



Palestine Polytechnic University

College of Information Technology and Computer Engineering

KidTrack Web Application

Team members:

Meera Qunaibe

Nadeen Aljubeh

Raghad Manasrah

Supervisor name:

Nabil Arman

2022-2023

المخلص

يهدف المشروع في الدرجة الأولى إلى بناء تطبيق إلكتروني يساعد الأهل في التواصل مع المعلمين من أجل متابعة أطفالهم ومعرفة مدى تطور مستواهم التعليمي، كما يساعد الأهل في المشاركة في العملية التعليمية من خلال دمج ألعاب تساعد الطفل في مراجعة ما تم شرحه خلال الحصص الدراسية باستخدام تكنولوجيا (gamification) وهي نهج تعليمي يسعى إلى تحفيز الطلاب من خلال دمج ألعاب تعليمية تعزز من فهم الطالب والهدف منها هو تحقيق الاستمتاع خلال تلقي المعلومة و جذب اهتمام الطلاب مما يساعدهم في مواصلة التعلم. كما يمكن التطبيق من إتمام عملية التسجيل وتسليم الأوراق المطلوبة إلكترونيا دون الحاجة للذهاب إلى الروضة.

قام فريق العمل بدراسة النظام الحالي المستخدم في روضة الذكاء العقلي ، من خلال إجراء مقابلات مع مديرة الروضة المستهدفة والكادر التعليمي هناك. وتم بالنهاية عمل موقع إلكتروني حيث يساعد الأهالي على التواصل مع المعلمين وتتبع مستوى طفلهم التعليمي وسلوكه.

Abstract

The project's primary goal is to create a web application that enables parents and teachers to communicate so that parents may track their kids' progress and evaluate their intellectual growth. It also helps parents participate in the educational process by integrating games that help the child review what was explained during class lessons using gamification technology, which is an educational approach that seeks to motivate students by incorporating educational games that increase student understanding. The application also makes it possible to register without having to visit the kindergarten and submit the necessary paperwork remotely.

The work team gathered information about the system used in smart kids kindergarten by meeting with the administrator and the teachers there. Finally, a website was created to help parents communicate with teachers and track their child's educational level and behavior.

إهداء

إلى المعلم الأول.... رسولنا الكريم سيد البشرية محمد بن عبد الله

إلى من هم أحق منا بالحياة إلى.....الشهداء.

إلى الأسود الرابضة خلف القضبانإلى من كسروا قيد السجان الأسرى.

إلى.... أنشودة الصغر وقدوة الكبر إلى.....أبي العزيز.

إلى.... نبع العطاء وسيل الحنان إلى.....أمي العزيزة.

إلى عنوان سعادتني إلى.....إخوتي الأعزاء.

إلى.... هبة السماءأصدقائي الأوفياء.

إلى الشموع التي احترقت لتتير الدرب إلى.....أساتذتي.

إلى.... من عرفتهم في هذا الصرح العلميزملائي وزميلاتي.

إلى.... منهل العلم إلى.....جامعتي.

Table of contents

Chapter 1: Introduction	7
1.1. Overview	7
1.2. Problem Statement	7
1.3. Proposed Solution	8
1.4. The project's importance	9
1.5. Project goals	9
1.6. Alternatives	10
1.7. Project Scope	10
1.8. Timeline/ Project Scheduling	11
Chapter 2: Requirements Analysis	14
2.1. Overview	14
2.2. Context Diagram	14
2.3. Functional requirements	15
2.3.1. Functional requirements as user requirements	15
2.3.2. Functional requirements as a system requirement	15
2.4. Non-functional Requirements	16
2.5. Use Case	17
2.6. Use Case Description	18
Chapter 3: System Design	23
3.1. Overview	23
3.2. System Architecture	23
3.3. Conceptual Schema of the Database	25
3.4. Logical Data Model	26
3.5. Normalized Relational Database	28
3.6. Description of Database Table	28
3.7. Screens	36
Chapter 4: Software Demonstration	44
4.1. Overview	44
4.2. Technology And Tools Used	44
4.3. Implementation Details	50
4.4. Implementation Issues	51
Chapter 5: Testing	52
5.1. Overview	52
5.2. Functional Requirements Testing	52
5.3. Unit Testing	53
Chapter 6: Conclusion and Future Work	55
6.1. Conclusion	55
6.2. Future Work	55
References	56

List of Tables

Chapter 1: Introduction	8
Table 1.1: Project Tasks	12
Chapter 2: Requirements Analysis	15
Table 2.1 Register Student.	19
Table 2.2 Gives Feedback about Kindergarten.	20
Table 2.3 Gives Feedback about Student.	20
Table 2.4 Finds Location.	21
Table 2.5 Play a Game	21
Table 2.6 Update Student Info	22
Table 2.7 Delete a Student	22
Chapter 3: System Design	24
Tabel:3.1 Database tables	29
Table 3.13 Staff table	30
Table 3.6 Student table	31
Table 3.15 Student strength table	31
Table 3.16 Student weakness table	32
Table 3.2 After time student table	32
Table 3.9 Previous kindergarten table	33
Table 3.8 Student-disease table	34

