

The screenshot shows the 'Add New Course' page. The left sidebar is identical to Figure 136. The main area has a light theme with a header bar for English language and user profile. The title 'Add New Course' is at the top. The form fields include 'Title' (input field), 'Description' (input field), 'Course Image' (input field with 'Choose file' button and 'No file chosen' message), 'Department' (dropdown menu with options 'CIS', 'CS', 'AI', etc., showing 'Choose from the list...'), 'Years' (dropdown menu with 'Choose from the list...'), 'Published' (dropdown menu with 'Active'), and a green 'Add' button at the bottom right. A copyright notice '© Copyright 2024 . EduAi' is at the bottom.

Figure 137: Instructor GUI 23

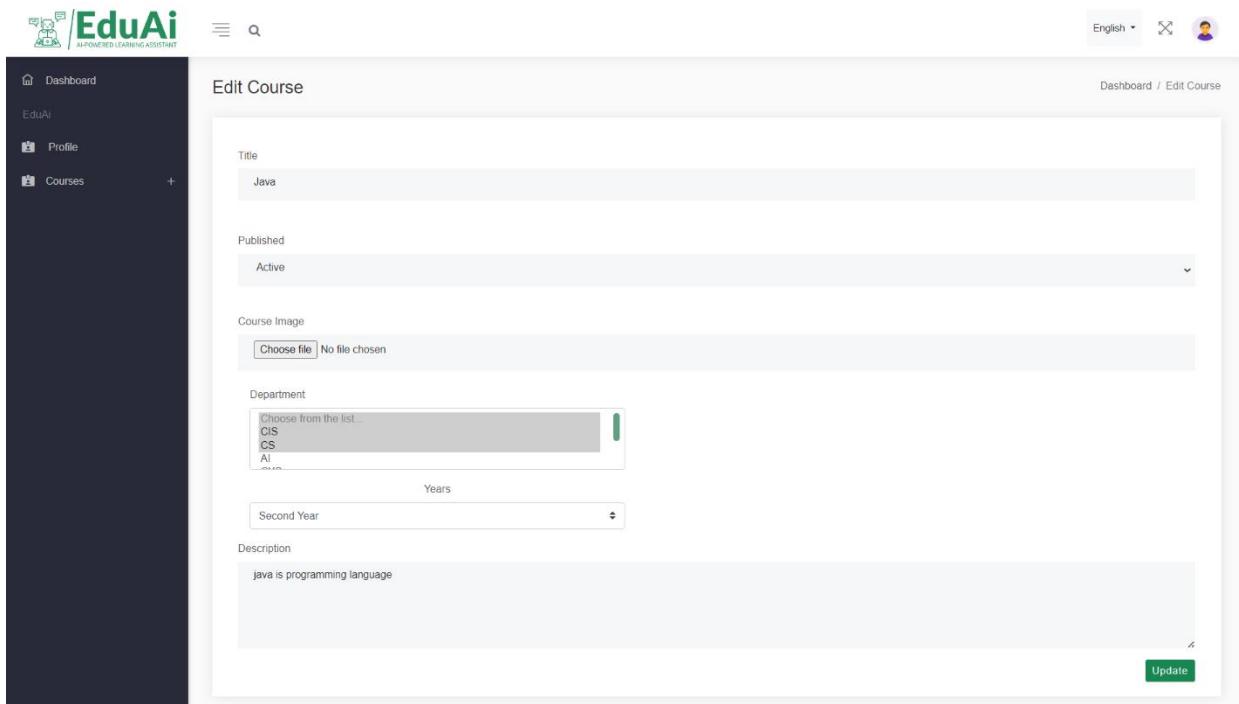


Figure 138: Instructor GUI 24

The screenshot shows the 'Courses List' interface. On the left is a dark sidebar with navigation links: Dashboard, Profile, Courses, and a plus sign. The main area has a light gray header 'Courses List' and a breadcrumb 'Dashboard / Courses List'. A modal dialog titled 'Delete Course' asks 'Are you sure you want to delete this course?'. The dialog has 'Cancel' and 'Delete' buttons. Below the dialog is a table with one entry:

Title	Description	Course Image	Published	Action
Java	java is programming language		Active	<button>Edit</button> <button>Delete</button>

At the bottom, it says 'Showing 1 to 1 of 1 entries' and has 'Previous' and 'Next' buttons. The footer includes a copyright notice: '© Copyright 2024 EduAi'.

Figure 139: Instructor GUI 25

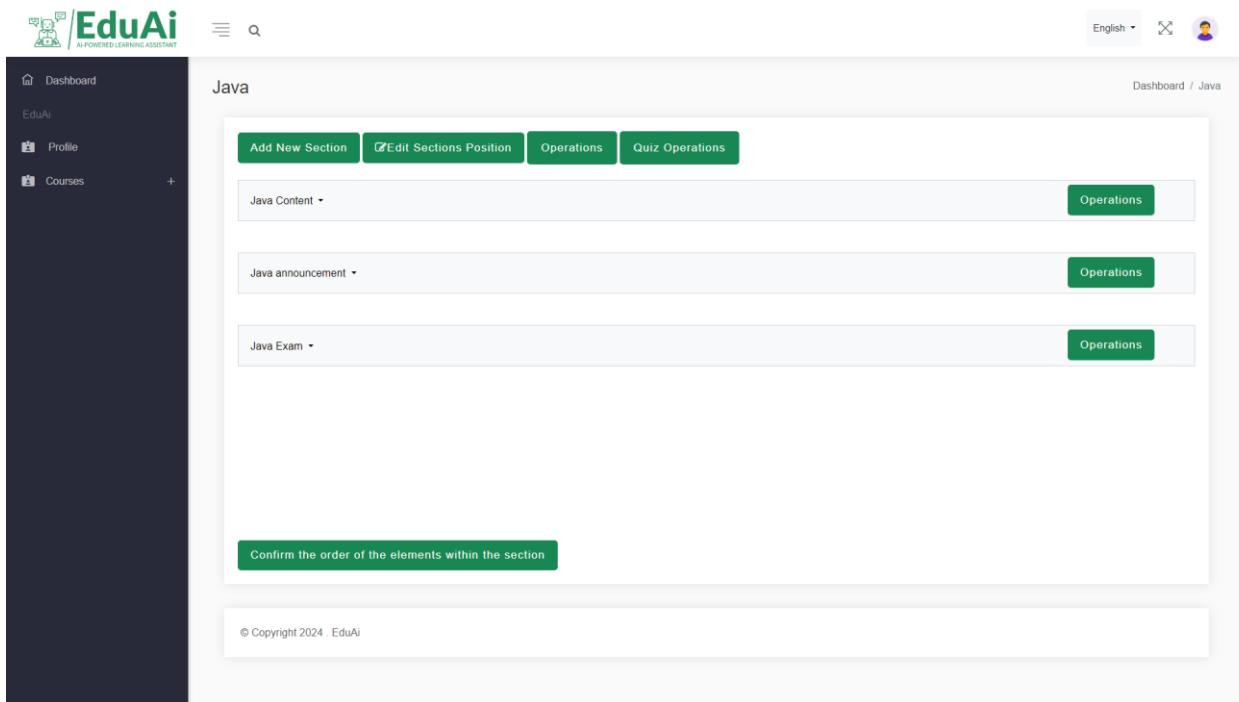


Figure 140: Instructor GUI 26

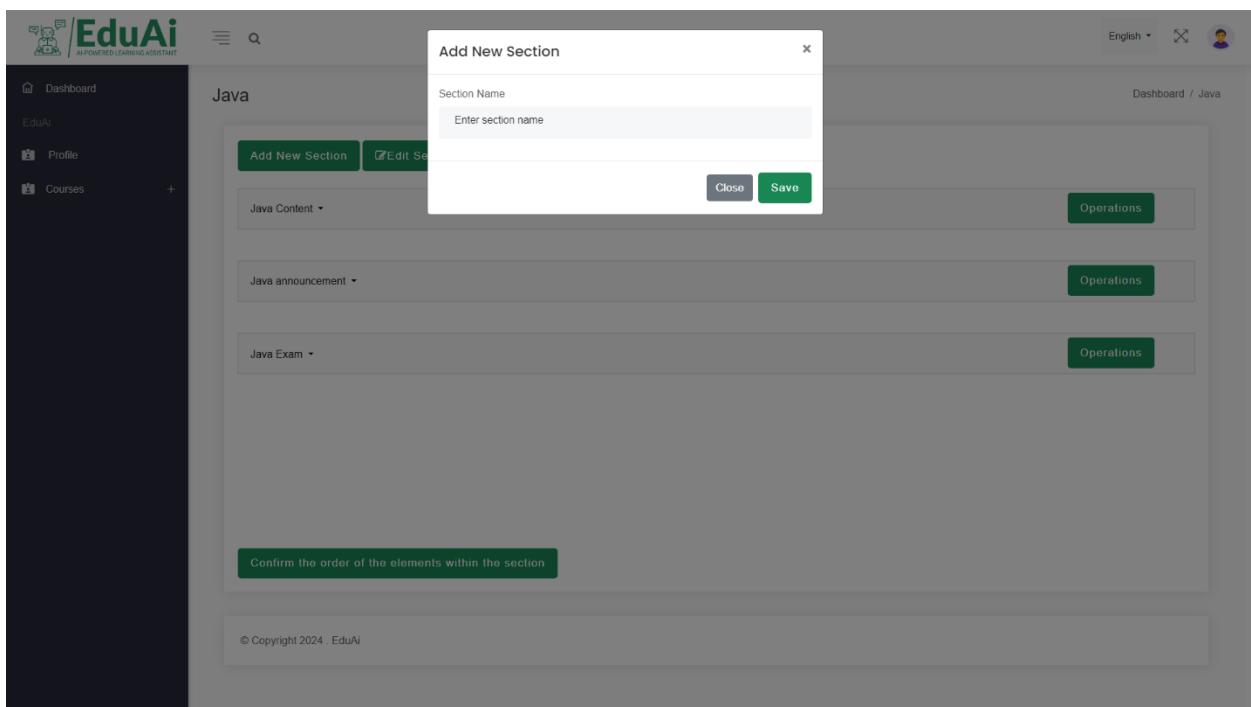


Figure 141: Instructor GUI 27

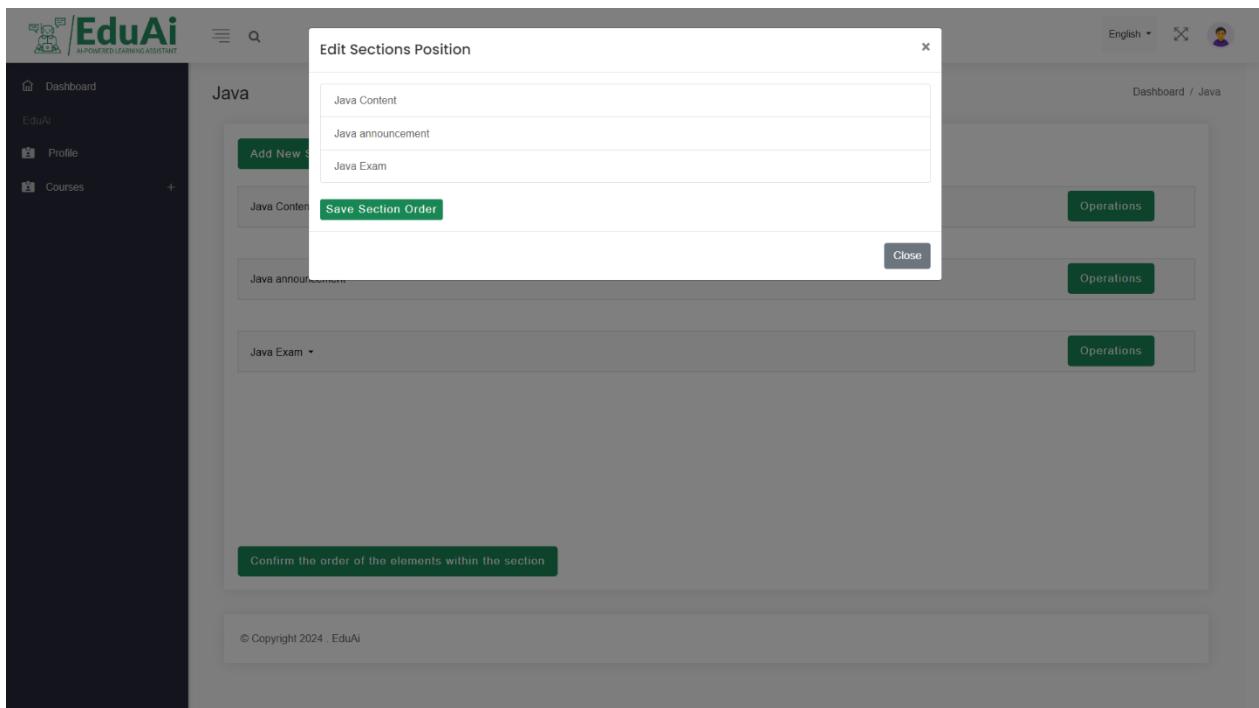


Figure 142: Instructor GUI 28

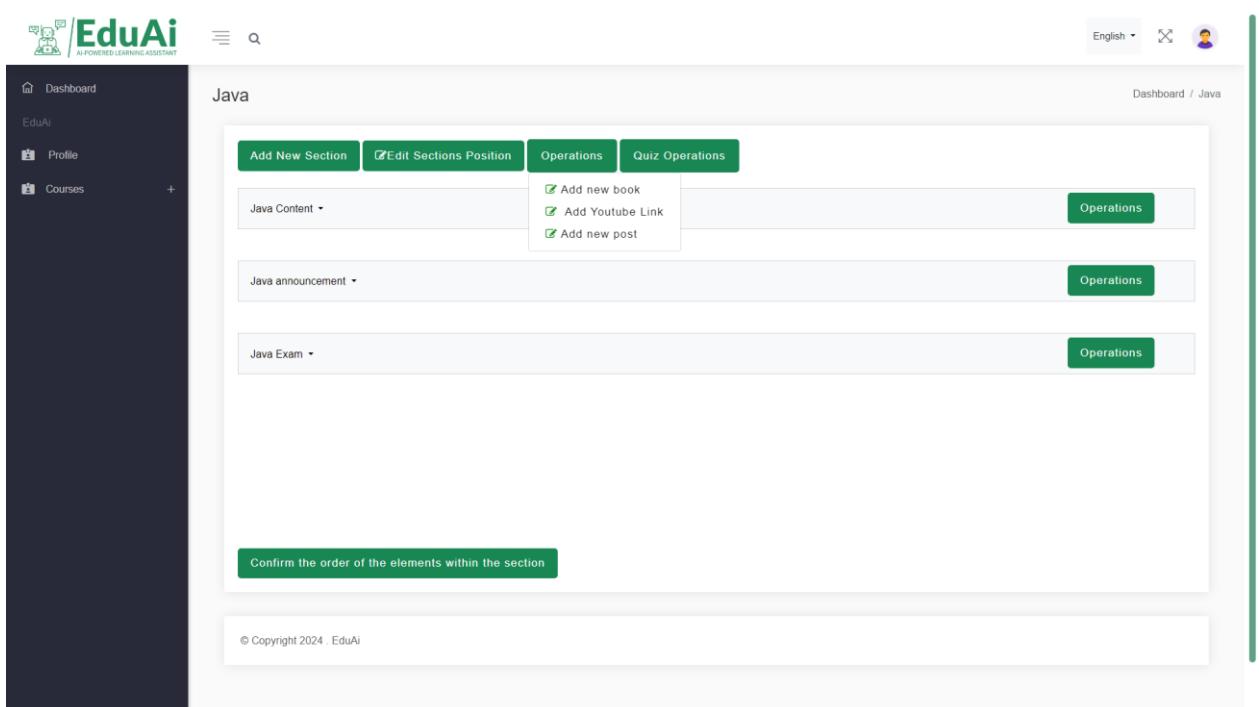


Figure 143: Instructor GUI 29

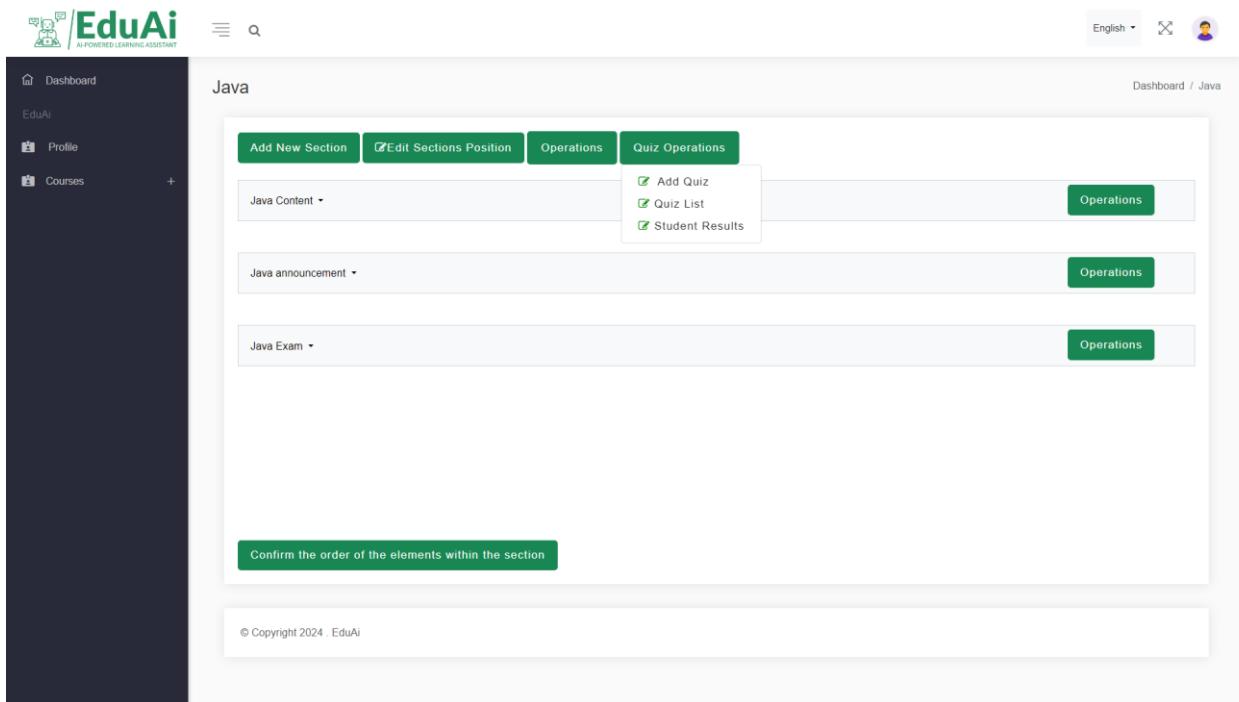


Figure 144: Instructor GUI 30

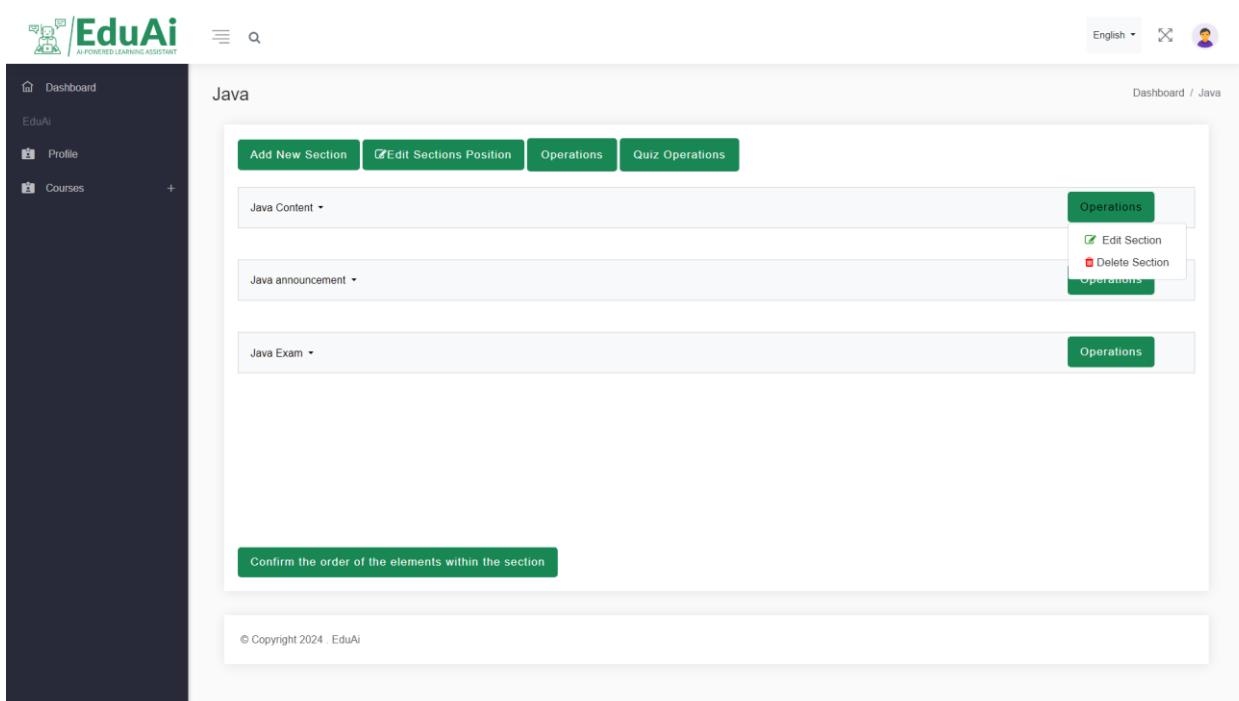


Figure 145: Instructor GUI 31

5.3.1 admin interface

The screenshot shows the EduAi Admin GUI. On the left is a dark sidebar with the EduAi logo and navigation links: Dashboard, Profile, Student (+), Teachers (+), Course List, and Course Overview. The main area has a header with English language selection, a search bar, and a user profile icon. The title 'Edit Instructor' is displayed above a form. The form fields include: Email (adnan@gmail.com), Password (empty), Name_ar (عذن سامر), Name_en (Adnan Samer), Joining_Date (2020-03-12), and Address (Amman). A green 'Add' button is at the bottom right of the form. The footer contains the copyright notice: © Copyright 2024 EduAi.

Figure 146: Admin GUI 1

The screenshot shows the EduAi Admin GUI. On the left is a dark sidebar with navigation links: Dashboard, Profile, Student, Teachers, Course List, and Course Overview. The main area is titled "Instructor List". A modal dialog box is open, asking "Delete Instructor data" and "Are you sure to delete the Instructor?". Below the modal is a table with one entry:

#	Name Instructor	Joining_Date	Processes
1	Adnan Samer	2020-03-12	Processes

At the bottom of the table, it says "Showing 1 to 1 of 1 entries". The footer of the page includes a copyright notice: "© Copyright 2024 EduAi".

Figure 147: Admin GUI 2

The screenshot shows the EduAi Admin GUI. The sidebar is identical to Figure 147. The main area is titled "Courses List". A table with one entry is displayed:

#	Course Name	Department(s)	Process
1	java	CIS, CS	Published

At the bottom of the table, it says "Showing 1 to 1 of 1 entries". The footer of the page includes a copyright notice: "© Copyright 2024 EduAi".