List of Figures

Chapter 1: Introduction	8
Figure 1.1: Number of weeks required to achieve each task using Gantt Chart.	13
Chapter 2: Requirements Analysis	15
Figure 2.1 Context model	15
Figure 2.2: Use case model	18
Chapter 3: System Design	24
Figure 3.1 MVC Design Pattern	25
Figure 3.1 EER model	26
	57

Chapter 1

Introduction

- 1.1. Overview
- 1.2. Problem Statement
- 1.3. Solution
- 1.4. The Project's Importance
- 1.5. Project Goals
- 1.6. Project Scope
- 1.7. Timeline/ Project Scheduling

Chapter 1: Introduction

1.1. Overview

Technology has impacted almost every aspect of life today. The positive impact of technology on society has brought change and helped us to reach new heights that have never before been conceived of. However, we have observed that this change did not affect the educational sector accurately and completely.

The project's primary goal is to create an application that enables parents and teachers to communicate so that parents may track their kids' progress and evaluate their intellectual growth. It also helps parents participate in the educational process by integrating games that help the child review what was explained during class lessons using gamification technology, which is an educational approach that seeks to motivate students by incorporating educational games that increase student understanding. In addition, the application streamlines the registration process for parents by allowing them to complete their child's registration paperwork online, without needing to visit the kindergarten in person. This makes it easier for parents to enroll their child in the educational program, and reduces the administrative burden for the kindergarten staff.

1.2. Problem Statement

Modern technology is now a basic requirement, and there is no doubt that the traditional methods of registration require a lot of time and effort. During the registration period, many parents come at once to submit their children's required documents, and some of the paperwork may be lost or overlapped. Even after registering, some parents are unable to visit the kindergarten to see the environment in which their child is kept. Additionally, some parents may not notice the skills that their children need to improve in order to excel academically.

1.3. Proposed Solution

We will develop a web application that will present an overview of the kindergarten, which would make it easier for the parents to determine the place where their kids will be so that they can feel safe. Also the application also allows parents to upload important papers to complete the student registration process, and an individual account that would be accessed by the parent will be created for each child using student id. Finally, the application integrates games that help the child review what was explained during class lessons using gamification technology

1.4. The project's importance

The project benefits different users, including:

- **Administration and teaching staff:**
 1. Creating a database of all students.
 2. Encouraging Healthy Competition Among Students
 3. Facilitating and reducing work pressure.
 4. Increasing the loyalty of parents by taking their opinions into account.
 5. Saving time and effort.
- **Parents:**
 1. Allowing parents to know their child's educational development by logging into his/her account.
 2. Encouraging the child to be involved in different activities and follow the rules of the kindergarten.
- **Team:**
 1. Allowing practical application of what has been learned during the study stage at the university.
 2. Gaining experience to engage in the labor market after graduation.

1.5. Project goals

The project team aims to design and build a web application for the registration and follow-up of the child, where information is exchanged and accessed between the target groups based on specific criteria to ensure the security and confidentiality of the exchanged information. This is done by developing a database that helps parents track their child's development from the moment he/she joins kindergarten, taking into account the ease of use and interactive design.

1.6. Alternatives

The alternative project is an online platform that connects parents with tutors to help their children excel academically. The platform offers a variety of tutors specializing in different subjects, so parents can select a tutor who is best suited for their child's academic needs. Tutors provide online lessons and personalized learning plans based on the child's academic level and learning style. The platform also offers progress tracking tools, enabling parents to track their child's progress over time. Additionally, the platform provides a feedback system that enables parents to rate and review tutors, ensuring that the best tutors are recommended to other parents.

Compared to the alternative project mentioned earlier, the primary difference with our project is that it focuses on collaboration between parents and teachers in addition to offering games that help children review and understand class lessons. Our project aims to bridge the communication gap between parents and teachers by providing regular progress updates, ensuring that parents have a better understanding of their child's educational progress and areas of improvement

1.7. Project Scope

Creating a web application for the (روضة الذكاء العقلي) in the first place, with the possibility that this application can be used in other kindergartens on the basis and criteria agreed upon between the kindergartens and us.

As the project targets parents, students, and educational staff. It simplifies the registration process and helps the parents to be informed of the documents necessary for registration and the general situation of the children.

1.8. Timeline/ Project Scheduling

We will follow the Software Development Life Cycle to develop the web application. As presented in Table 1.1 these are the tasks that we should do. Also, Figure 1.1 shows the number of weeks required to achieve each task as a Gantt Chart Figure.

Table 1.1: Project Tasks

Task number	Task name	The time required (week)
1	System definition and planning	5
2	Determine the project requirements.	4
3	Description of the project Requirements.	4
4	System design.	4
5	System development and implementation.	9
6	system testing.	5
7	Documenting the application.	During the working period

	First Semester				Second semester			
weeks	2-6	7-10	11-14		2-6	7-11	12-14	
System definition and planning								
Determine the project requirements.								
Description of the project Requirements.								
System design.								
System development and programming.								
Integration and system testing.								
Documenting the application.								

Figure 1.1: Number of weeks required to achieve each task using Gantt Chart.

Notation Used:

- Estimate time to finish the task.
- Real time to finish the task.
- Holiday between semester.

Chapter 2

Requirements Specification

- 2.1. Overview
- 2.2. Context Diagram
- 2.3. Functional requirements
- 2.4. Non-functional Requirements
- 2.5. Use Case
- 2.6. Use Case Description

Chapter 2: Requirements Analysis

2.1. Overview

This stage is one of the most important stages that the project is going through. In this chapter all functional and non-functional requirements will be presented.

2.2. Context Diagram

The context diagram shows the interactions between a system and the actors or between the system and other software systems as shown in figure 2.1

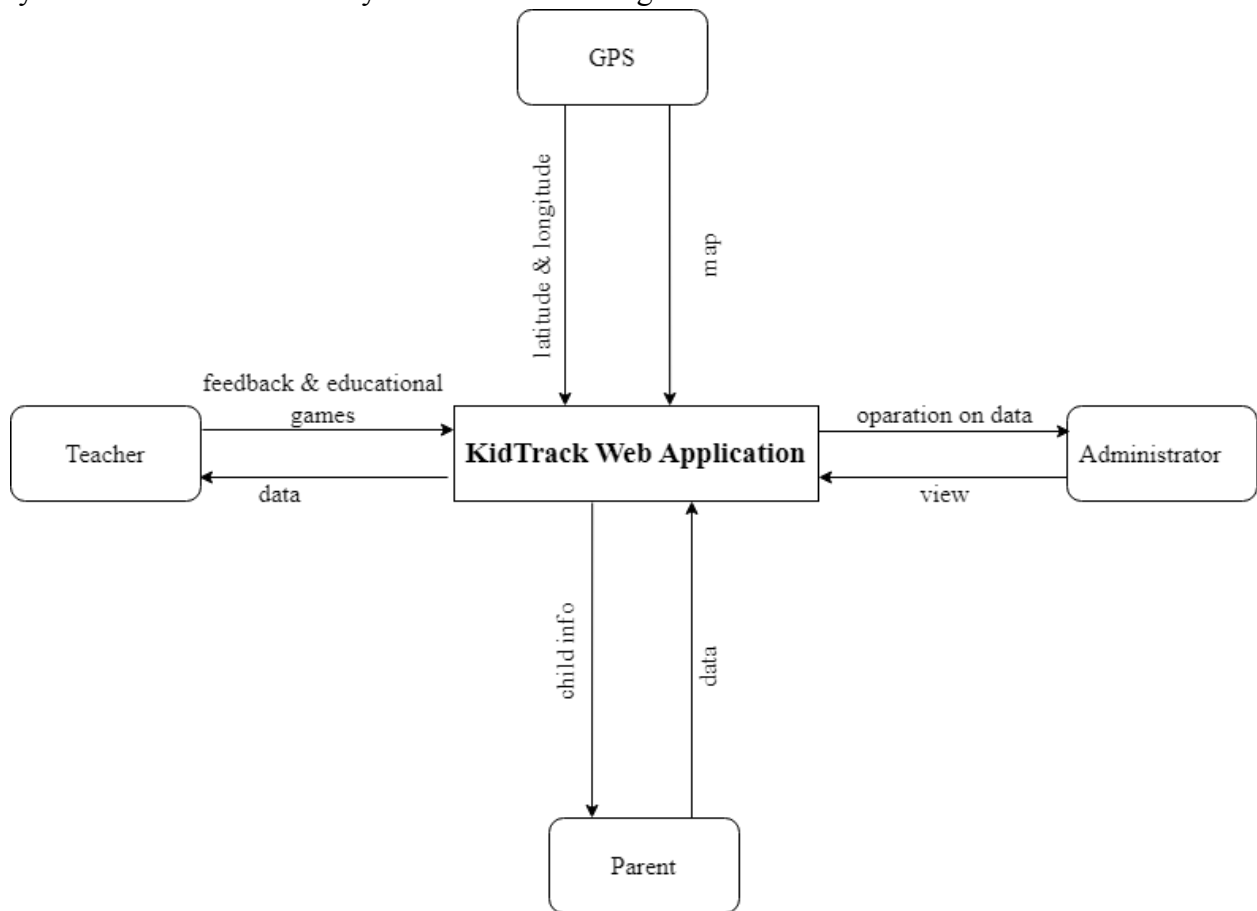


Figure 2.1 Context model

2.3. Functional requirements

The objective of these requirements is to define the functional aspects of the system, how it interacts with inputs and outputs, and how it behaves in certain cases. It includes the requirements of the system and user.