Figure 148: Admin GUI 3

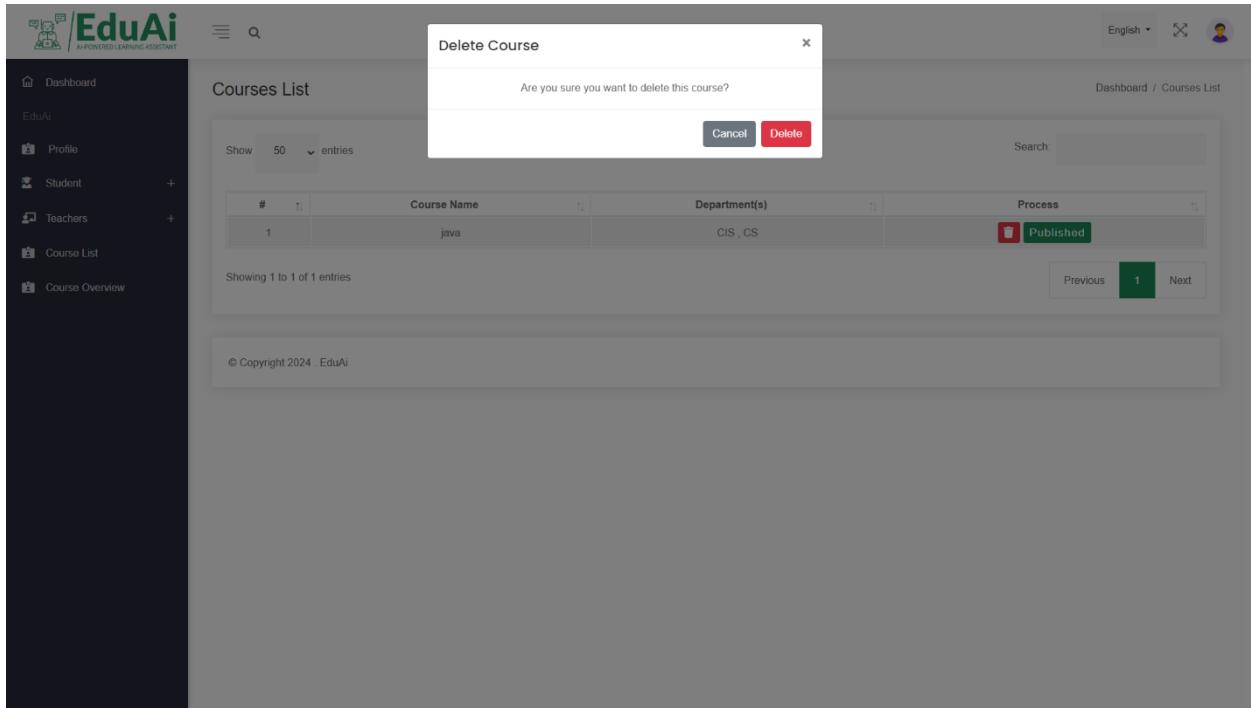


Figure 149: Admin GUI 4

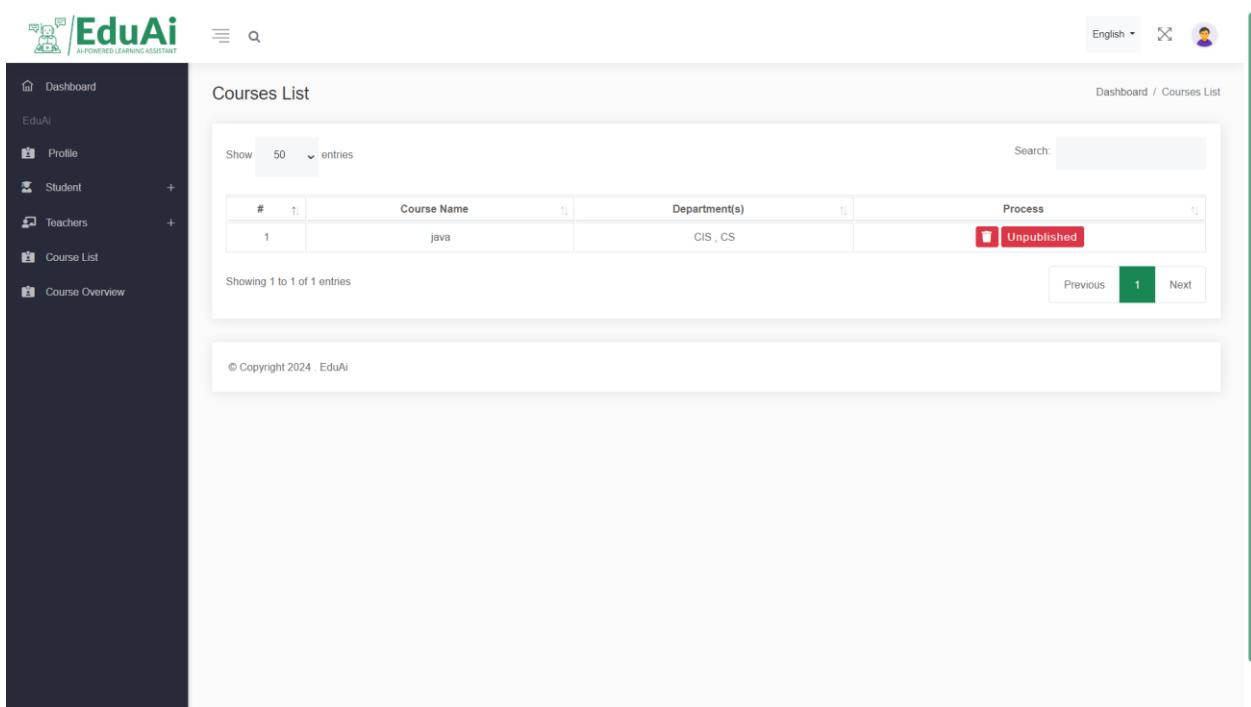


Figure 150: Admin GUI 5

Course Overview

#	Course Name	Students	Students who completed the course
1	java	0	0

Showing 1 to 1 of 1 entries

Previous **1** Next

© Copyright 2024 EduAi

Figure 151: Admin GUI 6

ABOUT US

Welcome to EduAi, where we believe in personalized education for every student. At EduAi, we understand that every student is unique, with different learning styles, strengths, and challenges. That's why we've created a platform that adapts to each student's needs, providing tailored educational resources and support to help them succeed. Our mission is to bridge the gap between students by providing equal opportunities for learning, regardless of background or circumstances. Whether you're struggling with a particular subject or aiming to excel in your studies, EduAi is here to support you every step of the way. Our team of dedicated educators and technologists is committed to empowering students to reach their full potential. Join us on this journey to revolutionize education and unlock the power of personalized learning for all.

Select the login method

CONTACT US

If you're interested in joining our team or simply reaching out to communicate, we're open and eager to connect with you. [Click Here](#)

Figure 152: Admin GUI 7

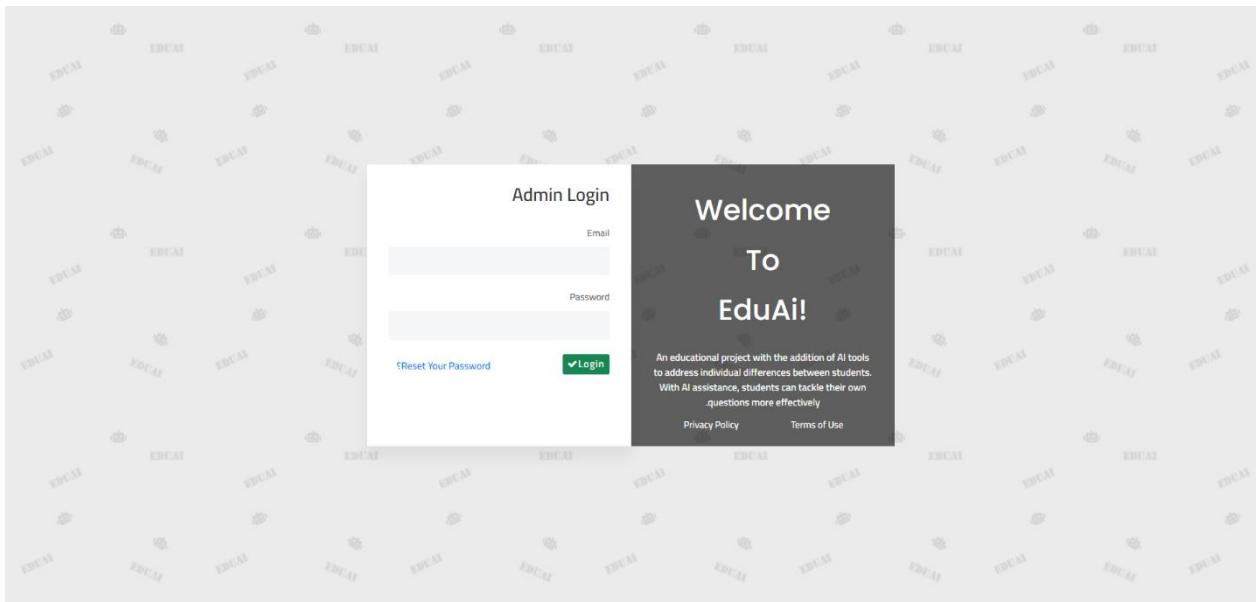


Figure 153: Admin GUI 8

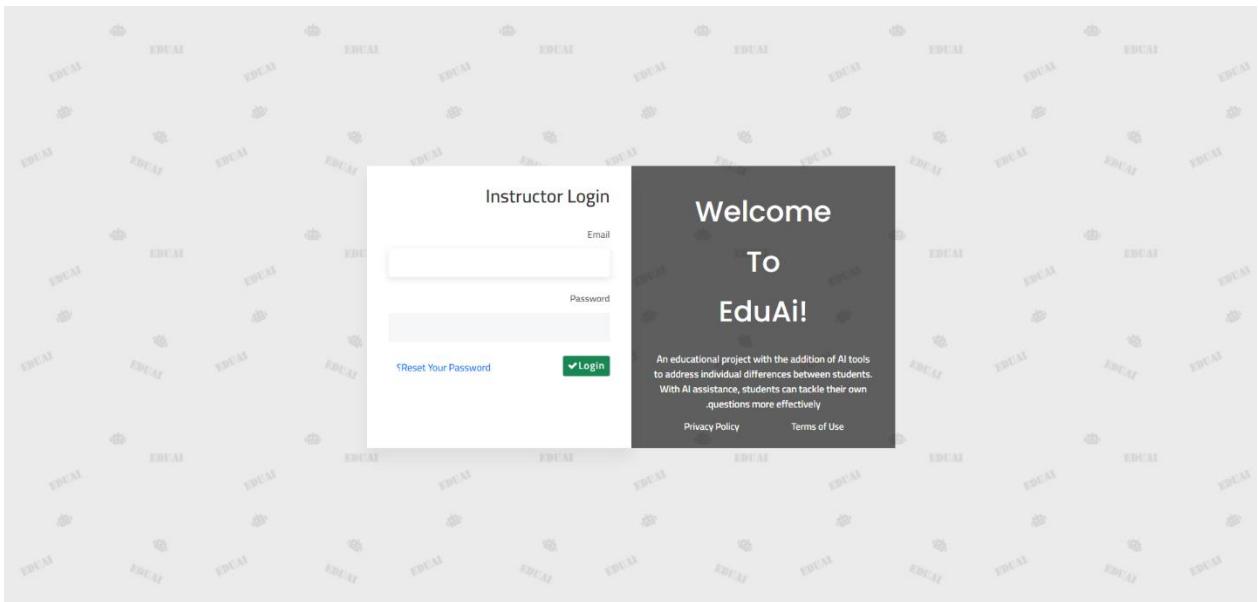


Figure 154: Admin GUI 9

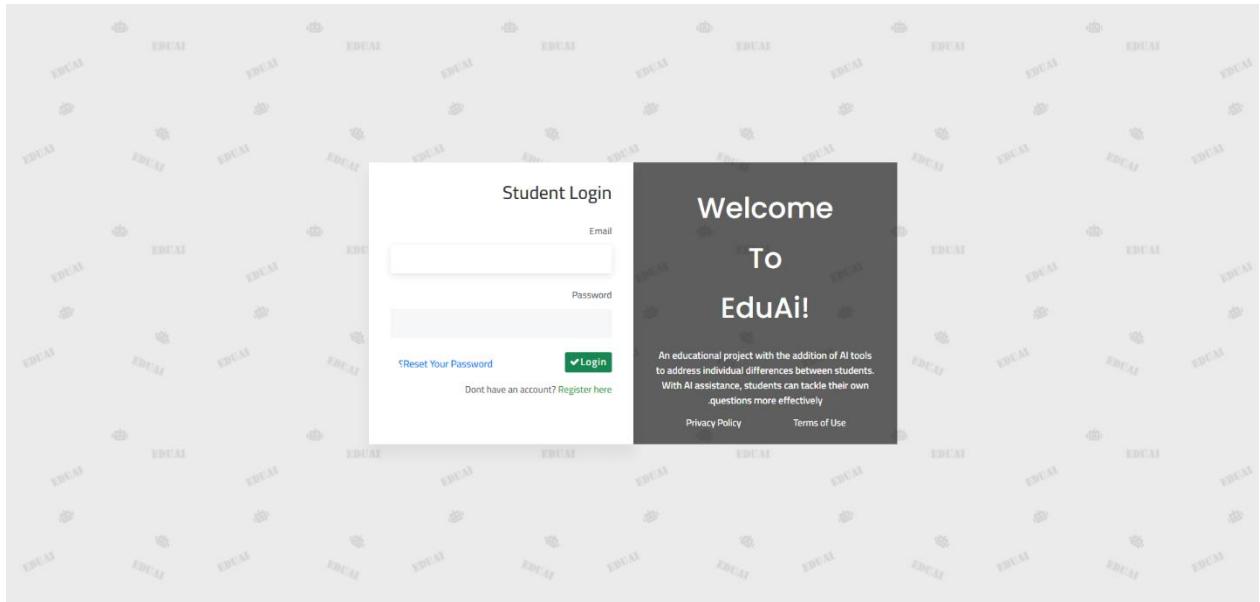


Figure 155: Admin GUI 10

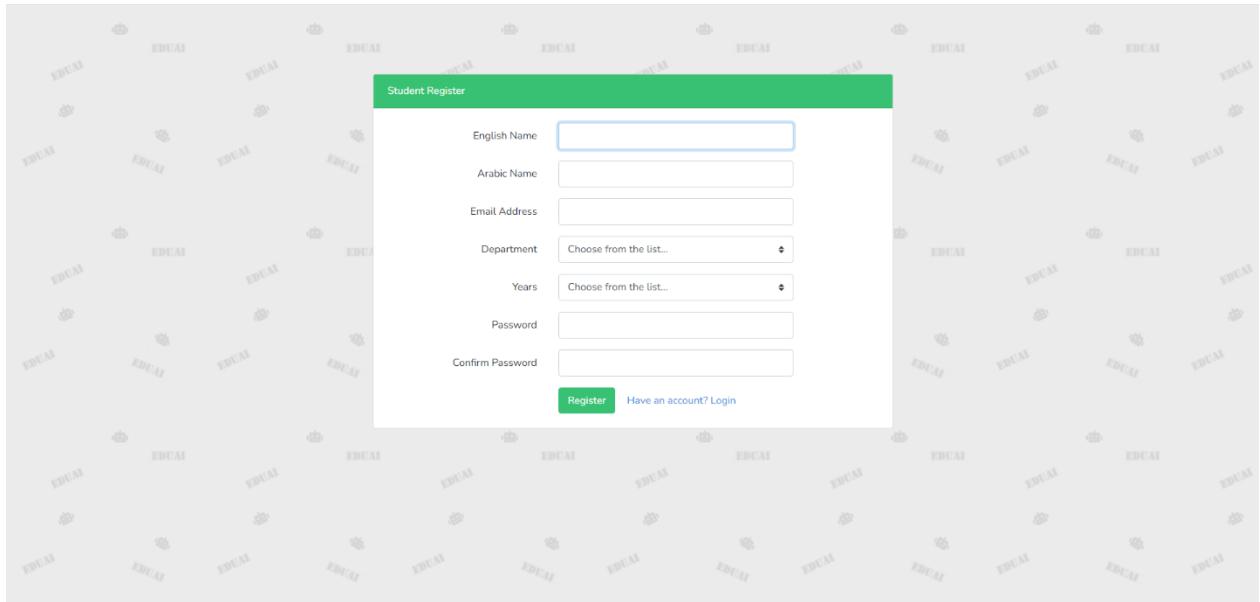


Figure 156: Admin GUI 11

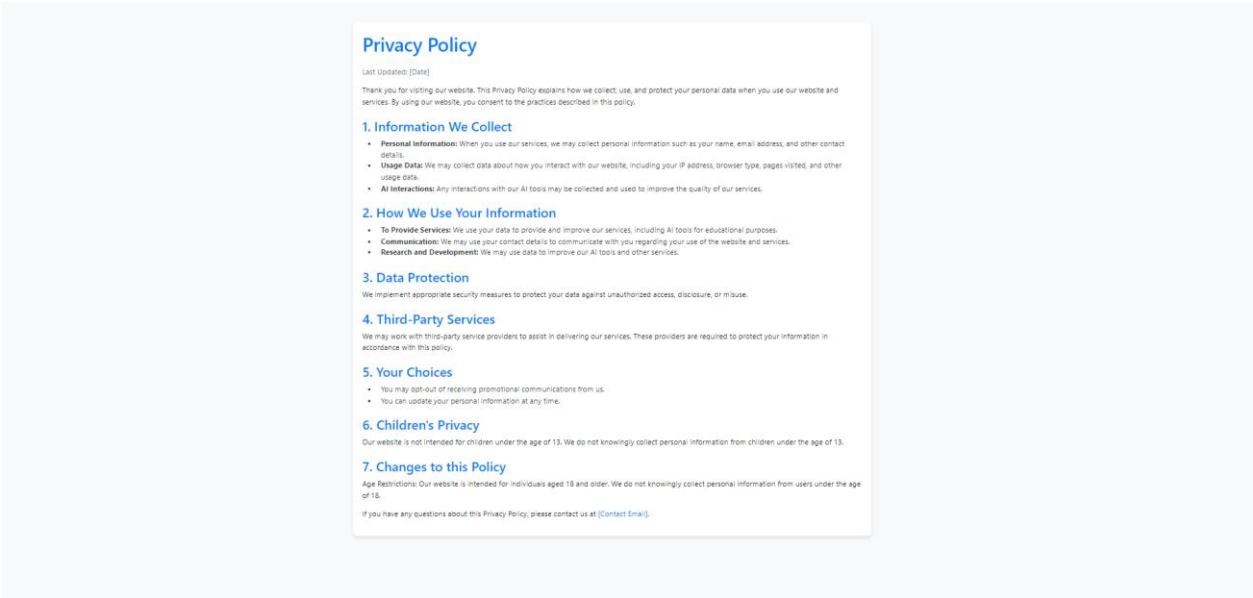


Figure 157: Admin GUI 12

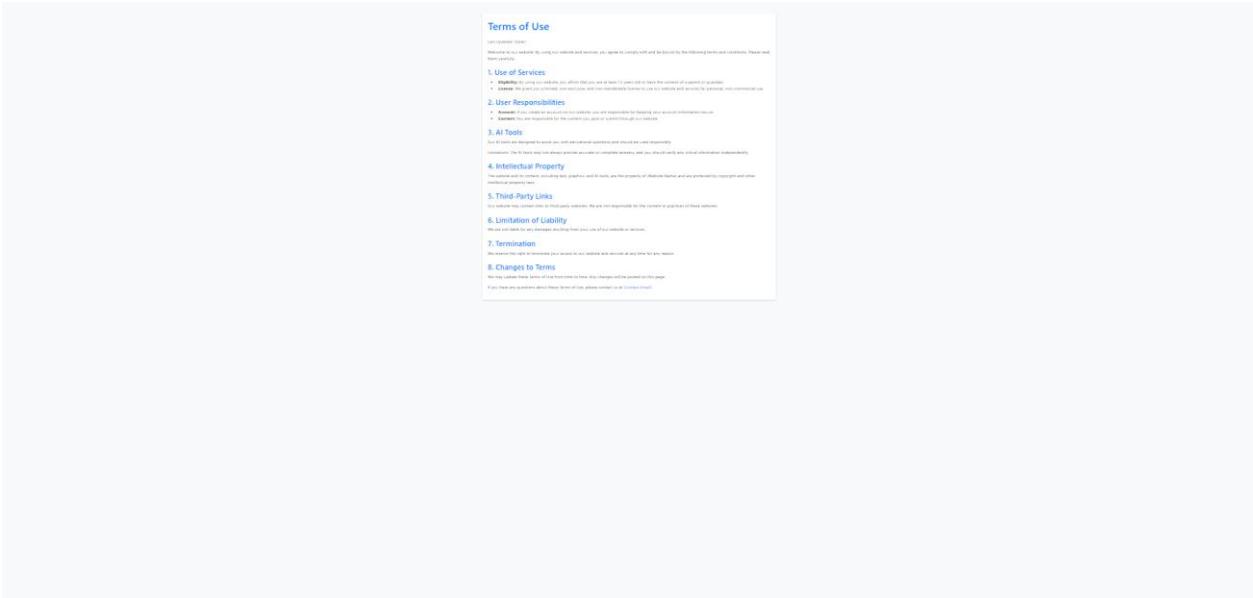


Figure 158: Admin GUI 13

The screenshot shows the EduAi Admin GUI. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, Student, Teachers, Course List, and Course Overview. The main area is titled "Dashboard". It features two summary cards: "Number Of Student" (1) with a graduation cap icon and "Number Of Instructor" (1) with a person at a desk icon. Below these are two "Data Display" sections. A table titled "Latest Processes On The System" lists one entry: Student Name (mohammmd), Student Email (mohammed@gmail.com), and Date added (2024-05-03 14:24:36). At the bottom is a copyright notice: © Copyright 2024 . EduAi.

Figure 159: Admin GUI 14

The screenshot shows the EduAi Admin GUI. The sidebar includes: Dashboard, EduAi, Profile, Student, Teachers, Course List, and Course Overview. The main area is titled "profile". It displays a user profile for "Mohanned" with a blue circular icon containing a person in a gear. The profile details are: mohanned@gmail.com and admin. To the right is a form for editing the profile: "Admin name in Arabic" (empty input field), "Admin name in English" (input field containing "Mohanned"), "password" (empty input field), and a "Show password" checkbox (unchecked). A "Data Edit" button is at the bottom of the form. At the top right, there is a breadcrumb navigation: Dashboard / profile. A copyright notice: © Copyright 2024 . EduAi is at the bottom.

Figure 160: Admin GUI 15

Add Student

Personal information

Student name in Arabic	Student name in English
Email	Password
Department Choose...	Year Choose...