2.3.1. Functional requirements as user requirements

The User Requirements Specification describes the business needs for what users require from the system.

- The parents shall view the weekly feedback for their child
- The teacher shall give weekly feedback to each student
- The teacher shall add new material to improve student understanding.
- The parent shall view the added material by the teacher.
- The parents shall register their child using an online form.
- The administrator shall register the child using an online form.
- The parent shall find the location of the kindergarten using GPS technology.
- The parent shall give feedback to the kindergarten.
- The administrator shall view the feedback that parents give about the kindergarten.
- The student shall recap daily lessons by playing educational games.

2.3.2. Functional requirements as a system requirement

A functional system requirement is a statement of how a system must behave. It defines what the system should do in order to meet the user's needs or expectations.

- The application shall display an avatar or real image for each student.
- The application shall give different educational games.

2.4. Non-functional Requirements

Regarding non-functional requirements, our work focused on the following:

- **Ease of dealing with the application:**

1. The interface design should be appropriate depending on the user by contrasting a color scheme.
2. Present information in an understandable way so it's easy to scan the content of the website.
3. The application should be responsive when it is opened from different platforms.

- **Security:**

The application has been able to maintain trust between users by following these methods during the establishment of the website:

1. Validates user input.
2. Every user can provide his personal information to prove his identity to other users.
3. Do not permit an unauthorized person to access the system or modify the database.

- **Accuracy:**

The system must be enabled to continue operating effectively even when unexpected errors occur.

1. The system works permanently and for the longest possible period
2. The server has the capacity to handle responses for 500 users within a span of 1 minute.

2.5. Use Case

The use case describes the user's view of the application and what the users can do.

We have four actors, and each one of them has a number of actions, as shown in Figure 2.1.

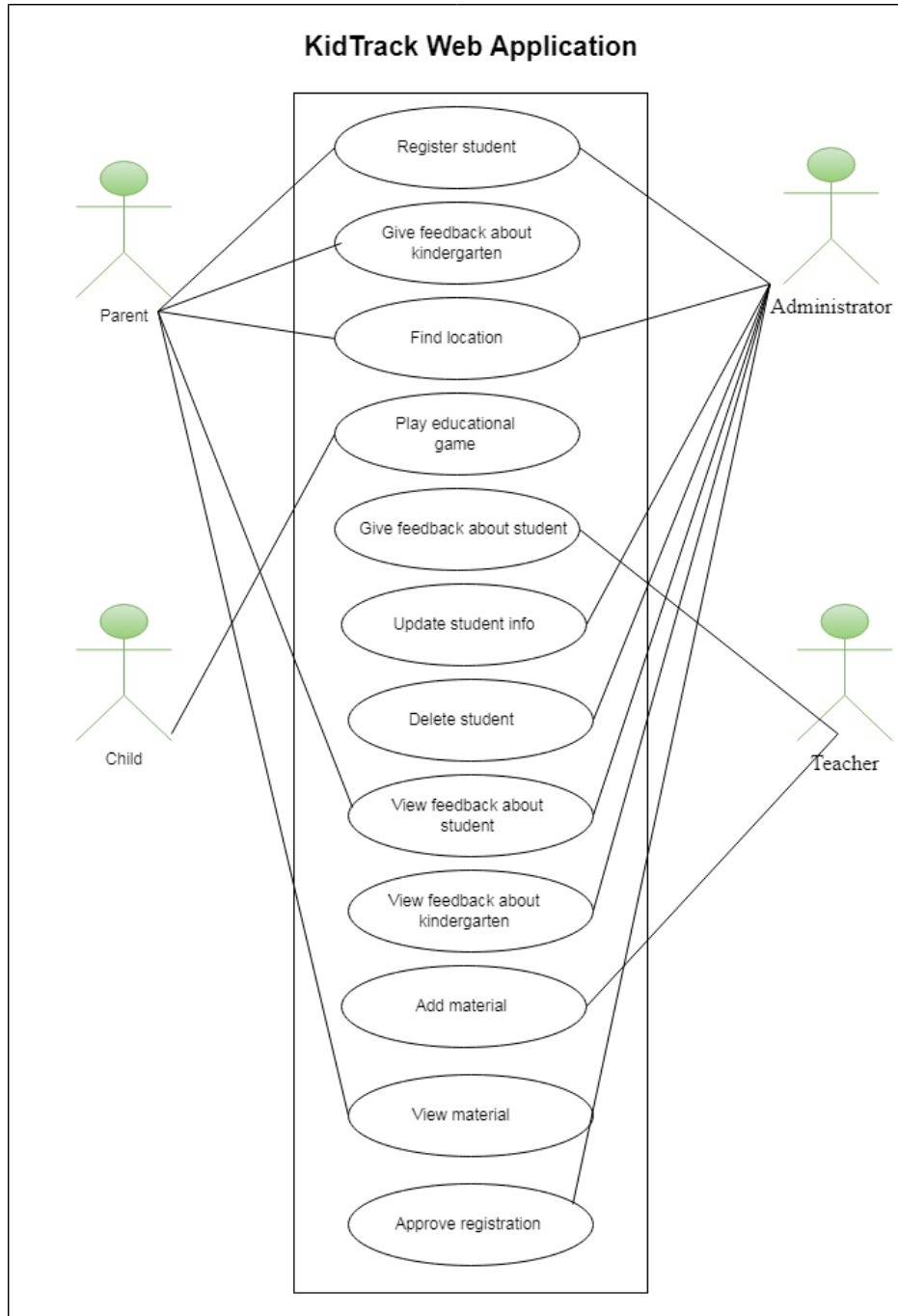


Figure 2.2: Use case model

2.6. Use Case Description

A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.

Table 2.1 Register Student.

Use case	Register student.
Aactor	Parent, administrator
Goal	Complete the student registration process.
Preconditions	The user should be on the sign-up page.
Scenario	<ol style="list-style-type: none">1. Users enter the URL for the sign-up page.2. The system displays a sign-up page that asks the user to enter a unique username(id number) and password.3. The user submits their data to the sign-up system by clicking on the register button.4. The sign-up system checks the uniqueness of the username.5. System stores the user's data in a database.6. The system redirects to the home page for each user.
Exceptions	<ol style="list-style-type: none">1. The username is not unique(the user has already registered)2. The user does not fill all the fields.3. No internet connection is available.

Table 2.2 Gives Feedback about Kindergarten.

Use case	Give feedback about kindergarten
Actor	Parent
Goal	<ol style="list-style-type: none"> 1. Kindergarten Improvement 2. Give feedback about kindergarten for the parents who intend to register their child
Preconditions	Having an account.
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the student page, choose the option to send feedback. 3. The feedback dialog opens. 4. Write the note on its field. 5. Click the send button.
Exceptions	<ol style="list-style-type: none"> 1. Closing the form dialog before pressing the submit button. 2. No internet connection is available.

Table 2.3 Gives Feedback about Student.

Use case	Give feedback about student
Actor	Teacher
Goal	Help parents to check their child's academic progress.
Preconditions	Having an account as a teacher.
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the progress page, a table of all students in the teacher class rendering. 3. The teacher chooses the feedback option for the target student. 4. A feedback form opens. 5. The teacher writes the current week's feedback. 6. Click the send button.
Exceptions	<ol style="list-style-type: none"> 1. Closing the form dialog before pressing the submit button. 2. No internet connection is available.

Table 2.4 Finds Location.

Use case	Find location
Actor	Parent, Teacher and Administrator
Goal	Find the location of kindergarten
Preconditions	There doesn't
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the landing page, choose the map icon in the navbar. 3. Moving to a new screen where the map is displayed.
Exceptions	<ol style="list-style-type: none"> 1. No internet connection is available. 2. GPS is not enabled.

Table 2.5 Play a Game

Use case	Playing education game
Actor	Child, teacher, administrator
Goal	increase student understanding
Preconditions	Having an account.
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the home page, choose the material option. 3. Play the game or watch the video that has been added by the teacher. 4. Then the system will increase the number of points(tokens) for the student. 5. The system redirects to the material page.
Exceptions	<ol style="list-style-type: none"> 1. No internet connection is available.

Table 2.6 Update Student Info

Use case	Update student info
Actor	Parent, administrator
Goal	Edit student info
Preconditions	Having an account.
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the home page, the user chooses the setting icon. 3. A drop-down list appears then the user clicks edit profile. 4. After editing, the user clicks on the save button 5. The system updates the student's info.
Exceptions	<ol style="list-style-type: none"> 1. No internet connection is available. 2. The user does not fill in all fields.

Table 2.7 Delete a Student

Use case	Delete student
Actor	administrator
Goal	Delete student permanently from the database
Preconditions	Having an account.
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the student page, the user chooses the delete button for the student he/she wants to delete 3. A confirmation message appears. 4. The administrator confirms the deletion. 5. The system deletes the student.
Exceptions	<ol style="list-style-type: none"> 1. No internet connection is available.