Add

© Copyright 2024 EduAi

Figure 161: Admin GUI 16

Student List

Add student

#	Name	Email	Processes
1	mohammd	mohammed@gmail.com	Processes

Showing 1 to 1 of 1 entries

Previous **1** Next

© Copyright 2024 EduAi

Figure 162: Admin GUI 17

Add Student

Personal information

Student name in Arabic	Student name in English
<input type="text"/>	<input type="text"/>

Email	Password
<input type="text"/>	<input type="password"/>

Department	Year
<input type="text" value="Choose .."/>	<input type="text" value="Choose .."/>

Add

© Copyright 2024 EduAi

Figure 163: Admin GUI 18

Student Details

Name	Email	Year	Department
mohammmd	mohammed@gmail.com	First Year	CIS

© Copyright 2024 EduAi

Figure 164: Admin GUI 19

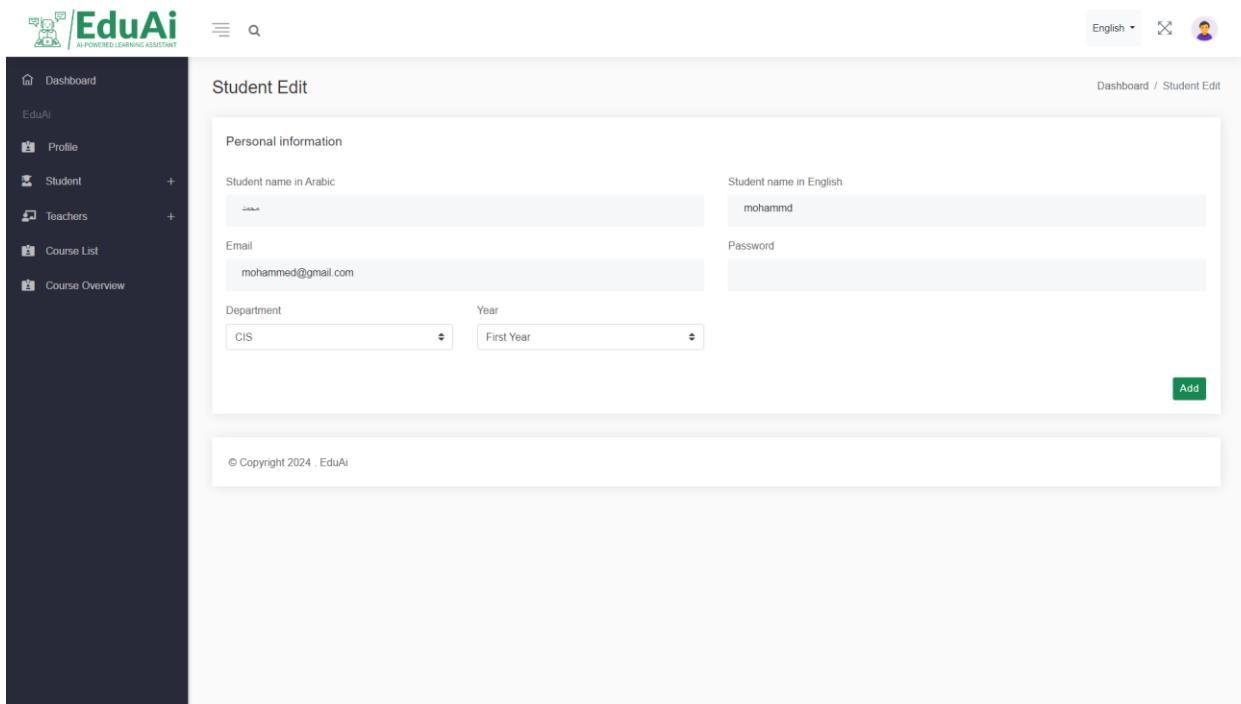


Figure 165: Admin GUI 20

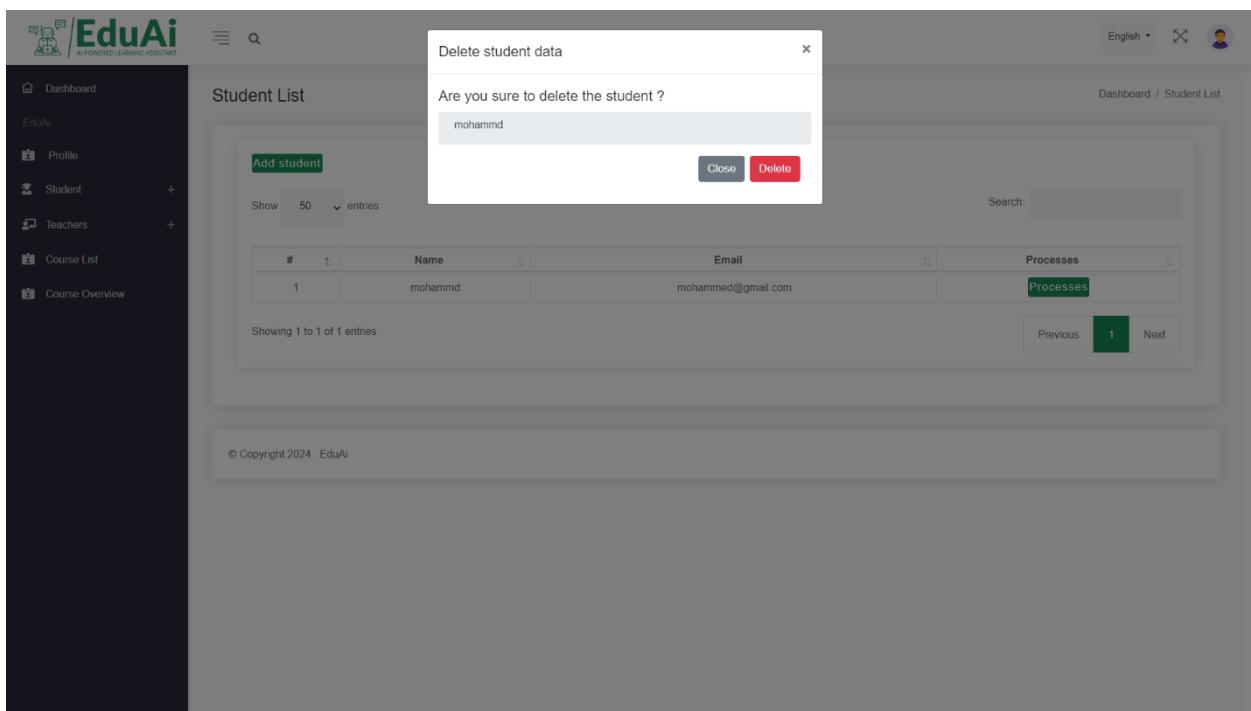


Figure 166: Admin GUI 21

The screenshot shows the EduAi Admin GUI. The left sidebar contains navigation links: Dashboard, Profile, Student, Teachers, Course List, and Course Overview. The main content area is titled "Instructor List". It features a table with one entry:

#	Name Instructor	Joining_Date	Processes
1	Adnan Samer	2020-03-12	<button>Processes</button>

Below the table, a message says "Showing 1 to 1 of 1 entries". At the bottom right are buttons for "Previous", "1", and "Next". The footer contains the copyright notice "© Copyright 2024 . EduAi".

Figure 167: Admin GUI 22

The screenshot shows the EduAi Admin GUI. The left sidebar contains navigation links: Dashboard, Profile, Student, Teachers, Course List, and Course Overview. The main content area is titled "Add Instructor". It contains several input fields:

Email	Password
Name_ar	Name_en
Joining_Date	
Address	

At the bottom left is a green "Add" button. The footer contains the copyright notice "© Copyright 2024 . EduAi".

Figure 168: Admin GUI 23

Chapter six: Testing and installation

6.1 Testing

6.1.1 Heuristic Evaluation

H1. Visibility of System Status

The system should always keep the user informed about what is going on, through appropriate feedback within a reasonable time.

Table 8: Visibility of System Status

#	Review Checklist	0	1	2	3	4
1.1	Does every display begin with a title or header that describes screen contents?	(X)	()	()	()	()
1.2	Do menu instructions, prompts, and error messages appear in the same place(s) on each menu?	()	()	()	()	(X)
1.3	Is there some form of system feedback for every operator action?	()	(X)	()	()	()
1.4	Are response times appropriate to the user's cognitive processing?	()	()	(X)	()	()
1.5	Is there visual feedback in menus or dialog boxes about which choices are selectable?	(X)	()	()	()	()

H2. Match Between System and the Real World

The system should speak the user's language, with words, phrases, and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

Table 9: Match Between System and the Real World

#	Review Checklist	0	1	2	3	4
2.1	Are icons concrete and familiar?	(X)	()	()	()	()
2.2	Are menu choices ordered in the most logical way, given the user, the item names, and the task variables?	()	(X)	()	()	()
2.3	Do related and interdependent fields appear on the same screen?	()	(X)	()	()	()
2.4	When prompts imply a necessary action, are the words in the message consistent with that action?	()	()	(X)	()	()
2.5	On data entry screens, are tasks described in terminology familiar to users?	(X)	()	()	()	()

H3. User Control and Freedom

Users should be free to select and sequence tasks (when appropriate), rather than having the system do this for them. Users often choose system functions by mistake and will need a clearly marked “emergency exit” to leave the unwanted state without having to go through an extended dialogue. Users should make their own decisions (with clear information) regarding the costs of exiting current work. The system should support undo and redo.

Table 10: User Control and Freedom

#	Review Checklist	0	1	2	3	4
3.1	When a user's task is complete, does the system wait for a signal from the user before processing?	()	(X)	()	()	()
3.2	Are users prompted to confirm commands that have drastic, destructive consequences?	()	()	(X)	()	()
3.3	Are character edits allowed in data entry fields?	()	(X)	()	()	()
3.4	If menu lists are long (more than seven items), can users select an item either by moving the cursor or by typing a mnemonic code?	(X)	()	()	()	()
3.5	If the system uses a pointing device, do users have the option of either clicking on menu items or using a keyboard shortcut?	()	()	(X)	()	()

H4. Consistency and Standards

Table 11: Consistency and Standards

#	Review Checklist	0	1	2	3	4
4.1	Has a heavy use of all uppercase letters on a screen been avoided?	()	()	(X)	()	()
4.2	Are icons labeled?	(X)	()	()	()	()
4.3	Are there no more than twelve to twenty icon types?	(X)	()	()	()	()
4.4	Does each window have a title?	()	(X)	()	()	()
4.5	Are vertical and horizontal scrolling possible in each window?	()	()	(X)	()	()
4.6	Are menu choice lists presented vertically?	(X)	()	()	()	()
4.7	Are menu titles either centered or left-justified?	()	(X)	()	()	()
4.8	Are menu items left-justified, with the item number or mnemonic preceding the name?	()	()	(X)	()	()
4.9	Do embedded field-level prompts appear to the right of the field label?	(X)	()	()	()	()
4.10	Are attention-getting techniques used with care?	()	()	(X)	()	()

H5. Help Users Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in plain language (NO CODES).

Table 12: Help Users Recognize, Diagnose, and Recover from Errors

#	Review Checklist	0	1	2	3	4
5.1	Is sound used to signal an error?	()	(X)	()	()	()
5.2	Are error messages worded so that the system, not the user, takes the blame?	()	()	()	(X)	()
5.3	Do error messages suggest the cause of the problem?	()	()	()	()	(X)
5.4	Do error messages indicate what action the user needs to take to correct the error?	()	()	()	(X)	()
5.5	If the system supports both novice and expert users, are multiple levels of error-message detail available?	()	()	()	()	(X)
5.6	If an error is detected in a data entry field, does the system place the cursor in that field or highlight the error?	()	()	()	(X)	()
5.7	Do error messages inform the user of the error's severity?	()	()	(X)	()	()

H6. Error Prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Table 13: Error Prevention

#	Review Checklist	0	1	2	3	4
6.1	Are menu choices logical, distinctive, and mutually exclusive?	()	(X)	()	()	()
6.2	Are data inputs case-blind whenever possible?	()	(X)	()	()	()
6.3	Does the system prevent users from making errors whenever possible?	()	()	(X)	()	()
6.4	Does the system warn users if they are about to make a potentially serious error?	()	()	()	()	(X)
6.5	Do data entry screens and dialog boxes indicate the number of character spaces available in a field?	()	(X)	()	()	()
6.6	Do fields in data entry screens and dialog boxes contain default values when appropriate?	()	(X)	()	()	()

H7. Recognition Rather Than Recall

Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Table 14: Recognition Rather Than Recall

#	Review Checklist	0	1	2	3	4
7.1	For question and answer interfaces, are visual cues and white space used to distinguish questions, prompts, instructions, and user input?	()	(X)	()	()	()
7.2	Are inactive menu items grayed out or omitted?	(X)	()	()	()	()
7.3	Do data entry screens and dialog boxes indicate when fields are optional?	()	(X)	()	()	()
7.4	Are prompts, cues, and messages placed where the eye is likely to be looking on the screen?	()	()	(X)	()	()
7.5	Are field labels close to fields, but separated by at least one space?	(X)	()	()	()	0
7.6	Have items been grouped into logical zones, and have headings been used to distinguish between zones?	()	(X)	()	()	()
7.7	Are borders used to identify meaningful groups?	(X)	()	()	()	()
7.8	Is color coding consistent throughout the system?	()	(X)	()	()	()

H8. Flexibility and Minimalist Design

Accelerators-unseen by the novice user-may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. Provide alternative means of access and operation for users who differ from the “average” user (e.g., physical or cognitive ability, culture, language, etc.)

Table 15: Flexibility and Minimalist Design

#	Review Checklist	0	1	2	3	4
8.1	If menu lists are short (seven items or fewer), can users select an item by moving the cursor?	()	(X)	()	()	()
8.2	If the system uses a pointing device, do users have the option of either clicking on fields or using a keyboard shortcut?	()	()	(X)	()	()
8.3	On data entry screens, do users have the option of either clicking directly on a field or using a keyboard shortcut?	()	()	(X)	()	()
8.4	On menus, do users have the option of either clicking directly on a menu item or using a keyboard shortcut?	()	()	(X)	()	()
8.5	In dialog boxes, do users have the option of either clicking directly on a dialog box option or using a keyboard shortcut?	()	()	()	(X)	()

H9. Aesthetic and Minimalist Design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Table 16: Aesthetic and Minimalist Design

#	Review Checklist	0	1	2	3	4
9.1	Are all icons in a set visually and conceptually distinct?	(X)	()	()	()	()
9.2	Does each icon stand out from its background?	()	(X)	()	()	()
9.3	Does each data entry screen have a short, simple, clear, distinctive title?	(X)	()	()	()	()
9.4	Are field labels brief, familiar, and descriptive?	(X)	()	()	()	()
9.5	Are there pop-up or pull-down menus within data entry fields that have many, but well-defined, entry options?	()	()	(X)	()	()

H10. Help and Documentation

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Table 17: Help and Documentation

#	Review Checklist	0	1	2	3	4
10.1	Are on-line instructions visually distinct?	()	(X)	()	()	()
10.2	If menu choices are ambiguous, does the system provide additional explanatory information when an item is selected?	()	()	()	(X)	()
10.3	Is the help function visible; for example, a key labeled HELP or a special menu?	()	()	(X)	()	()
10.4	Navigation: Is information easy to find?	(X)	()	()	()	()
10.5	Presentation: Is the visual layout well designed?	()	(X)	()	()	()
10.6	Conversation: Is the information accurate, complete, and understandable?	()	(X)	()	()	()
10.7	Is the information relevant?	(X)	()	()	()	()
10.8	Can users easily switch between help and their work?	()	()	(X)	()	()
10.9	Is it easy to access and return from the help system?	()	(X)	()	()	()
10.10	Can users resume work where they left off after accessing help?	()	(X)	()	()	()

6.1.2 Cooperative Evaluation

Table 18: Participants Details

No.	Criteria	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6
1.	Gender	Male	Female	Male	Male	Male	Male
2.	Age	30	28	20	21	25	29
3.	Educationa l Level	Bachelor's	Bachelor's	Bachelor's	Bachelor's	Bachelor's	Bachelor's
4.	Programme Taken	Cis	Cs	Cs	Cs	Cis	Cis
5.	Institution	Universit y of Jordan					

a) Pre-Evaluation Procedures

Participants were contacted through telephone conversations asking them about the possibility to participate in the co-operative evaluation. A brief introduction to the A system of continuous assistance and evaluation of academic processes system was given to the participants 10 minutes before they started the evaluation, and participants were asked to read that introductory document. The document also has a list of tasks, which will be performed by the participants throughout the co-operative evaluation. Users were told that they need to think aloud when facing any problem in the system. They were also told that, each task they perform is monitored and timed.

b) Evaluation Procedures

During the evaluation session, a moderator accompanied the users to do the co-operative evaluation. A comment from shown in Appendix A was used by the moderator to write down the comments of each user for each task. Users were helped when they really faced serious problems performing the tasks. The following tables show the comments from prepared by the moderator for each participant.

Table 19: Cooperative Evaluation 1

Task No.	Test	Time Taken to Complete the Task	Comments
A.	Login		
1.	Log in to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	1 minutes and 10 seconds.	<ul style="list-style-type: none"> • I can logged in fast and easily
B.	Register		
1.	Register to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	1 minutes and 10 seconds.	<ul style="list-style-type: none"> • I need more information to Register
C.	Logout		
1.	Logout from A system of continuous assistance and evaluation of academic processes	50 second	<ul style="list-style-type: none"> • I can logged out fast and easily
D.	Display department		
1.	the user can display department names and information about each department.	1 minutes and 30 seconds.	<ul style="list-style-type: none"> • How is the information displayed?
E.	Display department details		

	1. the user can display subjects for each department and plan for each department.	1 minutes	<ul style="list-style-type: none">• Is the information presented clearly and organized?
F.	Contact us		
1.	contains our email to contact with us	2 minutes and 30 seconds.	<ul style="list-style-type: none">• Are there any concerns about privacy or security?
G.	Search		
1.	the user can search for a specific major to apply an action on it.	2 minutes and 40 seconds.	<ul style="list-style-type: none">• Is the search scope clearly defined?

Table 20: Cooperative Evaluation 2

Task No.	Test	Time Taken to Complete the Task	Comments
A.	Login		
1.	Log in to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	2 minutes	<ul style="list-style-type: none"> • I can logged in fast and easily
B.	Register		
1.	Register to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	50 seconds.	<ul style="list-style-type: none"> • I need more information to Register
C.	Logout		
1.	Logout from A system of continuous assistance and evaluation of academic processes	2 minutes and 22 seconds.	<ul style="list-style-type: none"> • I can logged out fast and easily
D.	Display department		
1.	the user can display department names and information about each department.	2 minutes	<ul style="list-style-type: none"> • How is the information displayed?
E.	Display department details		

	1. the user can display subjects for each department and plan for each department.	1 minutes	<ul style="list-style-type: none">• Is the information presented clearly and organized?
F.	Contact us		
1.	contains our email to contact with us	2 minutes and 50 seconds.	<ul style="list-style-type: none">• Are there any concerns about privacy or security?
G.	Search		
1.	the user can search for a specific major to apply an action on it.	2 minutes and 40 seconds.	<ul style="list-style-type: none">• Is the search scope clearly defined?

Table 21: Cooperative Evaluation 3

Task No.	Test	Time Taken to Complete the Task	Comments
A.	Login		
1.	Log in to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	1 minutes and 20 seconds.	<ul style="list-style-type: none"> • I can logged in fast and easily
B.	Register		
1.	Register to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	2 minutes and 10 seconds..	<ul style="list-style-type: none"> • I need more information to Register
C.	Logout		
1.	Logout from A system of continuous assistance and evaluation of academic processes	2 minutes and 3 seconds.	<ul style="list-style-type: none"> • I can logged out fast and easily
D.	Display department		
1.	the user can display department names and information about each department.	3 minutes	<ul style="list-style-type: none"> • How is the information displayed?
E.	Display department details		

	1. the user can display subjects for each department and plan for each department.	2 minutes	<ul style="list-style-type: none">• Is the information presented clearly and organized?
F.	Contact us		
1.	contains our email to contact with us	2 minutes and 45 seconds.	<ul style="list-style-type: none">• Are there any concerns about privacy or security?
G.	Search		
1.	the user can search for a specific major to apply an action on it.	2 minutes and 50 seconds.	<ul style="list-style-type: none">• Is the search scope clearly defined?

Table 22: Cooperative Evaluation 4

Task No.	Test	Time Taken to Complete the Task	Comments
A.	Login		
1.	Log in to the A system of continuous assistance evaluation of academic processes using the following information: Username: Mohannad Password:123456	1 minutes and 15 seconds.	<ul style="list-style-type: none">• I can logged in fast and easily
B.	Register		

1.	Register to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	55 seconds..	<ul style="list-style-type: none"> I need more information to Register
C. Logout			
1.	Logout from A system of continuous assistance and evaluation of academic processes	1 minutes	<ul style="list-style-type: none"> I can logged out fast and easily
D. Display department			
1.	the user can display department names and information about each department.	2 minutes and 20 seconds.	<ul style="list-style-type: none"> How is the information displayed?
E. Display department details			
1.	the user can display subjects for each department and plan for each department.	3 minutes	<ul style="list-style-type: none"> Is the information presented clearly and organized?
F. Contact us			
1.	contains our email to contact with us	2 minutes and 30 seconds.	<ul style="list-style-type: none"> Are there any concerns about privacy or security?
G. Search			

1.	the user can search for a specific major to apply an action on it.	2 minutes and 20 seconds.	<ul style="list-style-type: none"> • Is the search scope clearly defined?
----	--	---------------------------	--

Table 23: Cooperative Evaluation 5

Task No.	Test	Time Taken to Complete the Task	Comments
A.	Login		
1.	Log in to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	50 seconds..	<ul style="list-style-type: none"> • I can logged in fast and easily
B.	Register		
1.	Register to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	2 minutes	<ul style="list-style-type: none"> • I need more information to Register
C.	Logout		
1.	Logout from A system of continuous assistance and evaluation of academic processes	2 minutes	<ul style="list-style-type: none"> • I can logged out fast and easily
D.	Display department		

1.	the user can display department names and information about each department.	2 minutes and 10 seconds.	<ul style="list-style-type: none"> • How is the information displayed?
E.	Display department details		
1.	the user can display subjects for each department and plan for each department.	1 minutes and 30 seconds.	<ul style="list-style-type: none"> • Is the information presented clearly and organized?
F.	Contact us		
1.	contains our email to contact with us	2 minutes and 50 seconds.	<ul style="list-style-type: none"> • Are there any concerns about privacy or security?
G.	Search		
1.	the user can search for a specific major to apply an action on it.	2 minutes and 22 seconds.	<ul style="list-style-type: none"> • Is the search scope clearly defined?

Table 24: Cooperative Evaluation 6

Task No.	Test	Time Taken to Complete the Task	Comments
A.	Login		
1.	Log in to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	30 seconds	<ul style="list-style-type: none"> • I can logged in fast and easily
B.	Register		
1.	Register to the A system of continuous assistance and evaluation of academic processes using the following information: Username: Mohannad Password:123456	2 minutes and 50 seconds.	<ul style="list-style-type: none"> • I need more information to Register
C.	Logout		
1.	Logout from A system of continuous assistance and evaluation of academic processes	1 minutes	<ul style="list-style-type: none"> • I can logged out fast and easily
D.	Display department		
1.	the user can display department names and information about each department.	1 minutes and 45 seconds.	<ul style="list-style-type: none"> • How is the information displayed?
E.	Display department details		

	1. the user can display subjects for each department and plan for each department.	1 minutes and 40 seconds.	• Is the information presented clearly and organized?
F.	Contact us		
1.	contains our email to contact with us	3 minutes and 30 seconds.	• Are there any concerns about privacy or security?
G.	Search		
1.	the user can search for a specific major to apply an action on it.	2 minutes and 45 seconds.	• Is the search scope clearly defined?

Table 25: Task Completion Times in Minutes and Seconds

Task No.	Default	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6
A. Login							
1.	00:20	01:10	02:00	01:20	01:15	00:50	00:30
B. Register							
1.	01:00	01:10	00:50	02:10	00:55	02:00	02:50
C. Logout							
1.	00:30	00:50	02:22	02:03	01:00	02:00	01:00
D. Display department							
1.	01:20	01:30	02:00	03:00	02:20	02:10	01:45
E. Display department details							
1.	00:50	01:00	01:20	02:00	03:00	01:30	01:40
F. Contact us							
1.	01:50 02:30		02:50	02:45	02:30	02:50	03:30
G. Search							
1.	02.00	02.30	02.40	02.50	02.20	02.22	02.45
Total Completion Time	07:50	10:40	14:02	16:08	13:20	13:42	14:00

c) Post-Evaluation Procedures

After completing the co-operative evaluation, participants were given a post-test questionnaire to fill in, which is shown in Appendix B. This questionnaire was important to capture their thoughts and feelings and the system of continuous assistance and evaluation of academic processes while they were still fresh. The questionnaire was then followed by a short interview and discussion, which mainly focused on the initial modified design of the the system of continuous assistance and evaluation of academic processes. Table 23 shows the responses of the 6 participants to the post-test questionnaire

Table 26: Participants Responses to the Post-Test Questionnaire

No.	Statement	Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Average
1	A system of continuous assistance and evaluation of academic processes is easy to use.	4	2	1	3	4	1	2.50
2	A system of continuous assistance and evaluation of academic processes has accomplished its goals.	2	3	3	4	3	2	2.83
3	A system of continuous assistance and evaluation of academic processes interface is interactive.	4	3	3	2	3	1	2.66
4	A system of continuous assistance and evaluation of academic processes is easy to understand and easy to follow.	4	2	2	3	1	3	2.50
5	It is easy to understand the functionality of the system without prior experience.	4	4	1	4	3	3	3.16
6	A system of continuous assistance and evaluation of academic processes is enjoyable website.	3	4	1	4	4	4	3.33
7	Interactions tap provides clear and easy to understand information.	4	3	3	2	3	4	3.17
8	The concept of A system of continuous assistance and evaluation of academic processes was difficult to understand.	1	4	4	4	2	2	2.83
9	I liked the look and feel of the Website.	3	3	3	2	4	3	3.00
10	A system of continuous assistance and evaluation of academic processes felt complete.	4	2	3	3	3	2	2.83
11	I felt that the tasks were difficult to complete using the Website.	1	2	3	2	1	2	1.83
Average		3.09	2.9	2.18	3.00	2.81	2.45	2.78

Test (Post-test Questionnaire)

After using the tool and answering the system of continuous assistance and evaluation of academic processes test, please indicate the extent to which you agree or disagree with each of the following statements regarding to your experience with the tool.