Table 2.8 Approve Registration

Use case	Approve registration
Actor	administrator
Goal	Approve student registration from the database
Preconditions	Having an account.
Scenario	<ol style="list-style-type: none"> 1. Log in to the system. 2. Through the confirm page, the administrator chooses the confirm button for the student he/she wants to regist 3. A confirmation message appears. 4. The administrator confirms the registration . 5. The system adds the student.
Exceptions	<ol style="list-style-type: none"> 1. No internet connection is available.

Chapter 3

System Design

- 3.1. Overview
- 3.2. System Architecture
- 3.3. EER Diagram
- 3.4. Logical Data Model
- 3.5. Database tables
- 3.6. Screens (views)

Chapter 3: System Design

3.1. Overview

This chapter includes an explanation of the project's design, tools, and structure, as well as a detailed description of the system's components to provide a comprehensive understanding of the entire system.

3.2. System Architecture

A system's architecture reflects how the system is used and how it interacts with other systems and the outside world. It describes the interconnection of all system components as well as the data link that connects them.

Model-View and Control

MVC (Model-View-Controller) is an architectural design pattern that promotes better application organization by separating concerns into three parts:

- **Model:**

The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data.

- **View:**

View means the presentation of the model in a particular format. For creating views you may use template engines, frontend framework which is our choice, or other technologies.

- **Controller:**

Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component, and interact with the Views to render the final output.

The MVC design pattern decouples these major components, allowing for more efficient code and parallel development. The figure below shows the interaction between the user and the application components.

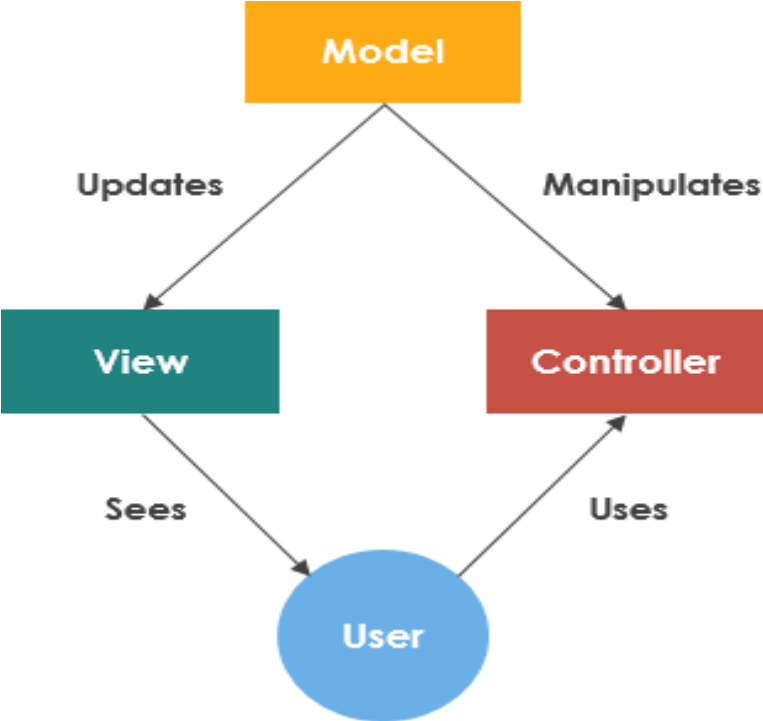


Figure 3.1 MVC Design Pattern

3.3. Conceptual Schema of the Database

A conceptual schema is a high-level description of information needs that are to be addressed by a certain database or information system. It is typically the first phase for designing a database schema from scratch.

The most common form for conceptual schemas is Enhanced Entity-relationship(EER) diagram which provides a visual representation of the relationships among the tables in the database model.

In our application we have twelve entities, and each one of them has a number of attributes, as shown in figure 3.1.