Table 27: Test (Post-test Questionnaire)

No.	Statements	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
		1	2	3	4	5
1.	the system of continuous assistance and evaluation of academic processes is easy to use.		✓			
2.	the system of continuous assistance and evaluation of academic processes has accomplished its goals.			✓		
3.	the system of continuous assistance and evaluation of academic processes interface is interactive.			✓		
4.	the system of continuous assistance and evaluation of academic processes is easy to understand and easy to follow.		✓			
5.	It is easy to understand the functionality of the system without prior experience.			✓		
6.	the system of continuous assistance and evaluation of academic processes is enjoyable Website.			✓		
7.	Interactions tap provides clear and easy to understand information.			✓		
8.	The concept of the system of continuous assistance and evaluation of academic processes was difficult to understand.			✓		
9.	I liked the look and feel of the Website.			✓		
10.	the system of continuous assistance and evaluation of academic processes felt complete.			✓		

11.	I felt that the tasks were difficult to complete using the website.		✓			
-----	---	--	---	--	--	--

6.2 Installation

1. Visual studio code.
2. Xampp.
3. Composer.
4. Git.
5. Node.js.
6. XMind.

6.3 User manual

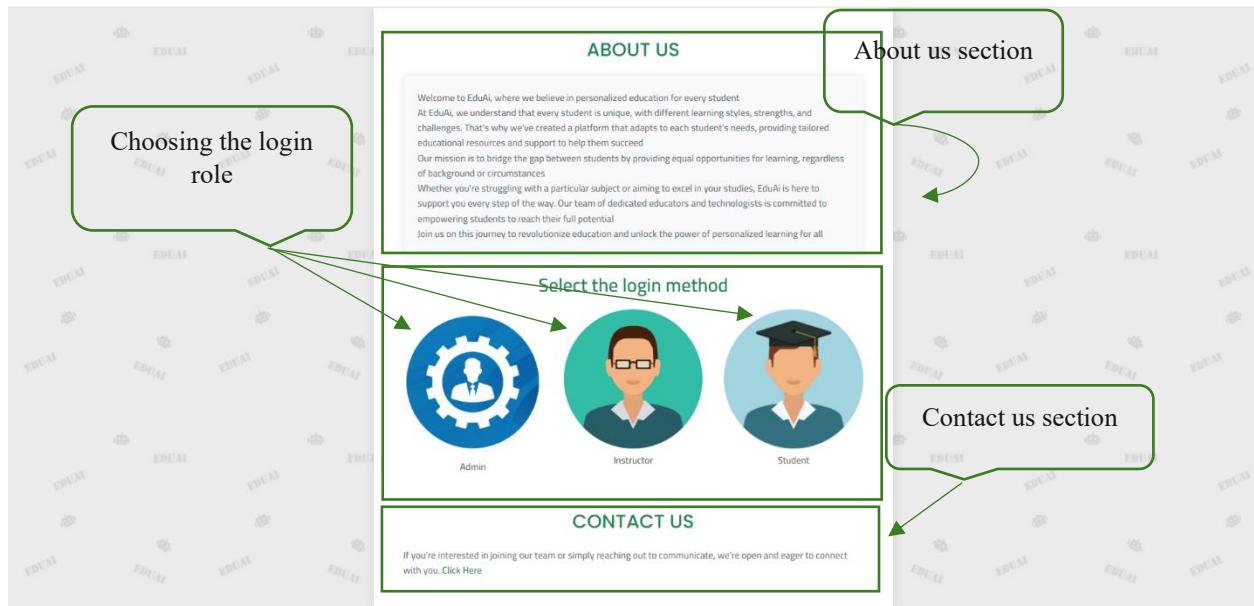


Figure 169: User guide 1

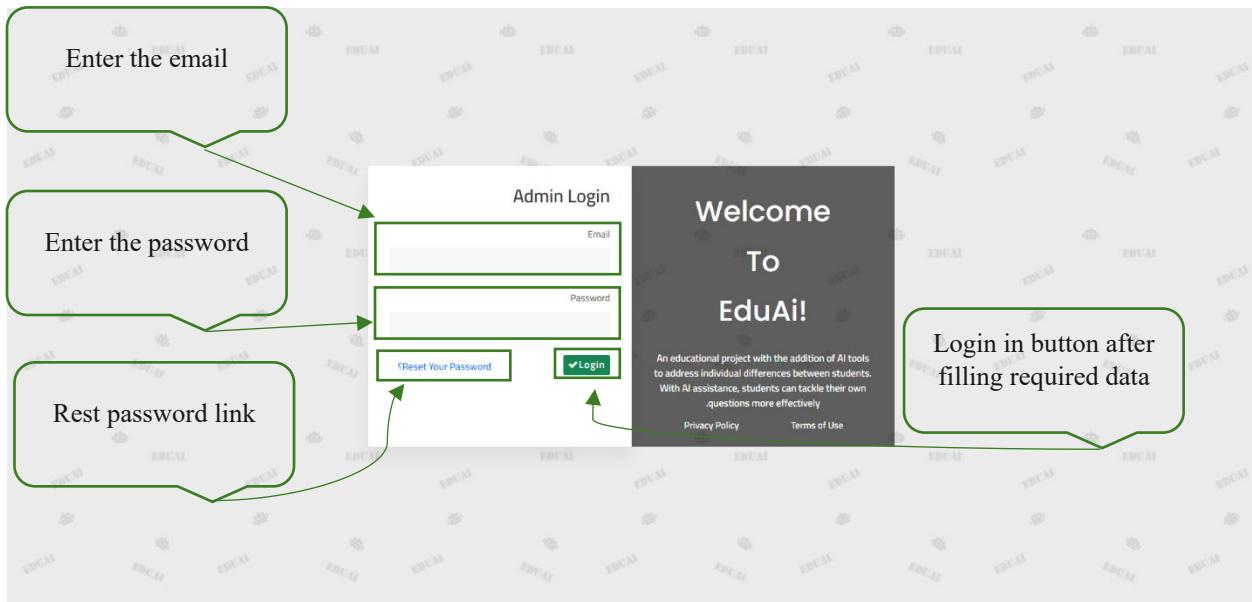


Figure 170: User guide 2

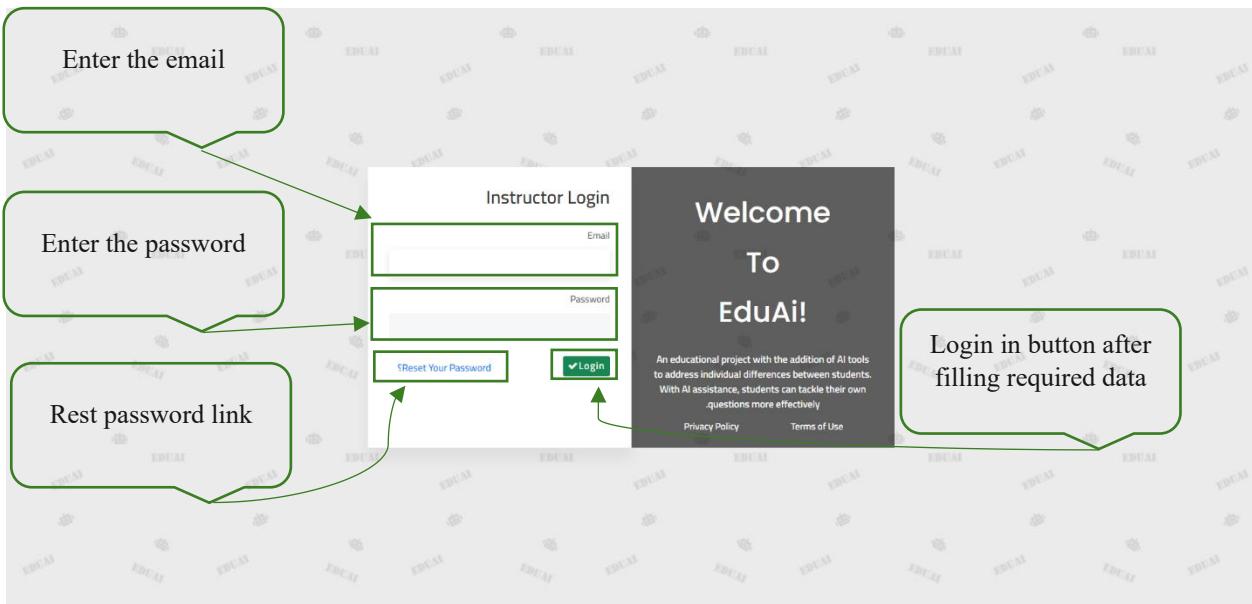


Figure 171: User guide 3

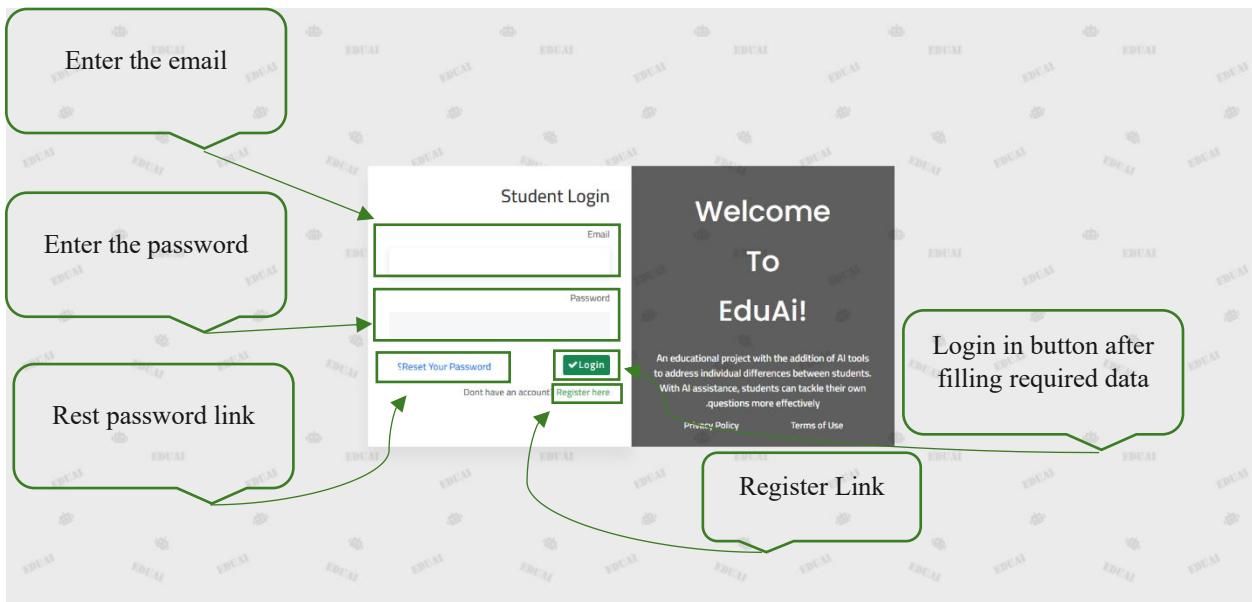


Figure 172: User guide 4

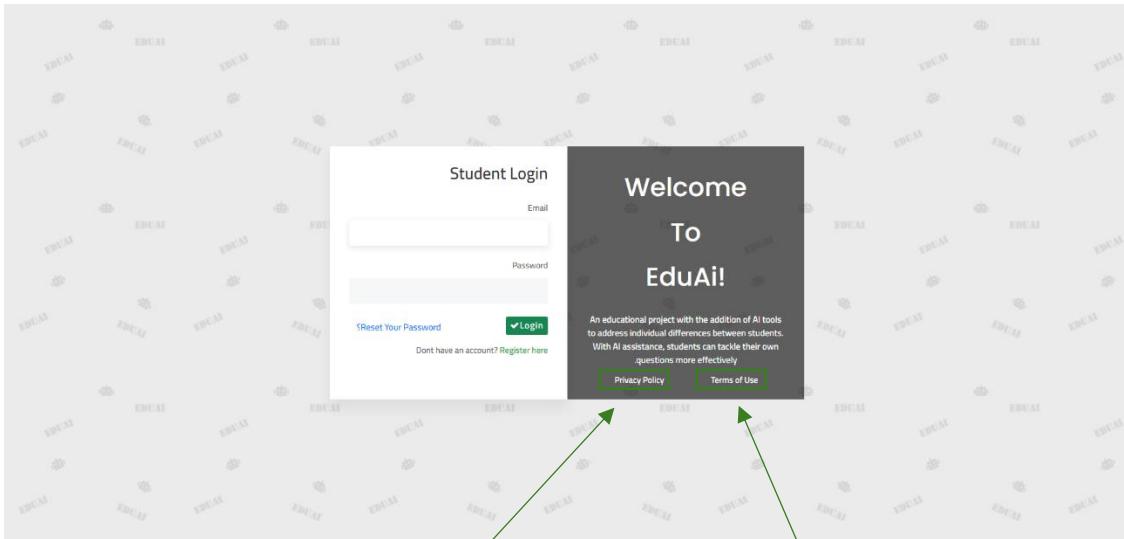


Figure 174: User guide 5

Privacy Policy

Last Updated: [Date]

Thank you for visiting our website. This Privacy Policy explains how we collect, use, and protect your personal data when you use our website and services. By using our website, you consent to the practices described in this policy.

1. Information We Collect

- Personal Information:** When you use our services, we may collect personal information such as your name, email address, and other contact details.
- Usage Data:** We may collect data about how you interact with our website, including your IP address, browser type, pages visited, and other usage data.
- AI Interactions:** Any interactions with our AI tools may be collected and used to improve the quality of our services.

2. How We Use Your Information

- To Provide Services:** We use your data to provide and improve our services, including AI tools for educational purposes.
- Communication:** We may use your contact details to communicate with you regarding your use of the website and services.
- Research and Development:** We may use data to improve our AI tools and other services.

3. Data Protection

We implement appropriate security measures to protect your data against unauthorized access, disclosure, or misuse.

4. Third-Party Services

We may work with third-party service providers to assist in delivering our services. These providers are required to protect your information in accordance with this policy.

5. Your Choices

- You may opt-out of receiving promotional communications from us.
- You can update your personal information at any time.

6. Children's Privacy

Our website is not intended for children under the age of 13. We do not knowingly collect personal information from children under the age of 13.

7. Changes to this Policy

Age Restrictions: Our website is intended for individuals aged 18 and older. We do not knowingly collect personal information from users under the age of 18.

If you have any questions about this Privacy Policy, please contact us at EduAi.

Terms of Use

Last Updated: [Date]

Welcome to our website! By using our website and services, you agree to comply with and be bound by the following terms and conditions. Please read them carefully.

1. Use of Services

- Eligibility:** By using our website, you affirm that you are at least 13 years old or have the consent of a parent or guardian.
- License:** We grant you a limited, non-exclusive, and non-transferable license to use our website and services for personal, non-commercial use.

2. User Responsibilities

- Account:** If you create an account on our website, you are responsible for keeping your account information secure.
- Content:** You are responsible for the content you post or submit through our website.

3. AI Tools

Our AI tools are designed to assist you with educational questions and should be used responsibly.

Limitations: The AI tools may not always provide accurate or complete answers, and you should verify any critical information independently.

4. Intellectual Property

The website and its content, including text, graphics, and AI tools, are the property of [Website Name] and are protected by copyright and other intellectual property laws.

5. Third-Party Links

Our website may contain links to third-party websites. We are not responsible for the content or practices of these websites.

6. Limitation of Liability

We are not liable for any damages resulting from your use of our website or services.

7. Termination

We reserve the right to terminate your access to our website and services at any time for any reason.

8. Changes to Terms

We may update these Terms of Use from time to time. Any changes will be posted on this page.

If you have any questions about these Terms of Use, please contact us at EduAi.

Figure 173: User guide 7

Figure 175: User guide 6

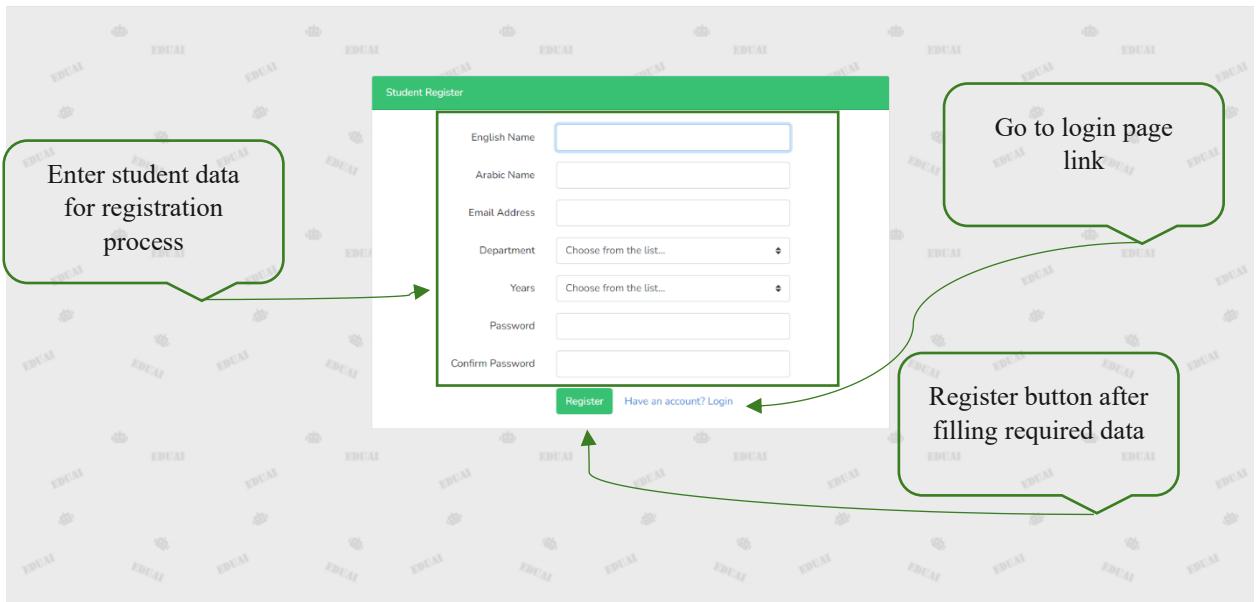


Figure 176: User guide 8

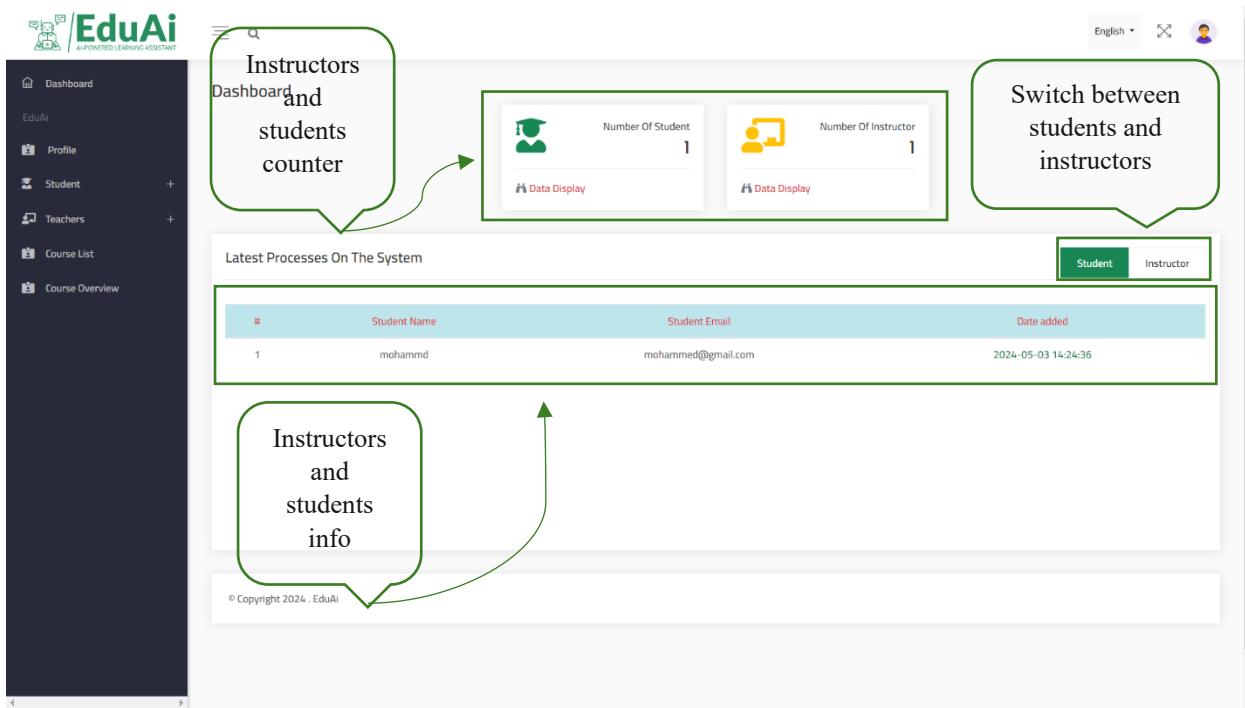


Figure 177: User guide 9

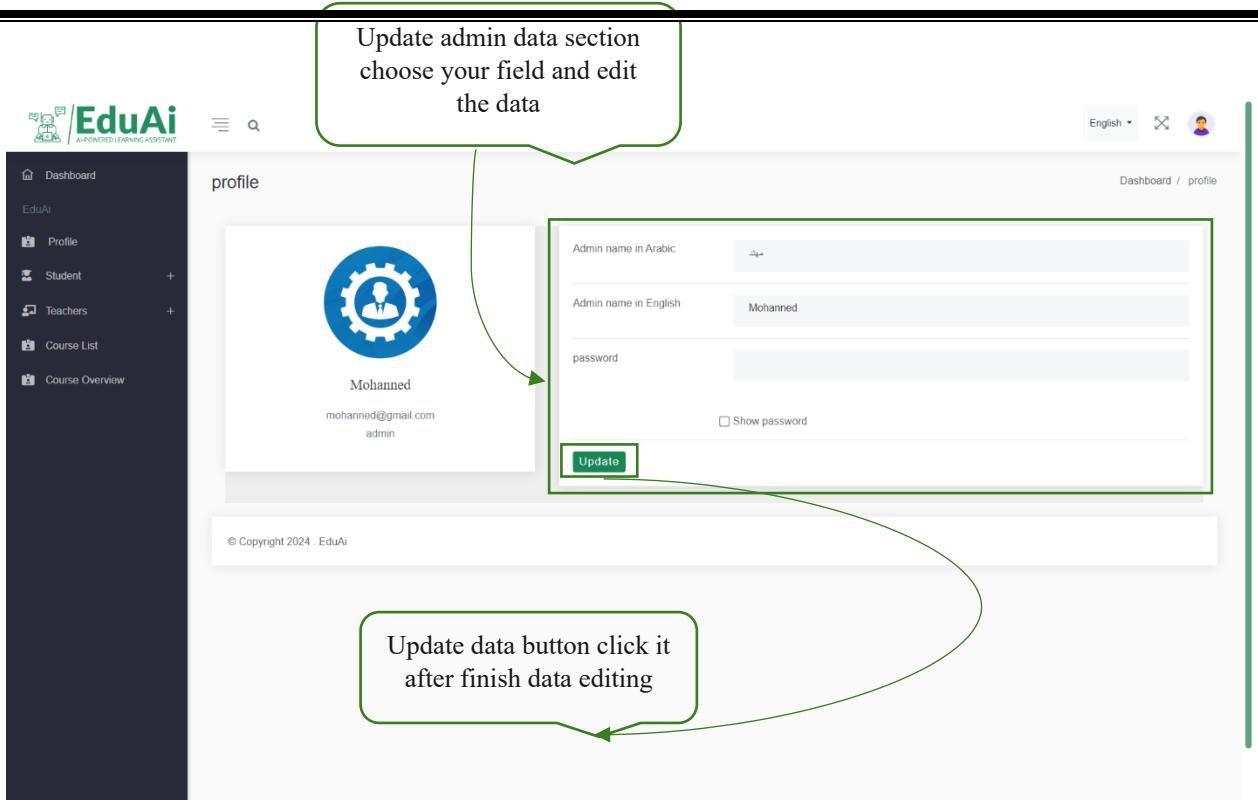


Figure 178: User guide 10

The screenshot shows the EduAi platform interface. On the left is a dark sidebar with navigation options: Dashboard, Profile, Student, Teachers, Course List, and Course Overview. The main area is titled "Student Details" and displays a table with one row of data:

Name	Email	Year	Department
mohammd	mohammed@gmail.com	First Year	CIS

Below the table is a copyright notice: © Copyright 2024 · EduAi. A green callout bubble on the right side points to the table, labeled "Student details list". The top right corner of the main area shows "Dashboard / Student Details".

Figure 179: User guide 11

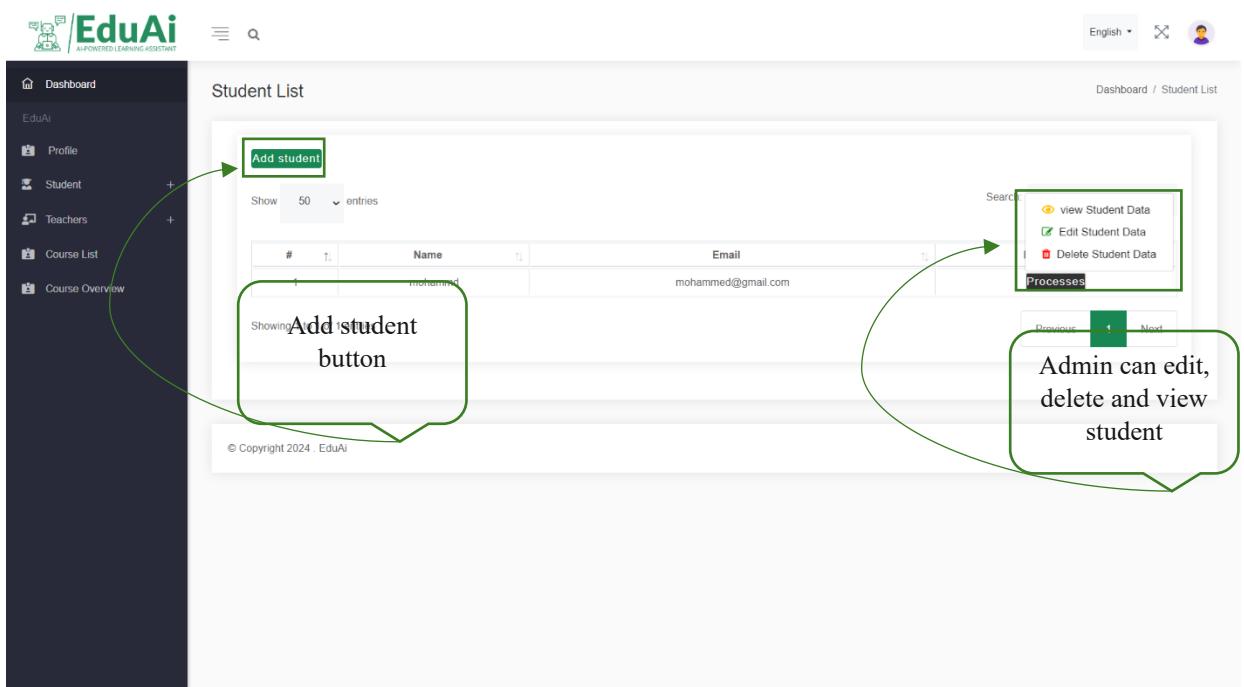


Figure 180: User guide 12

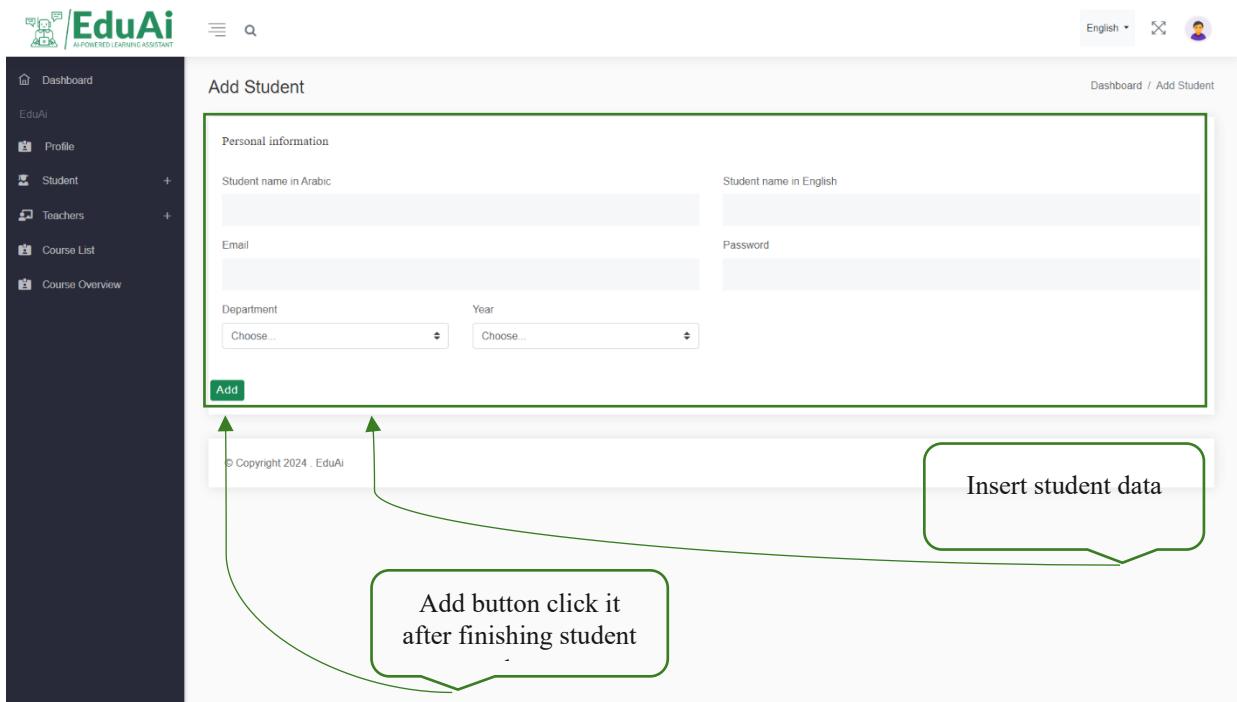


Figure 181: User guide 13

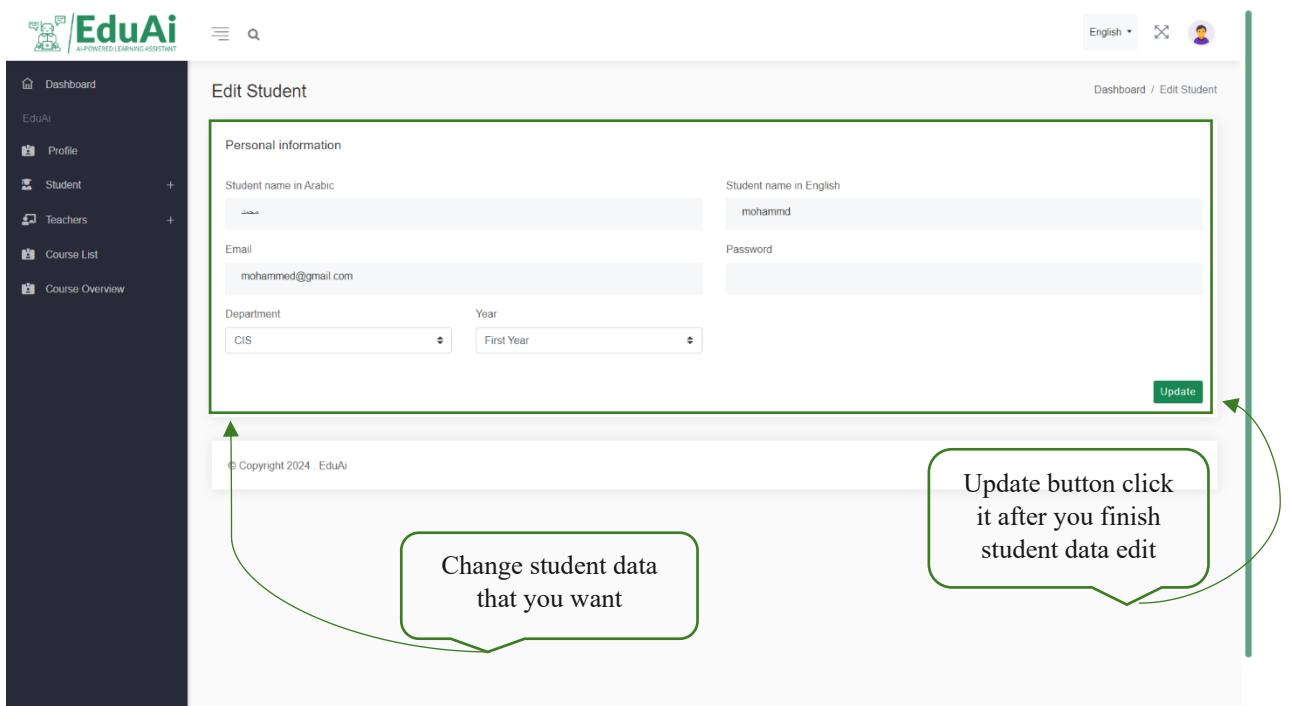


Figure 182: User guide 14

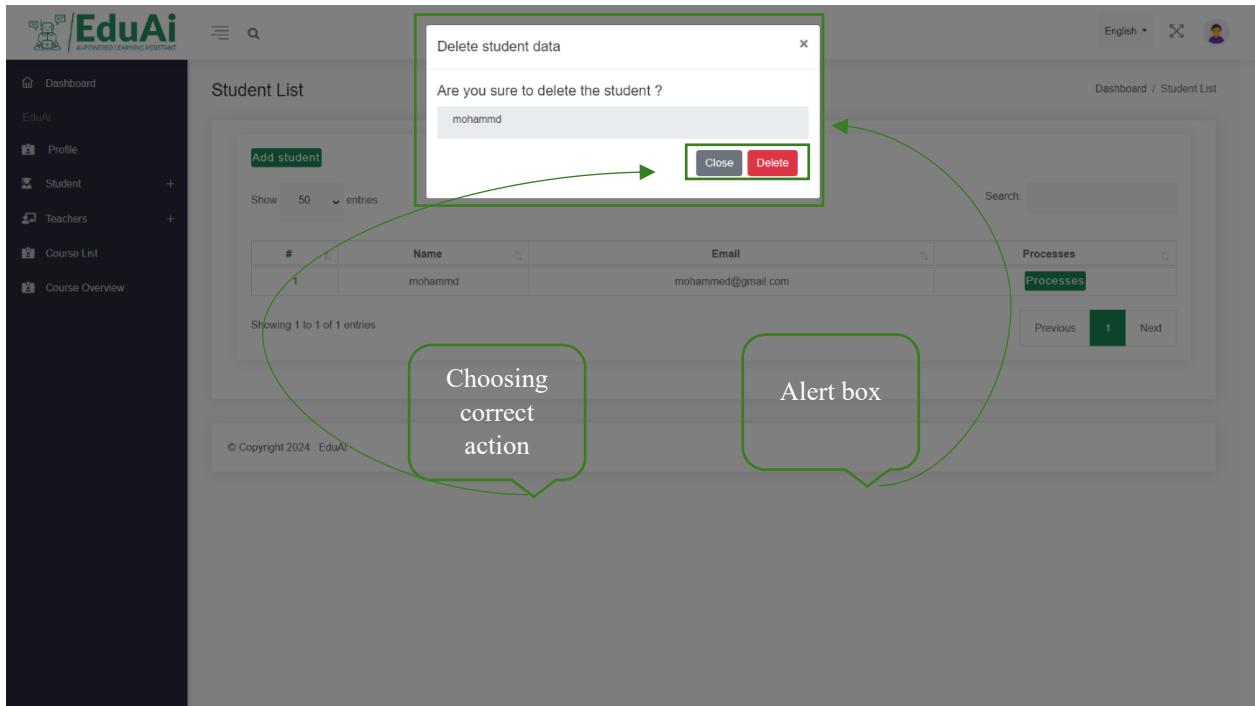


Figure 183: User guide 15

The screenshot shows the 'Instructor List' page of the EduAi application. The left sidebar has a dark theme with white icons and text, listing 'Dashboard', 'Profile', 'Student', 'Teachers', 'Course List', and 'Course Overview'. A green callout points from the 'Teachers' icon to the 'Add Instructor' button at the top of the main content area. The main content area has a light gray background and features a table with one entry. The table columns are '#', 'Name Instructor', and 'Joining Date'. The single row contains the value '1' under '#', 'Adnan Samer' under 'Name Instructor', and '2020-03-12' under 'Joining Date'. Below the table, a message says 'Showing 1 to 1 of 1 entries'. At the bottom right of the table, there is a copyright notice: '© Copyright 2024 EduAi'. To the right of the table is a 'Processes' section with two buttons: 'Edit Instructor data' (with a green checkmark) and 'Delete Instructor data' (with a red minus sign). A green callout points from the 'Edit Instructor data' button to a text box containing the text 'Admin can edit, delete instructor'. The top right corner of the page shows 'English' and a user profile icon.

Figure 184: User guide 16

The screenshot shows the EduAi platform's user interface. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, Student, Teachers, Course List, and Course Overview. The main area is titled "Add Instructor". It contains several input fields: Email, Password, Name_ar, Name_en, Joining_Date, and Address. At the bottom left is a green "Add" button. Two green callouts provide instructions: one pointing to the "Add" button with the text "Add button click it after finishing instructor data", and another pointing to the right side of the form with the text "Insert instructor data". The top right corner of the main window shows "English", a search icon, and a user profile icon.

Figure 185: User guide 17

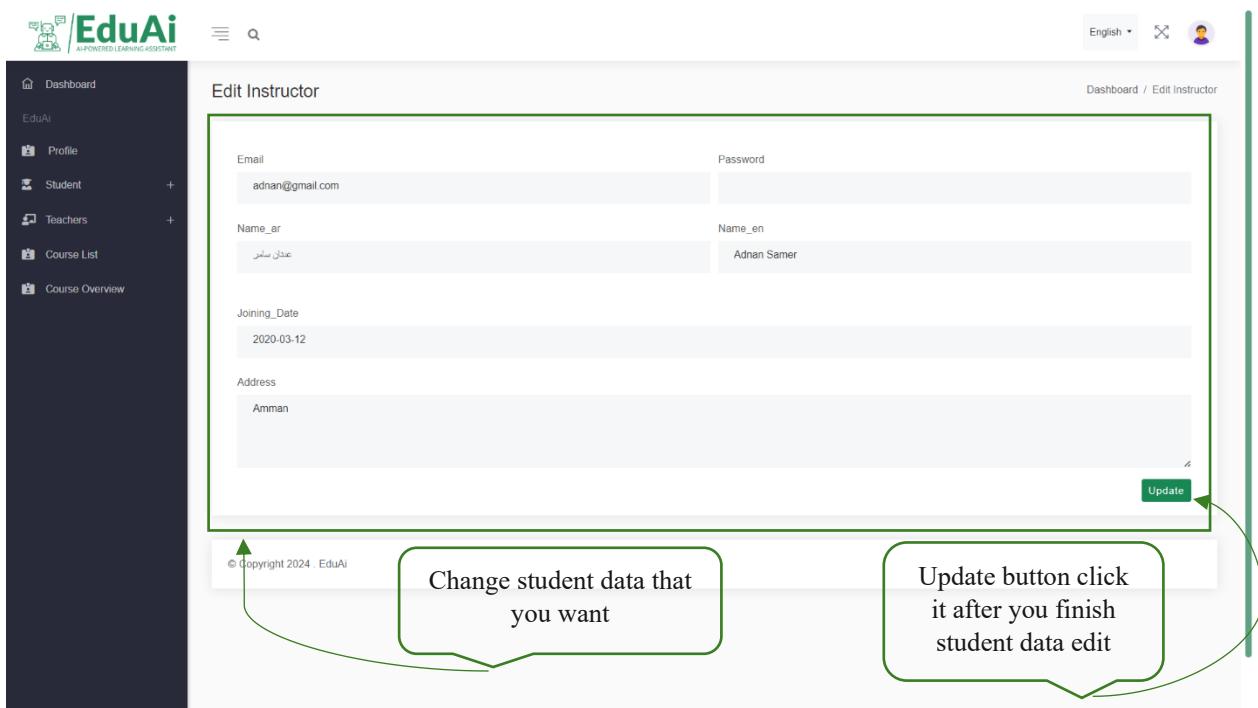


Figure 186: User guide 18

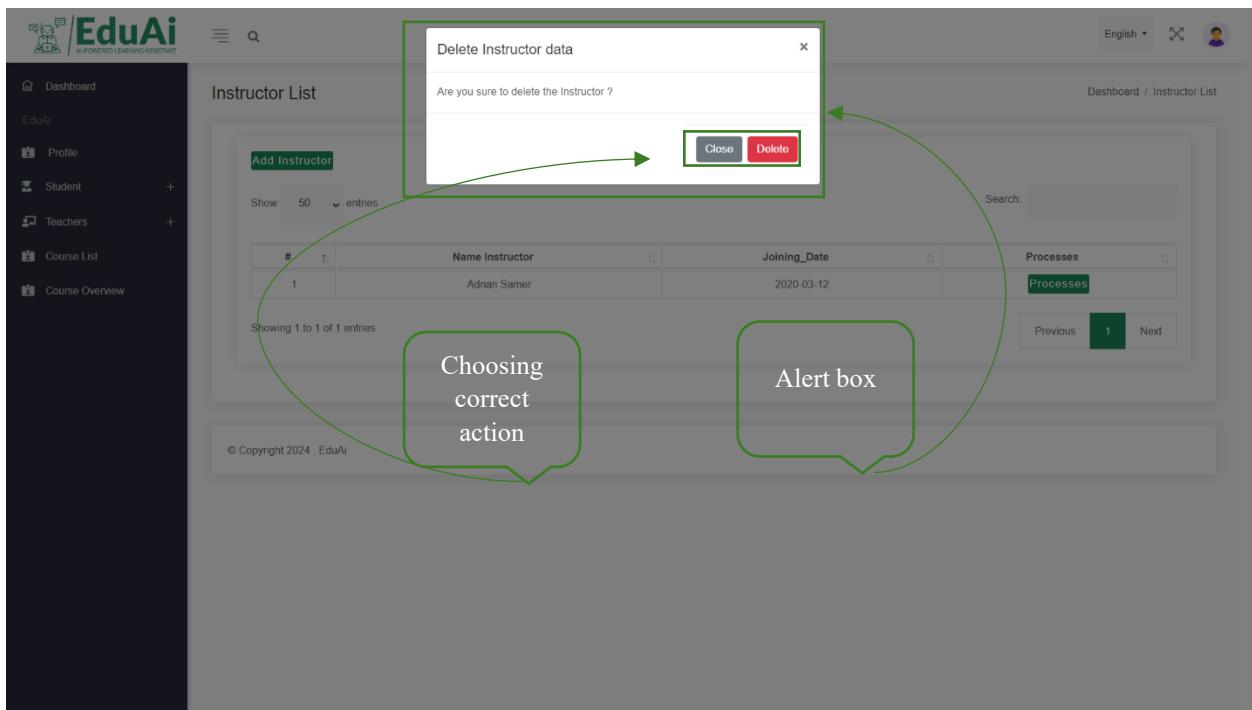


Figure 187: User guide 19

The screenshot shows the EduAi platform's Courses List page. The left sidebar includes links for Dashboard, Profile, Student, Teachers, Course List, and Course Overview. The main content area displays a table with one entry: Course Name (java), Department(s) (CIS, CS), and a Process column containing a red 'Published' button with a white icon. A green callout box labeled 'Delete and (publish / unpublish) button' points to this button. A red box labeled 'Unpublished' is shown at the top right of the screen.

#	Course Name	Department(s)	Process
1	java	CIS, CS	Published

Showing 1 to 1 of 1 entries

© Copyright 2024. EduAi

Unpublished

Delete and (publish / unpublish) button

Figure 188: User guide 20

The screenshot shows the EduAi platform's user interface. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, and Courses. The main content area has a header with a search bar and language selection (English). Below the header is a breadcrumb navigation: Dashboard / Result List. The central part of the screen displays a table titled "Result List". The table has columns: #, Title, Quiz Score, User Score, User Name, and Date. A "Return" button is located at the bottom left of the table. At the bottom of the page is a copyright notice: © Copyright 2024 EduAi. A green callout box with a curved arrow points from the text "List of all student result in specific quiz" to the "Result List" table.

List of all student result in specific quiz

Figure 189: User guide 21

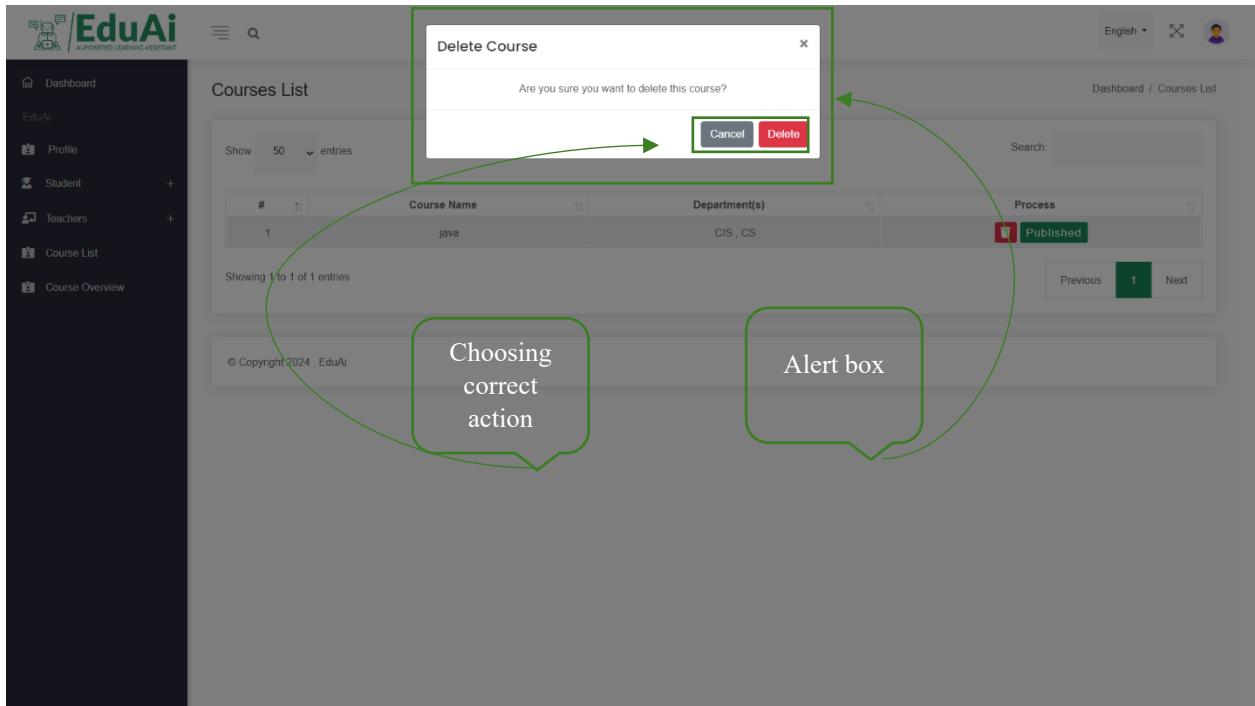


Figure 190: User guide 22

The screenshot shows the EduAi platform's Course Overview page. On the left is a dark sidebar with navigation links: Dashboard, Profile, Student, Teachers, Course List, and Course Overview. The main content area has a header "Course Overview" and a sub-header "Dashboard / Course Overview". It features a table with one entry: "Course Name" is "java", "Students" is "0", and "Students who completed the course" is "0". Below the table, it says "Showing 1 to 1 of 1 entries". At the bottom is a copyright notice: "© Copyright 2024 EduAi". A green callout box highlights the search bar at the top right, which contains the placeholder "Search:".