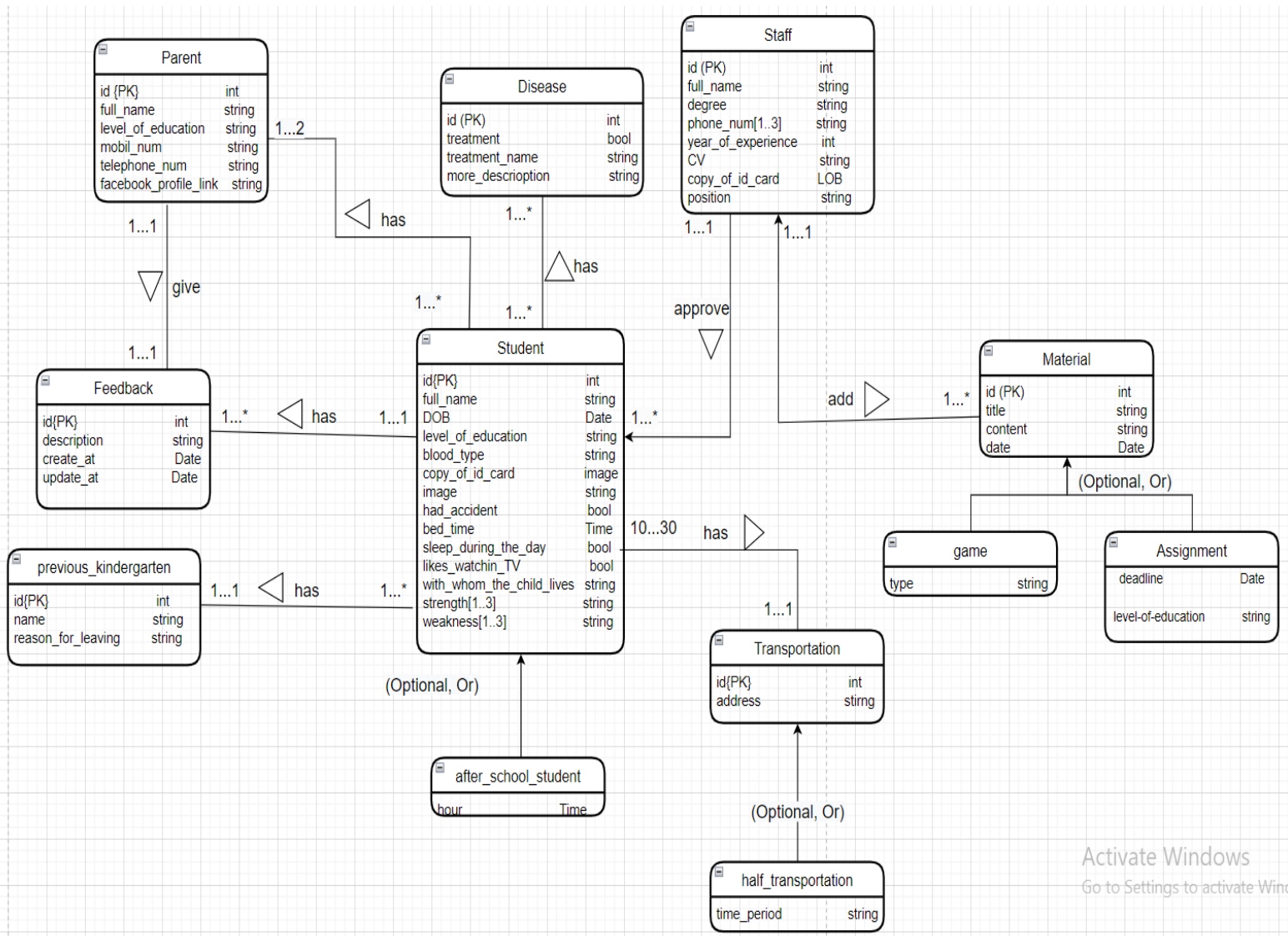


Figure 3.1 EER model

3.4. Logical Data Model

After creating the EER diagram, we have developed a logical data model that captures and represents the entities, attributes, and relationships within the database design. The logical data model serves as an intermediary between the business requirements and the physical implementation, providing a comprehensive and structured representation of the data elements.

Staff

(id, full-name, degree, year-of-experience, CV, copy-of-id, position).

Staff-phoneNum

(staff-id, phone-num)

Foreign key: staff-id reference Staff(id).

Student

(id, full_name, DOB, level-of-education, blood-type, copy-of-id-card, img, had-accident, bed-time, sleep-during-the-day, like-watching-TV, with-whom-child-live, staff-id).

Foreign key: staff-id reference Staff(id).

Student-Strength

(id, strength-desc).

Student-Weakness

(id, weakness-desc).

After-school-student

(id, full_name, DOB, level-of-education, blood-type, copy-of-id-card, img, had-accident, bed-time, sleep-during-the-day, like-watching-TV, with-whom-child-live, hour).

Previous-kindergarten

(id, name, reason-for-leaving, student-id).

Foreign key: student-id reference Student(id).

Feedback

(id, description, create-date, update-date, student-id, staff-id).

Foreign key: student-id reference Student(id).

Foreign key: staff-id reference Staff(id).

Disease (id, treatment, treatment-name, more-description).

Student-Disease

(student-id, disease-id).

Foreign key: student-id reference Student(id).

Foreign key: disease-id reference Disease(id).

Parent

(id, full-name, level-of-edu, mobile-num, telephone-num, facebook-profile-link)

Student-Parent

(student-id, parent-id).

Foreign key: student-id reference Student(id).

Foreign key: parent-id reference Parent(id).

Transportation

(id, address, student-id).

Foreign key: student-id reference Student(id).

Half-transportation

(id, address, time-period, student-id).

Foreign key: student-id reference Student(id).

Material

(id, title, content, date, staff-id)

Foreign key: staff-id reference Staff(id).

Game

(id, title, content, date, staff-id, type)

Foreign key: staff-id reference Staff(id).