Search:

Search box by name or id

Figure 191: User guide 23

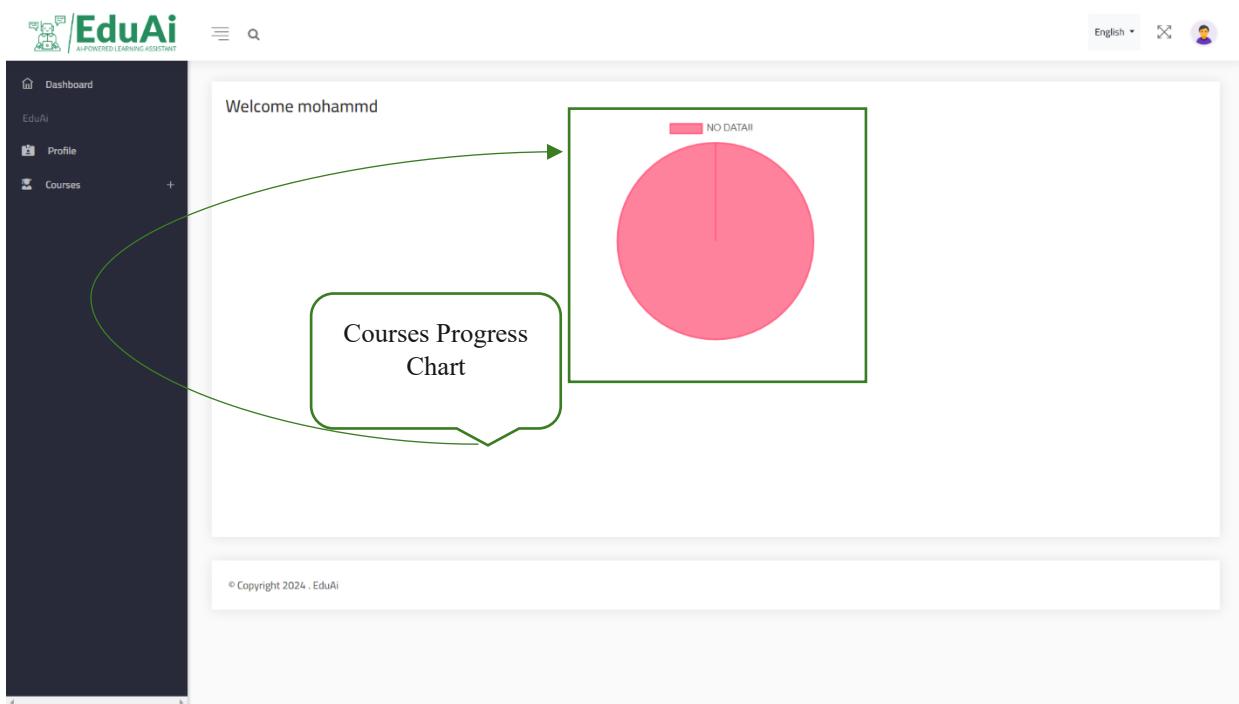


Figure 192: User guide 24

The screenshot shows the EduAi platform's Courses List page. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, Courses, and a plus sign. The main area has a header with a search bar and user profile icons. Below the header, the title "Courses List" is displayed. There are three filter dropdowns: "Filter by Course Title", "Filter by Department" (which is open, showing options like All Departments, CIS, CS, AI, CVS, BIT, DS), and "Filter by Year" (set to All Years). The main content area shows a course card for "Java", featuring its logo, a brief description ("java is programming language"), its year (2), and departments (CIS). Buttons for "Show More", "Show course", and "Enroll" are present. A green callout bubble with a curved arrow points to the "Filter by Department" dropdown, containing the text "Filter courses by department using drop list".

Figure 193: User guide 25

The screenshot shows the EduAi platform's Courses List page. On the left, a sidebar navigation includes Dashboard, EduAi, Profile, and Courses. The Courses section is expanded, showing a list of courses. The Java course is highlighted. Below the course list, there are three green callouts with arrows pointing to specific UI elements:

- A callout points to the "Show course" button with the text "Filter courses by year using drop list".
- A callout points to the "Enroll" button with the text "Filter courses by year using drop list".
- A callout points to the "All Years" dropdown menu with the text "Filter courses by year using drop list". The dropdown menu is open, showing options: All Years, All Years, First Year, Second Year, Third Year, and Fourth Year.

At the bottom of the page, there is a copyright notice: © Copyright 2024 .EduAi

Figure 194: User guide 26

The screenshot shows the EduAi platform's 'My Courses' section. On the left, a sidebar menu includes 'Dashboard', 'EduAi', 'Profile', and 'Courses' (with sub-options 'Courses List', 'My Courses', 'Progress Courses List', and 'Finished Courses List'). The main area displays a course card for 'Java'. The card features a Java logo at the top, followed by the word 'Java'. Below this, there is a 'Description:' field containing 'java is programming language', a 'Year:' field showing '2', a 'Departments:' field listing 'CIS', and a 'Show More' button. At the bottom of the card are two buttons: 'Go to course' (green) and 'Remove' (red). A green callout bubble points to the 'Remove' button with the text 'Remove or enter specific course content'. Another green callout bubble points to the 'Go to course' button with the text 'Student courses details after enrolling to the course'.

Figure 195: User guide 27

Remove or enter
specific course content

The screenshot shows the EduAi platform's 'Progress Courses List' page. The left sidebar includes links for Dashboard, EduAi, Profile, and Courses. The main content area displays a list of courses with a search bar at the top. A specific course, 'Java', is highlighted with a green callout box containing the following details:

- Java**
- Description: java is programming language
- Year: 2
- Departments:
 - CIS
- Show More
- Progress: 50.00%
- Buttons: Go to course, Remove

A green callout box labeled 'Student progress courses details' points to the 'Java' course card.

Figure 196: User guide 28

The screenshot shows the EduAi platform's 'Finished Courses List' page. The left sidebar includes links for Dashboard, EduAi, Profile, and Courses. The main content area displays a list of finished courses. One course, 'Java', is highlighted with a green callout box containing the text 'Student finished courses details'. The Java course card includes a logo, the title 'Java', a description ('java is programming language'), a year ('2'), departments ('CIS'), a progress bar ('100.00%'), and buttons for 'Go to course' and 'Remove'.

Figure 197: User guide 29



Figure 198: User guide 30

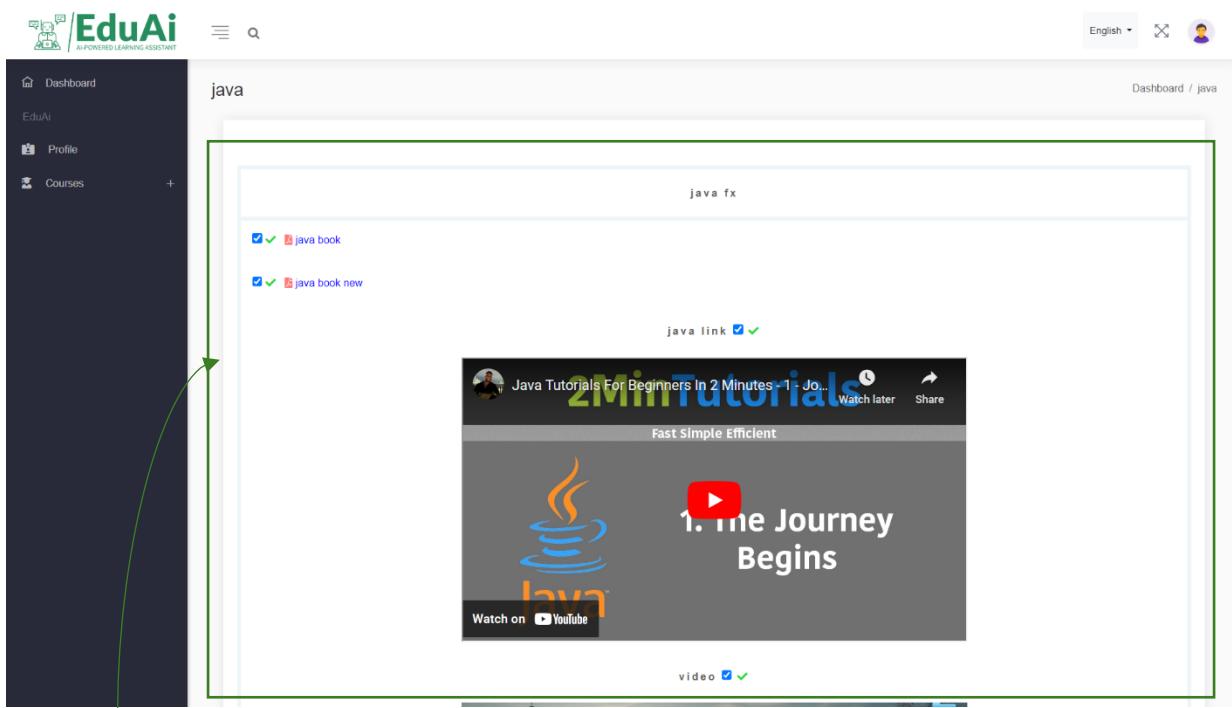


Figure 199: User guide 31

After click (go to course) button
display all section name with item
related to the course

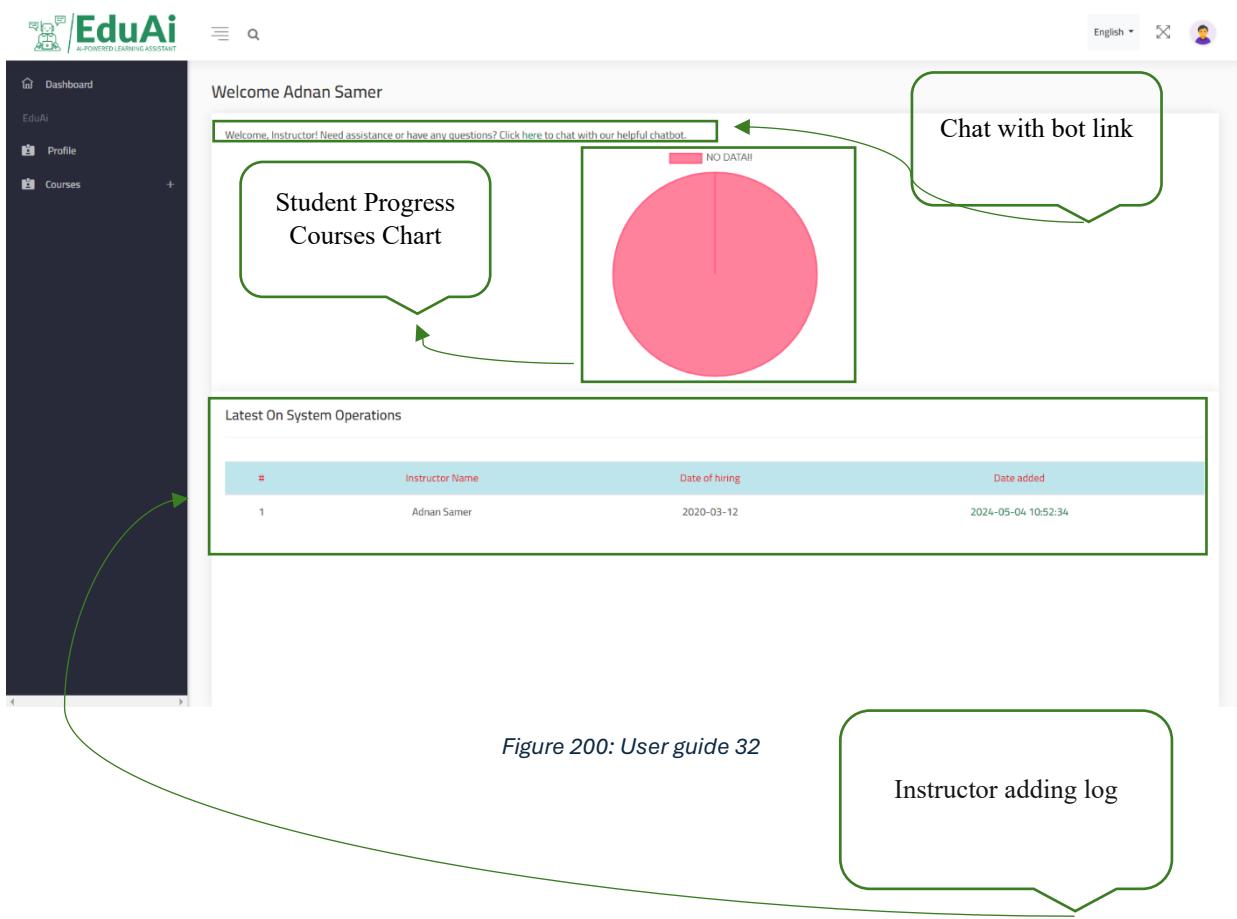


Figure 200: User guide 32

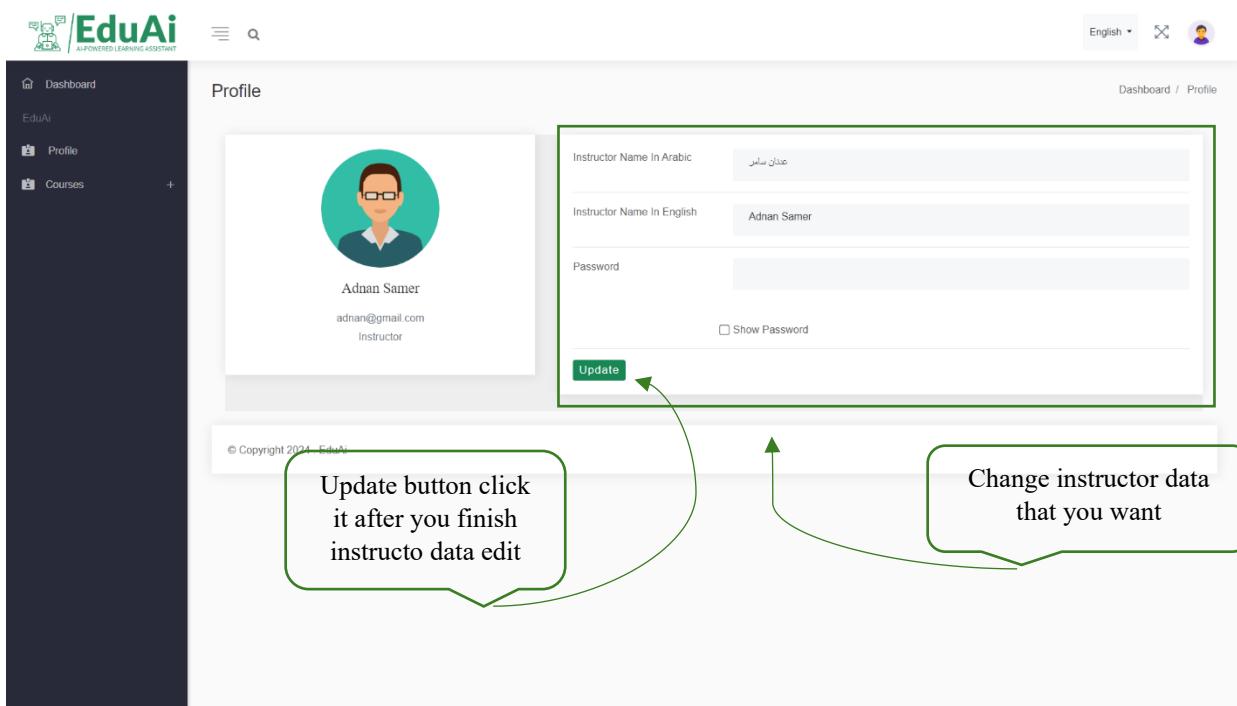


Figure 201: User guide 33

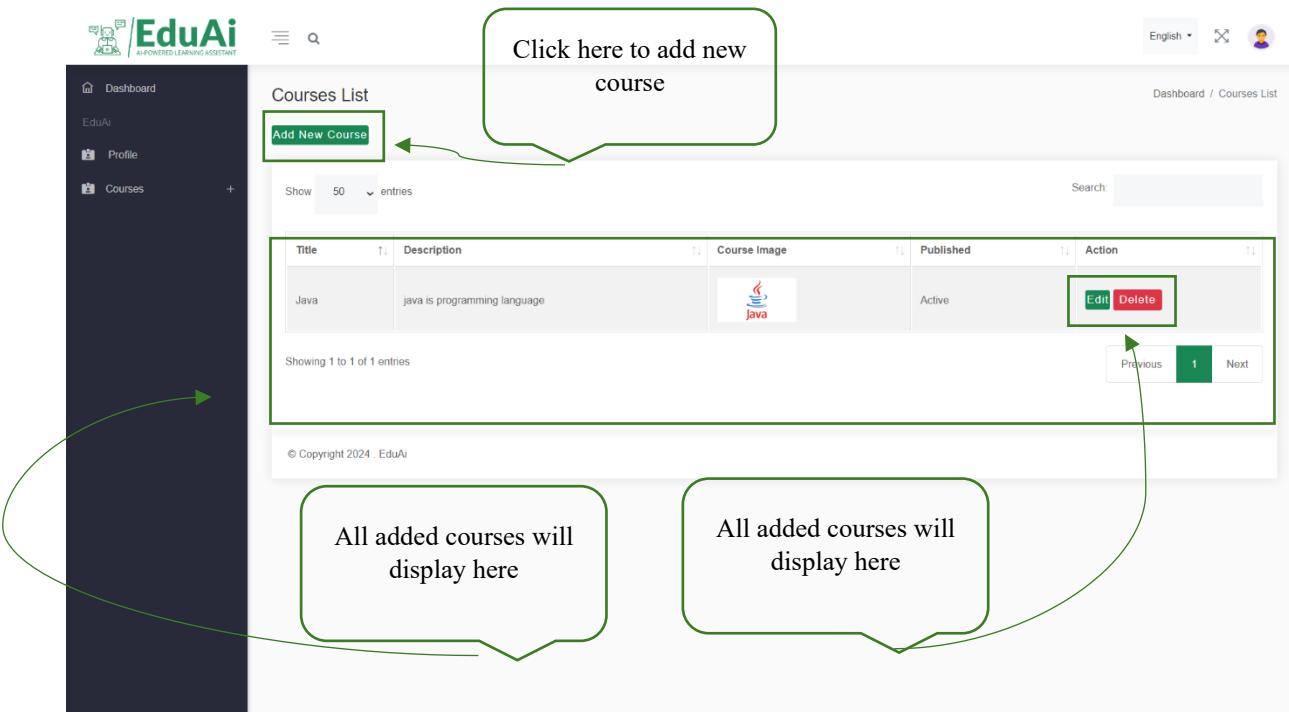


Figure 202: User guide 34

EduAi
AI-POWERED LEARNING ASSISTANT

Dashboard / Add New Course

Title

Description

Course Image :
 No file chosen

Department

CIS
CS
AI
...

Years

Published
Active

© Copyright 2024 - EduAi

Add

Figure 203: User guide 35

Fill all required course data to add one

Add course button click it after you finished insert course data

Edit Course

Dashboard / Edit Course

Title: Java

Published: Active

Course Image: Choose file No file chosen

Department: Choose from the list...
CIS
CS
AI
...

Years: Second Year

Description: java is programming language

Fill all required course data you want to edit

update course button
click it after you finished editing course data

Update

Figure 204: User guide 36

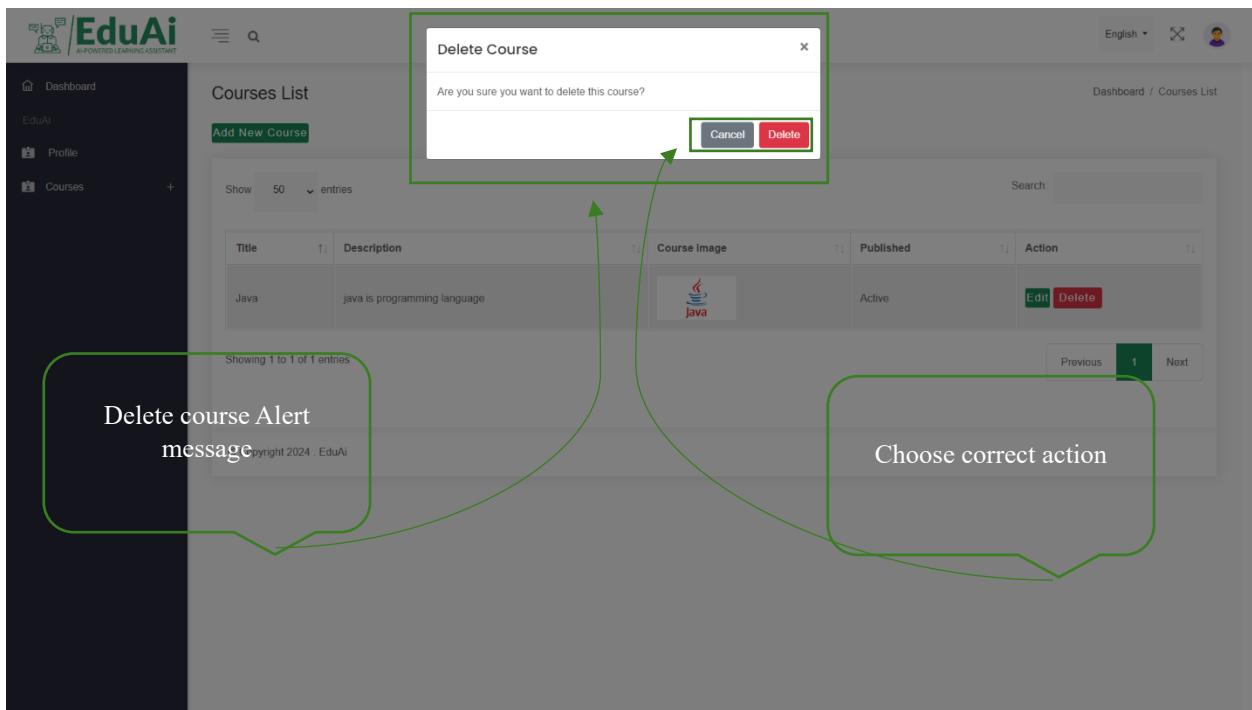


Figure 205: User guide 37

The screenshot shows the EduAi platform's Courses List page. On the left is a dark sidebar with navigation links: Dashboard, Profile, Courses, and a plus sign. The main area has a header with the EduAi logo, a search bar, and language settings. Below the header is a breadcrumb navigation: Dashboard / Courses List. The main content area is titled "Courses List". It features a card for a course titled "Java", which includes a Java logo, the word "Java", a description ("java is programming language"), a year ("Year: 2"), departments ("CIS", "CS", "..."), and a "View Course" button. A green callout bubble with a curved arrow points to the course card, containing the text "All instructor courses will display here". At the bottom of the page is a footer with the copyright notice "© Copyright 2024 EduAi".

Figure 206: User guide 38

Different operations on section (add/edit section)

Add or edit item and quiz at specific section

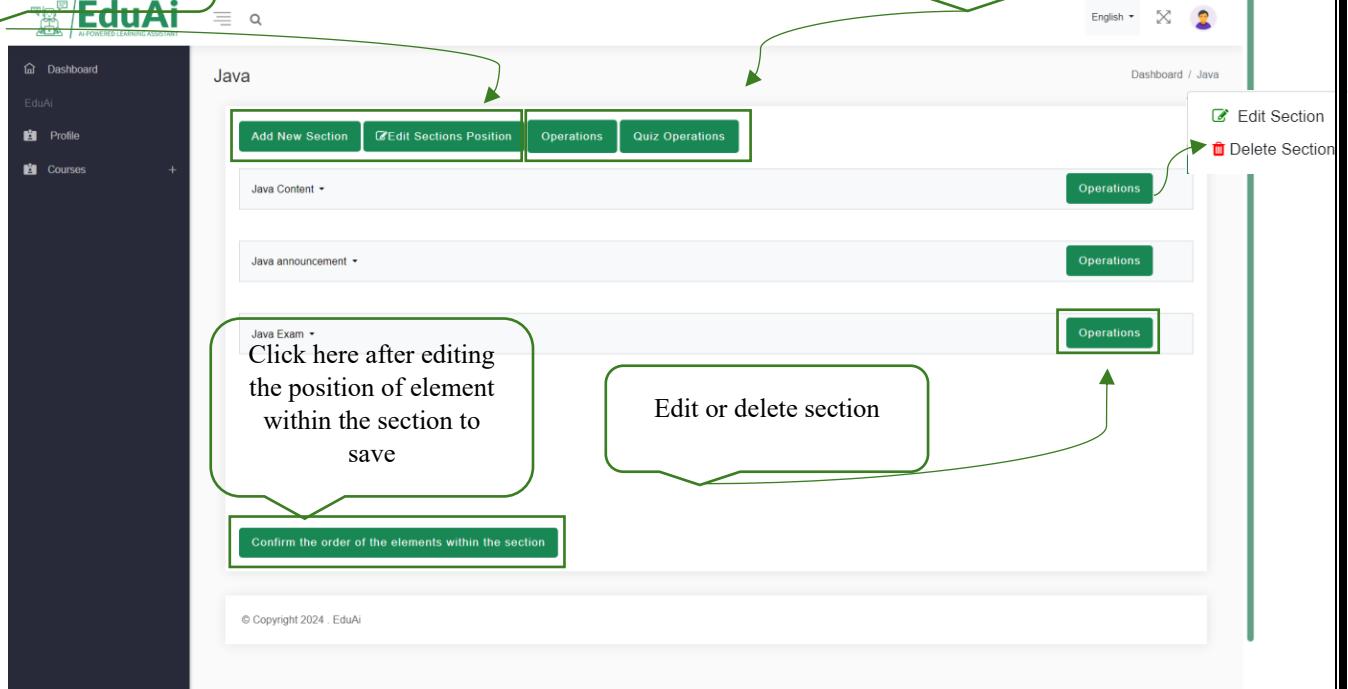


Figure 207: User guide 39

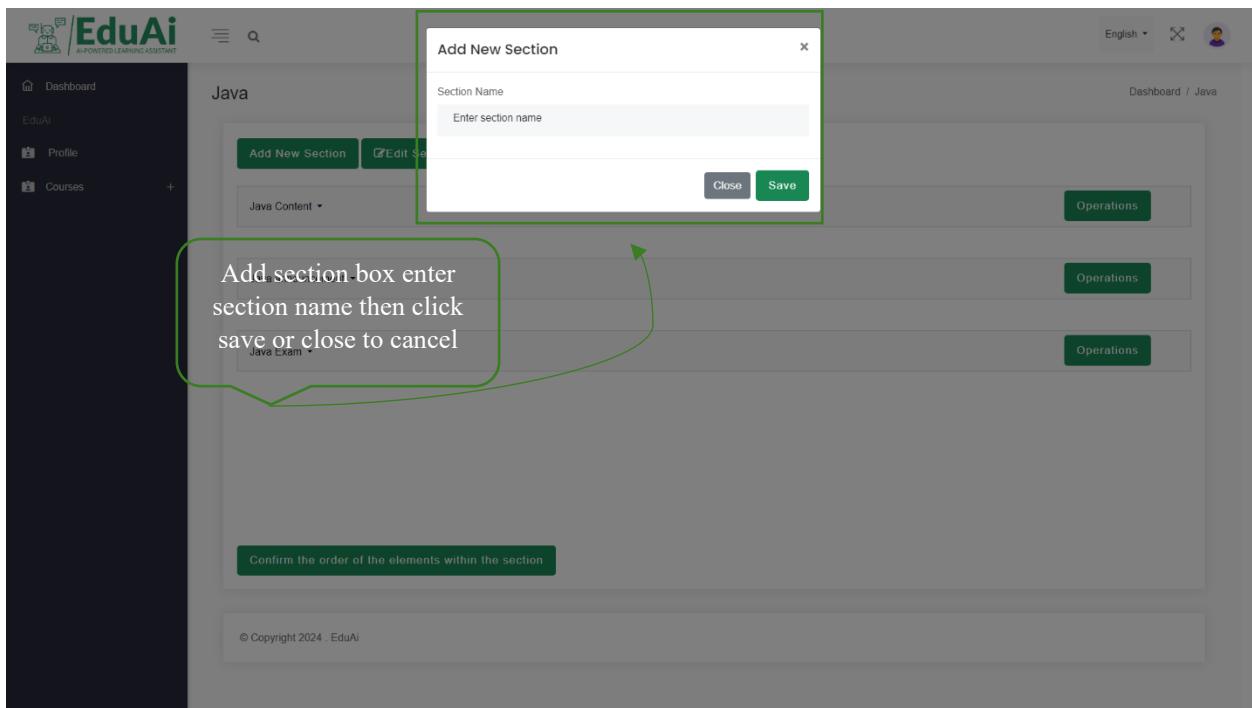


Figure 208: User guide 40

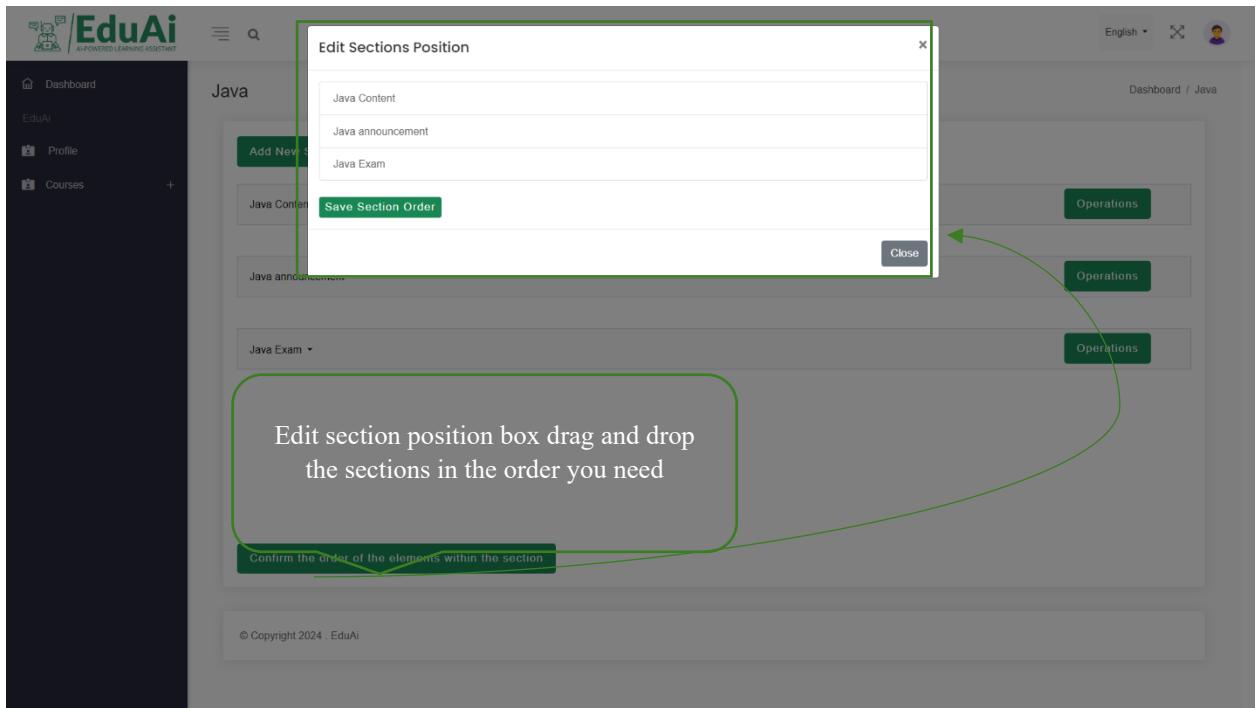


Figure 209: User guide 41

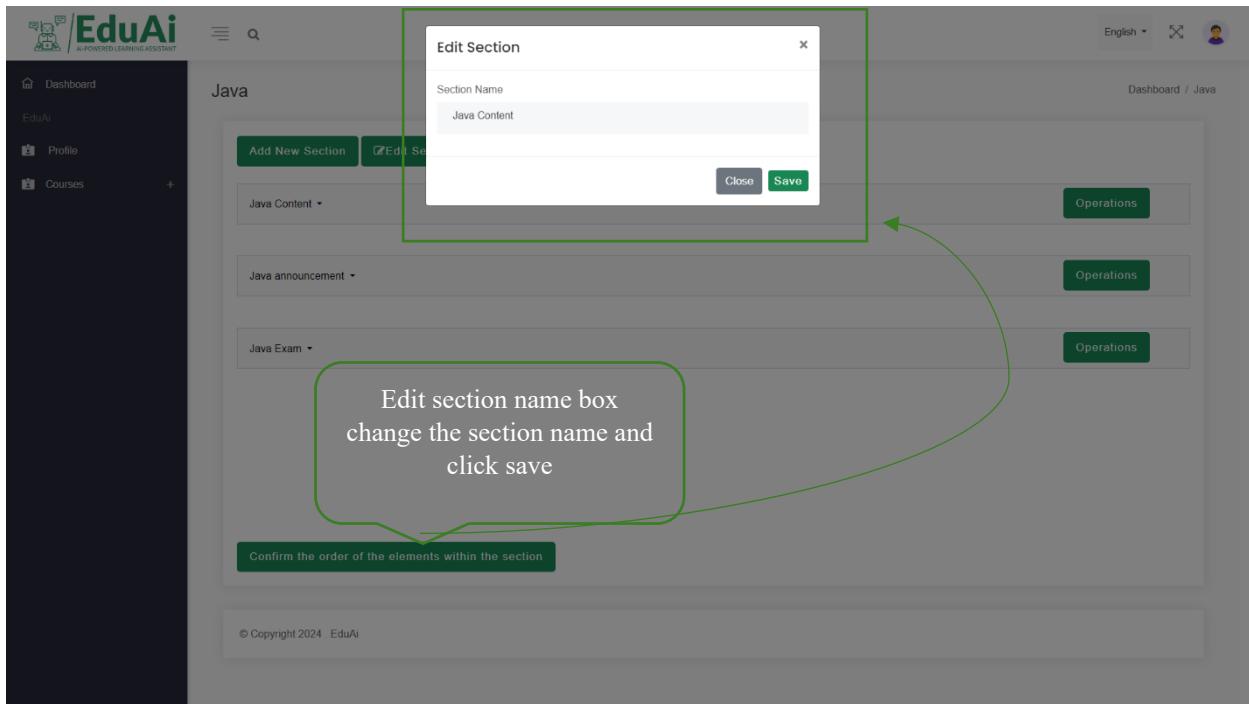


Figure 210: User guide 42

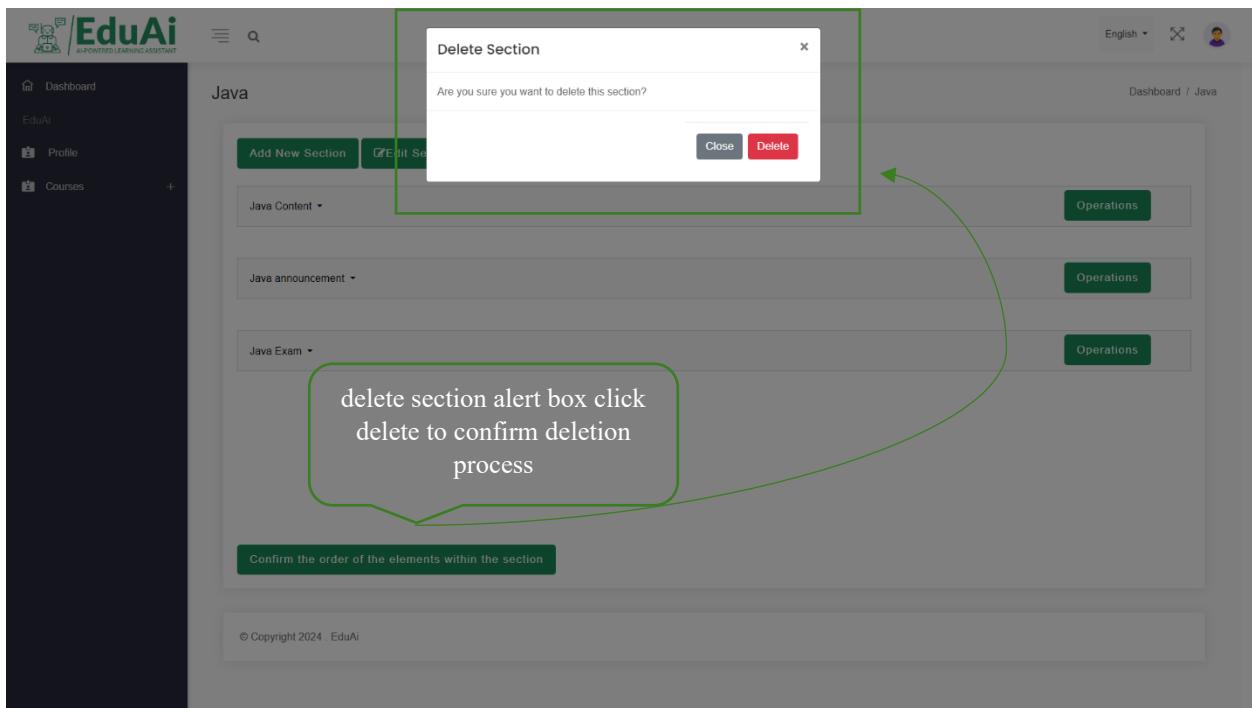


Figure 211:: User guide 43

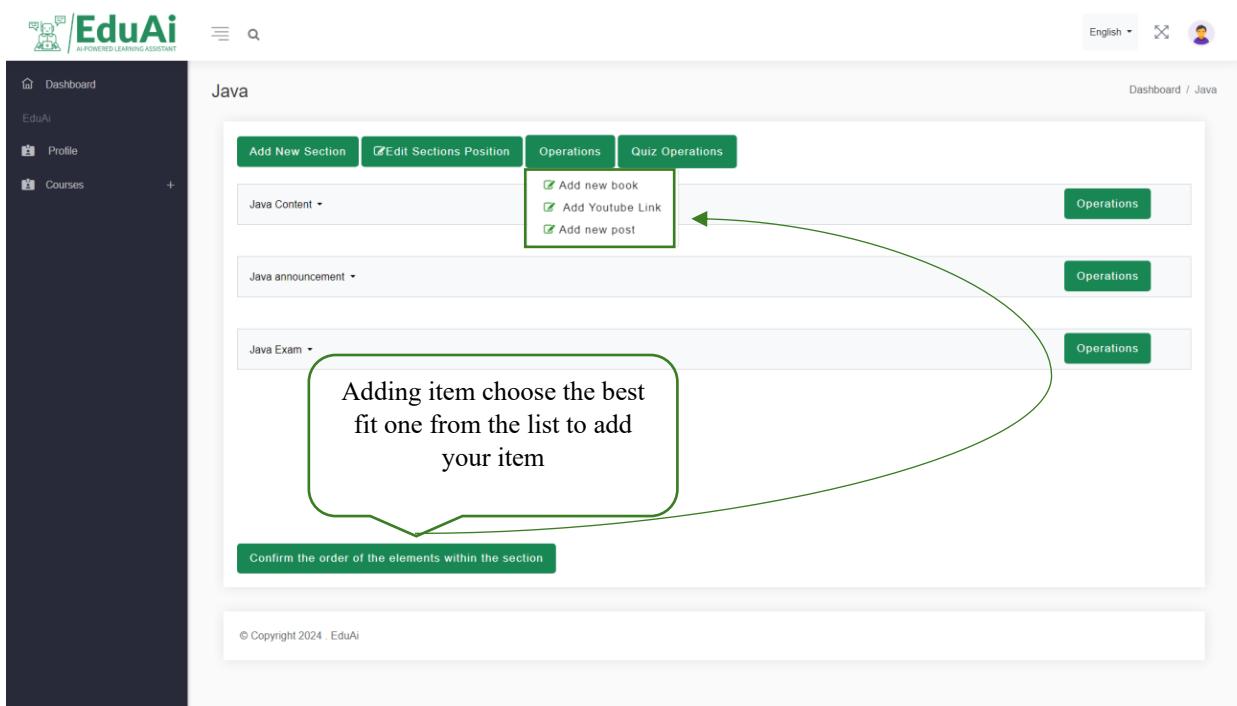


Figure 212: User guide 44

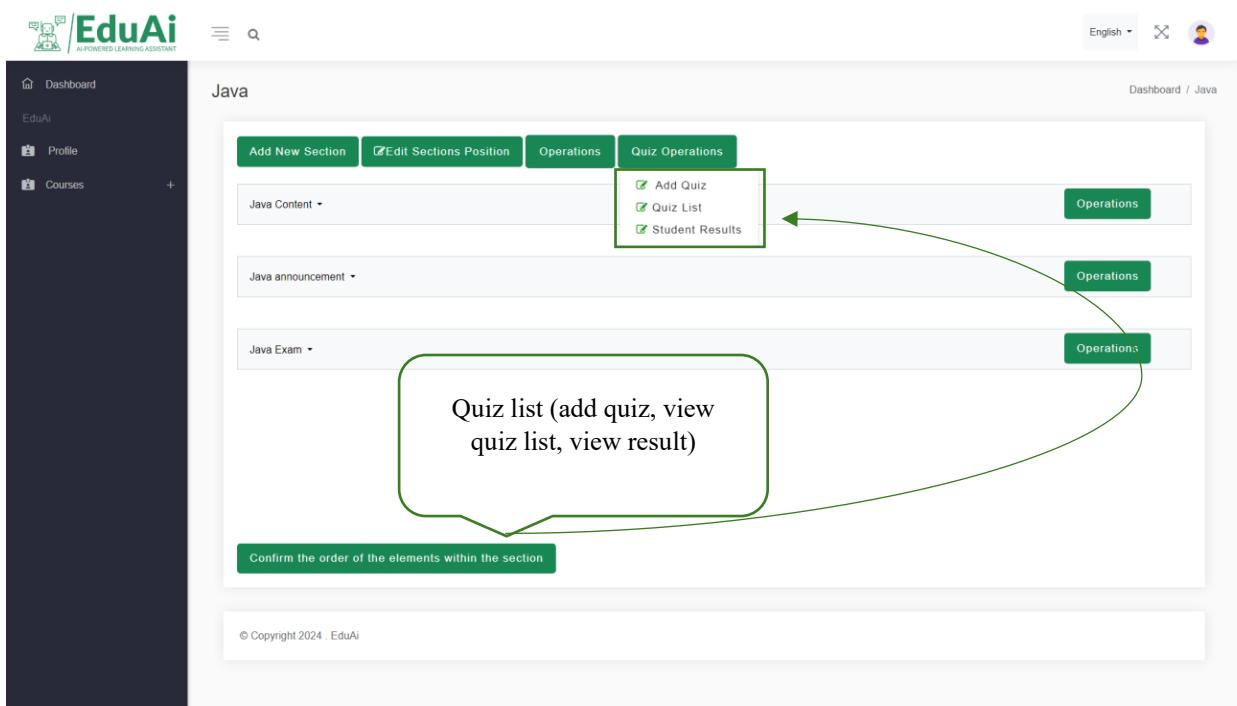


Figure 213: User guide 45

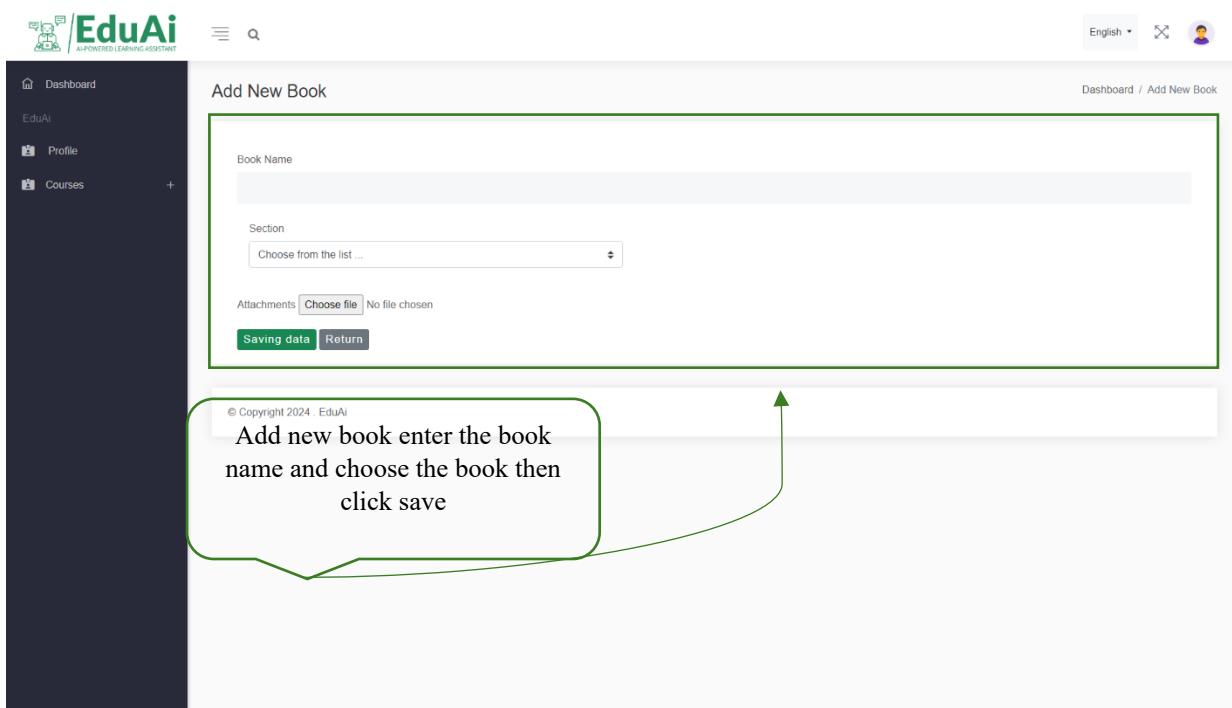


Figure 214: User guide 46

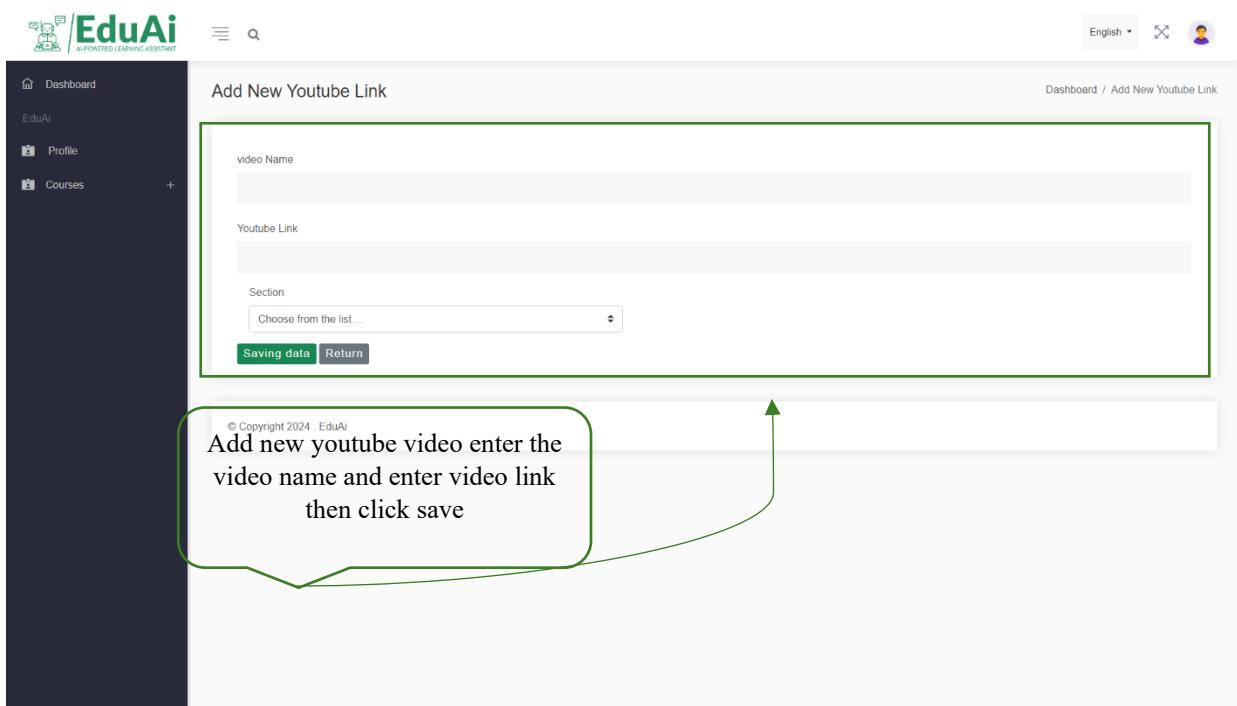


Figure 215: User guide 47

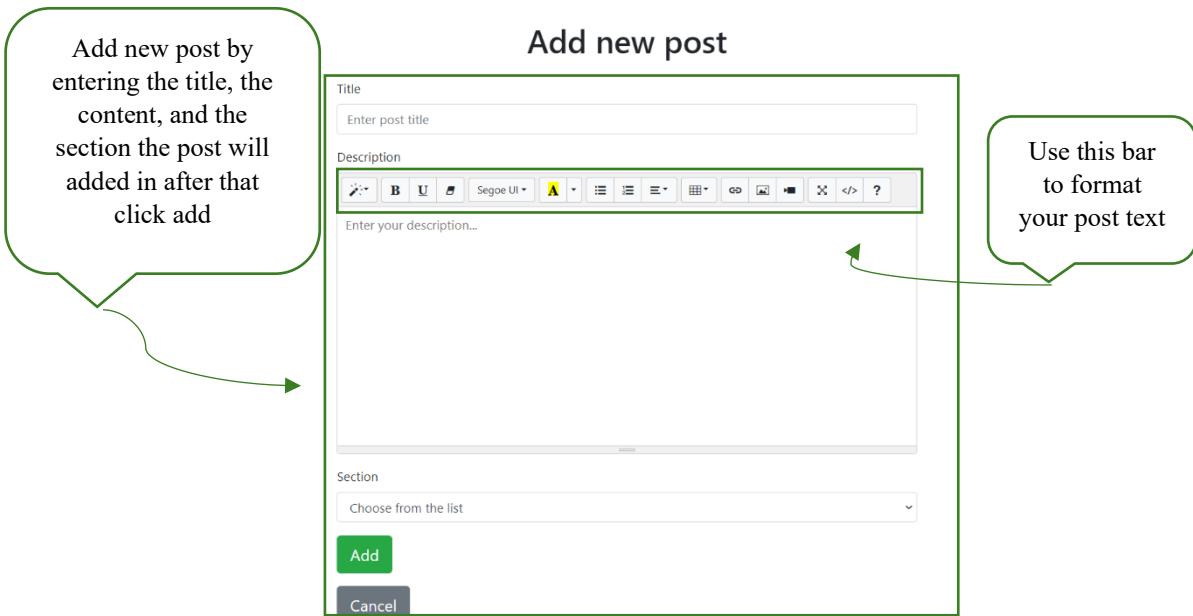


Figure 216: User guide 48

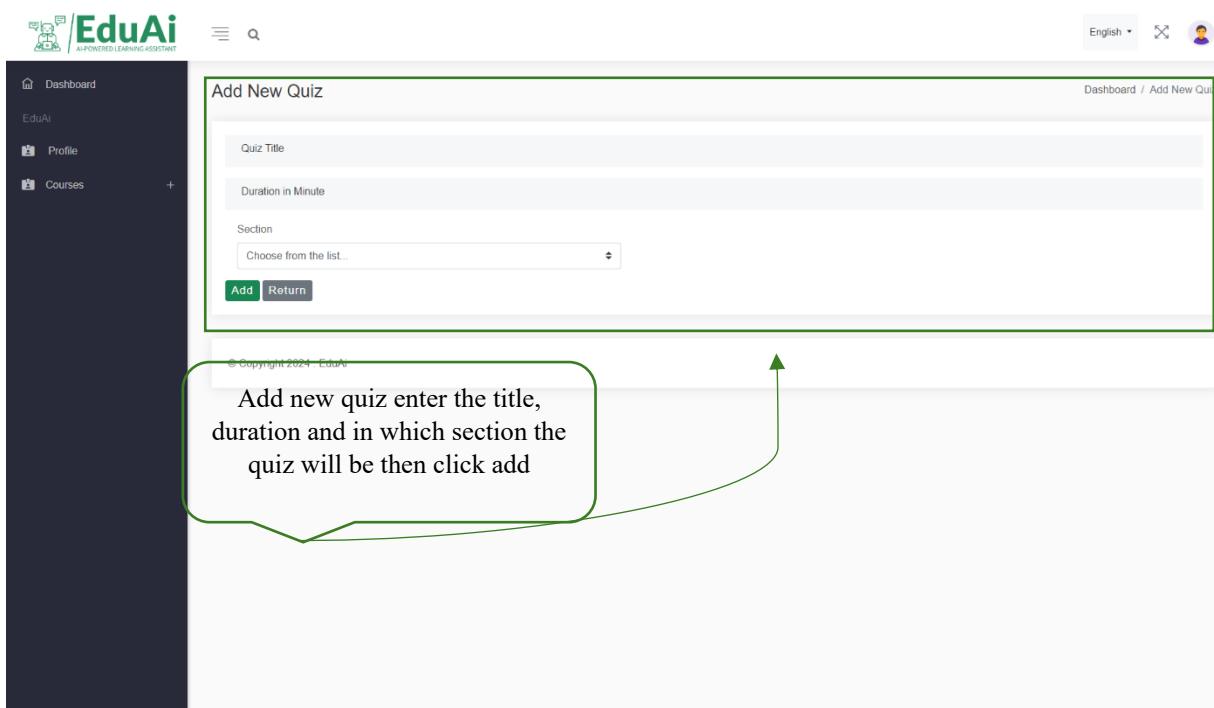


Figure 217: User guide 49

The screenshot shows the EduAi platform's user interface. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, Courses, and a plus sign. The main area has a light background with a top navigation bar featuring a search icon, a language dropdown set to English, and a user profile icon. Below this is a section titled "Quiz List" containing a table with one row. The table has columns for "#", "Title", and "Duration". The first row shows "# 1", "Java Fundamentals Quiz", and "60 minutes". A "Return" button is at the bottom of this section. At the very bottom is a copyright notice: "© Copyright 2024 . EduAi". A green callout bubble with an arrow points from the text "Quiz list here you will see all your quizzes with its duration" to the "Quiz List" section.

#	Title	Duration
1	Java Fundamentals Quiz	60 minutes

© Copyright 2024 . EduAi

Figure 218: User guide 50

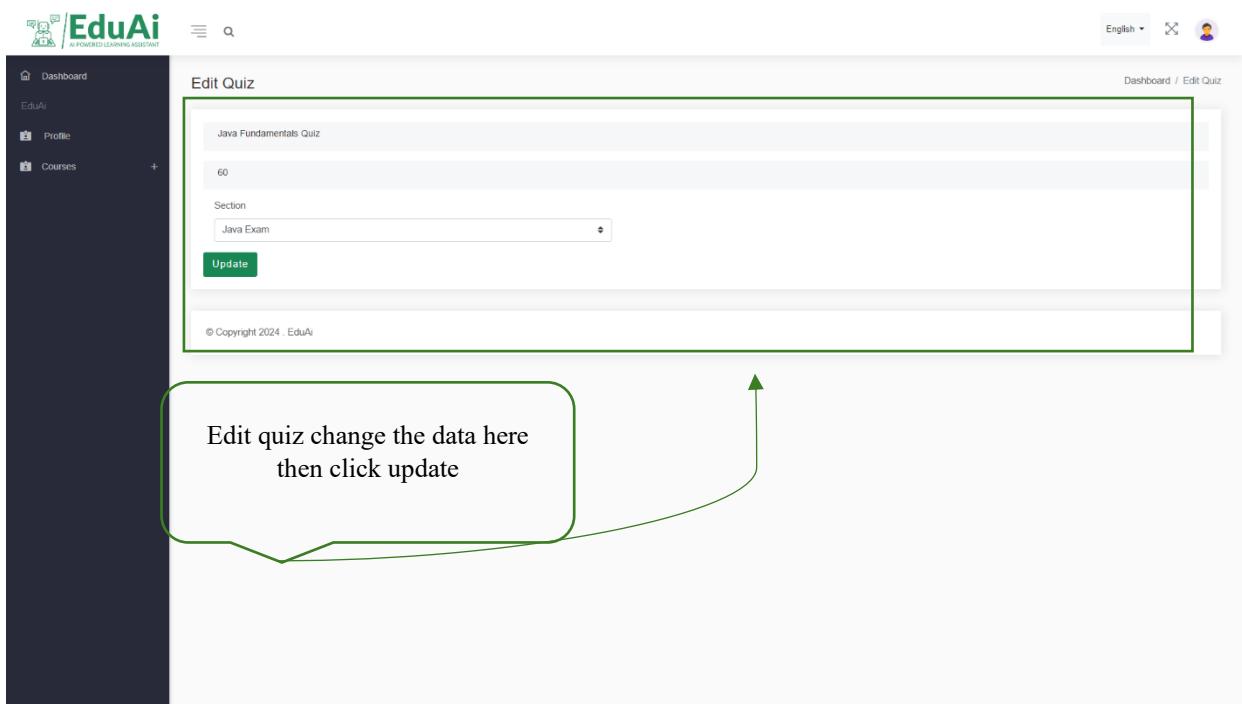


Figure 219: User guide 51

The screenshot displays the EduAi platform interface. On the left is a dark sidebar with icons for Dashboard, EduAi, Profile, and Courses. The main area shows a 'Java Content' section with a 'Java Book' item. This item has a 'Download' button, an 'Update' button, and a 'Delete' button. Below this is a 'Java Video' item, which is a thumbnail for a YouTube video titled 'Java Tutorials For Beginners In 2 Minutes - 1 - Jo...'. The video thumbnail includes the '2MinTutorials' channel logo, a play button, and the text '1. The Journey Begins'. At the bottom of the page are two more sections: 'Java announcement' and 'Java Exam', each with its own 'Delete' button.

Here you can see book item within a section with delete, update and download button

Java Content

Java Book

Download Update Delete

Java Video

Java Tutorials For Beginners In 2 Minutes - 1 - Jo... 2MinTutorials Watch later Share

Fast Simple Efficient

Java

1. The Journey Begins

Java announcement

Java Exam

Delete

Operations

Operations

Operations

Figure 220: User guide 52

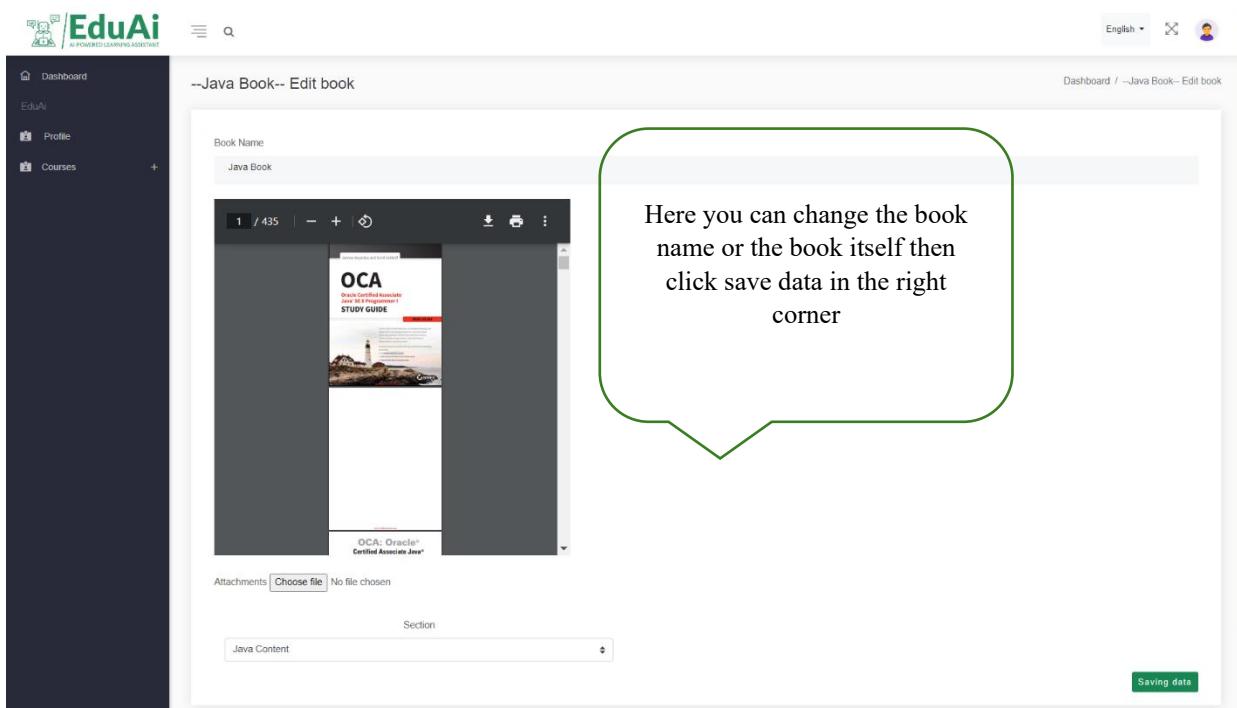


Figure 221: User guide 53

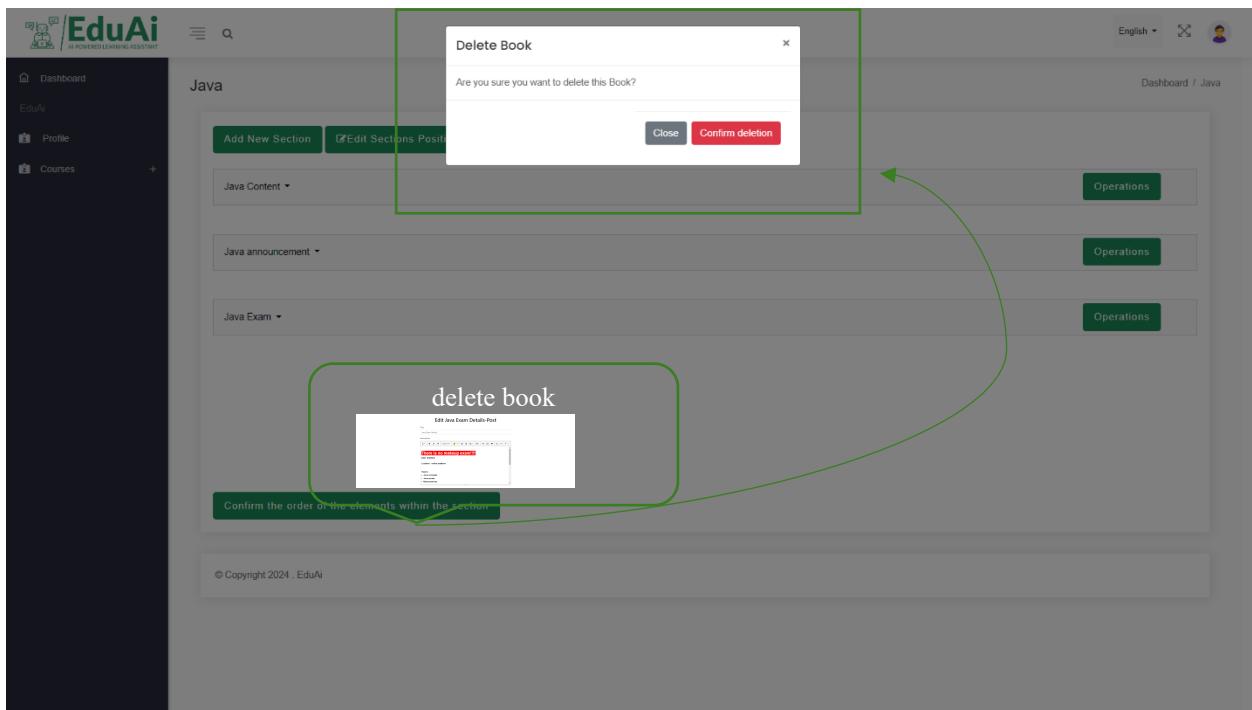


Figure 222: User guide 54

Edit Java Exam Details-Post

Title
Java Exam Details

Description

There is no makeup exam!!!!

Date: 5/5/2024

Location: online platform

Topics:

- Java concepts
- Java syntax
- Best practices

Section
Java announcement

Update Cancel

Edit post change the post content here then click update for confirmation

Figure 223: User guide 55

Add question for quiz by entering the question and the options and choosing the correct answer

Here list of all questions you have been added so far

#	Question	A	B	C	D	Correct	Operations
1	What is the correct syntax for declaring a variable in Java?	variable x;	x = 5;	int x;	var x;	option_c	Edit Delete
2	Which of the following is NOT a primitive data type in Java?	boolean	string	float	int	option_b	Edit Delete
3	What does the "public static void main(String[] args)" method indicate in a Java program?	It is a constructor method	It is used to declare variables	It is a method to print output	It is the starting point of the program	option_d	Edit Delete
4	What does the "this" keyword refer to in Java?	All class objects	Next class object	Previous class object	Current class object	option_d	Edit Delete
5	What is the purpose of the "static" keyword in Java?	To prevent inheritance	To allow access to a method or variable without creating an instance of the class	To declare a variable as constant	To make a method non-static	option_b	Edit Delete

Figure 224: User guide 56

What is the correct syntax for declaring a variable in Java?

variable x;

x = 5;

int x;

var x;

C

Update

#	Question	A	B	C	D	Correct
1	What is the correct syntax for declaring a variable in Java?	variable x;	x = 5;	int x;	var x;	option_c
2	Which of the following is NOT a primitive data type in Java?	boolean	string	float	int	option_b
3	What does the "public static void main(String[] args)" method indicate in a Java program?	It is a constructor method	It is used to declare variables	It is a method to print output	It is the starting point of the program	option_d
4	What does the "this" keyword refer to in Java?	All class objects	Next class object	Previous class object	Current class object	option_d
5	What is the purpose of the "static" keyword in Java?	To prevent inheritance	To allow access to a method or variable without creating an instance of the class	To declare a variable as constant	To make a method non-static	option_b

© Copyright 2024 - EduAI

Figure 225: User guide 57

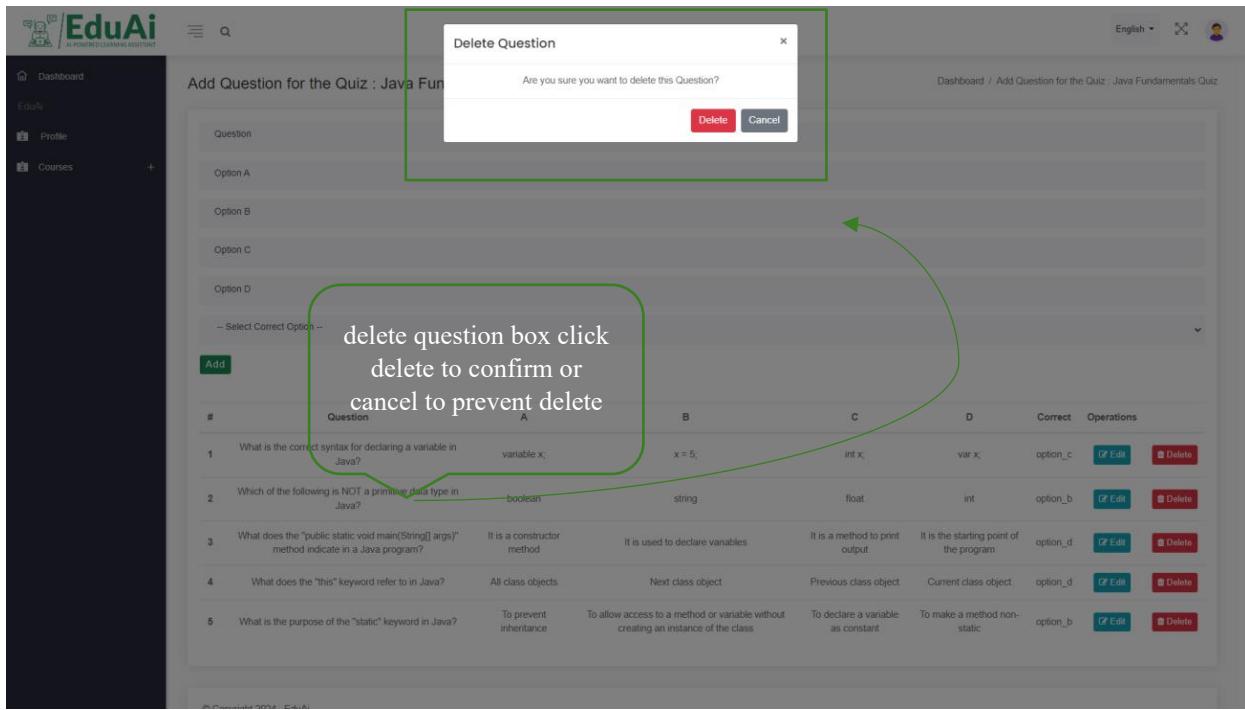


Figure 226: User guide 58

The screenshot shows the EduAi platform's user interface. On the left is a dark sidebar with navigation links: Dashboard, EduAi, Profile, Courses, and a plus sign. The main area is titled "Result List" and displays a table of quiz results:

#	Title	Quiz Score	My Score	Date
1	Java Fundamentals Quiz	5	3	2024-05-04 14:36:26
2	Java Fundamentals Quiz	5		2024-05-04 15:05:32

A "Return" button is located at the bottom left of the table. At the bottom of the page is a copyright notice: "© Copyright 2024 EduAi".

An annotation highlights a green speech bubble with the text "Click here to start chatting" pointing to a green robot icon. Another annotation highlights a green rounded rectangle containing the text "Click here to display the your result" pointing to a "Show Result" button. A third annotation highlights a green rounded rectangle containing the text "AI Suggestions result" pointing to another "Show Result" button.

Figure 227: User guide 59

Chapter seven: Project conclusions and future work

7.1 Conclusions

- **What We Did:** We made an online learning tool called "EduAI" to help university students learn better.
- **How It Helped:** Our tool had cool stuff like a smart chatbot and tracking how students learn.
- **What People Said:** Students and teachers liked our tool and said it helped them learn more.

7.3 Strengths

- **Made Learning Personal:** Our tool made learning more personal, like having a learning partner just for you.
- **Had Cool Features:** We added useful things like the chatbot and tracking how you learn.
- **People Liked Using It:** Students had more fun learning with our tool and felt more involved.

7.3 Weaknesses

- **Some Tech Problems:** Sometimes our tool had problems working perfectly, like bugs or slow parts.
- **Not Everyone Used It:** Some students didn't use our tool much, so we want to figure out why.
- **Things We Can Improve:** We have identified areas where we can improve, such as making it easier to use or adding more helpful features, like allowing different types of questions in the quiz, not just multiple-choice.

7.4 Future Direction

- We plan to enhance EduAI by adding features such as reading and analyzing the material of the sections and giving suggestions, improving the chatbot's responses, and exploring virtual reality simulations for certain courses.
- In terms of expansion, we aim to partner with more universities and schools, customize EduAI for different educational settings, and translate it into multiple languages for wider accessibility.

References

1-Software Engineering 9th Edition by Ian Sommerville:

<https://engineering.futureuniversity.com/BOOKS%20FOR%20IT/Software-Engineering-9th-Edition-by-Ian-Sommerville.pdf>

2-ChatGPT API documentation:

<https://platform.openai.com/docs/api-reference/introduction>

3-Laravel documentation:

<https://laravel.com/docs/8.x/readme>