Assignment

(id, title, content, date, staff-id, deadline, level-of-education)

Foreign key: staff-id reference Staff(id).

3.5. Normalized Relational Database

We first ensured that each table had a primary key and that each column contained atomic values without any repeating groups or arrays of data, which meant that our database was in first normal form (1NF). We then checked that each non-key column was fully dependent on the primary key, eliminating partial dependencies and resulting in our database being in second normal form (2NF). Finally, we ensured that there were no transitive dependencies between non-key columns, meaning that our database was in third normal form (3NF). By following these normalization steps, we have optimized our database for data integrity, reduced redundancy, and improved efficiency, ultimately resulting in a more effective and reliable database.

3.6. Description of Database Table

The system is associated with a database consisting of a number of tables interconnected with each other through common relationships between them as the following:

- Database Tables:

Tabel:3.1 Database tables

Tabel name	Description
Student	Save the information of the student.
parent	Save the information of the parent.
Staff	Save the information of the staff.
Material	Save the information of materials that staff added.
Transportation	Save the student's home address
Feedback	Save the teacher feedback about the student.
Previous-kindergarten	Save the reason for leaving from the previous kindergarten.
Disease	Store the health state for students.
After-time-student	Set the time that child will spend in kindergarten.
Game	Save information of a game.

Assignment	Save the deadline of the assignments that staff added.
Half-transportation	Store if the transportation will be in the morning or evening in case of half transportation.
Staff-phone-num	Store the staff numbers.
Parent-mobile-num	Store the parent numbers.
Student-strength	Store the student strengths.
Student-weakness	Store the student's weaknesses.

- Staff table:

Table 3.13 Staff table

Field name	Data type	Size	Null
id	Int	3	No
full_name	String	45	No
degree	String	10	No
year_of_experience	int	4	No
CV	String	65	No
id_card	int	10	No
position	String	20	No

- Staff-phone-num table:

Table 3.14 Staff phone num table

Field name	Data type	Size	Null
staff_id	Int	3	No
phone_num	Int	10	No

- Student table :

Table 3.6 Student table

Field name	Data type	Size	Null
id	Int	6	No
full_name	String	45	No
DOB	Date		No
blood_type	String	3	No
id_card	int	10	No
image	String	60	No
had_accident	bool	2	No
bed_time	Time		No
sleep_during_the_day	bool	2	No
likes_watching_TV	bool	2	No
with_whom_child_live	String	10	No
level_of_education	String	10	No
staff_id	Int	3	No

- Student-strength table:

Table 3.15 Student strength table

Field name	Data type	Size	Null
student_id	Int	6	No
strength	String	60	No

- Student-weakness table:

Table 3.16 Student weakness table

Field name	Data type	Size	Null
student_id	Int	6	No
weakness	String	60	No

- After-school-student table:

Table 3.2 After time student table

Field name	Data type	Size	Null
id	Int	6	No
full_name	String	45	No
DOB	Date		No
blood_type	String	3	No
copy_of_id_card	String	10	No
image	String	60	No
had_accident	bool	2	No
bed_time	Time	3	No
sleep_during_the_day	bool	2	No
likes_watching_TV	bool	2	No
with_whom_child_live	String	10	No
level_of_education	String	10	No
staff_id	Int	3	No
hour	Time		No

- Previous-kindergarten:

Table 3.9 Previous kindergarten table

Field name	Data type	Size	Null
student_id	Int	6	No
name	String	10	Yes
reason_for_leaving	String	80	No

- Feedback table:

Table 3.12 Feedback table

Field name	Data type	Size	Null
id	Int	6	No
description	String	90	No
create_at	Date	3	No
update_at	Date	3	No
student_id	Int	6	No
staff_id	Int	3	No

- Disease table:

Table 3.8 Disease table

Field name	Data type	Size	Null
ID	Int	6	No
treatment_name	String	20	No
treatment	bool	2	No
more_describtion	String	80	No

- Student-Disease table:

Table 3.8 Student-disease table

Field name	Data type	Size	Null
student_id	Int	6	No
disease_id	Int	6	No

- Parent table

Table 3.7 Parent table

Field name	Data type	Size	Null
ID	Int	6	No
full_name	String	50	No
level_of_education	String	10	No
mobile_num	String	10	No
telephone_num	String	10	No
facebook_profile_link	String	60	No

- Student-Parent table:

Table 3.8 Student-parent table

Field name	Data type	Size	Null
student_id	Int	6	No
parent_id	Int	5	No

- Transportation table:

Table 3.10 Transportation table

Field name	Data type	Size	Null
id	Int	6	No
address	String	50	No
student_id	Int	6	No

- Half-transportation table:

Table 3.11 Half transportation table

Field name	Data type	Size	Null
id	Int	6	No
address	String	50	No
time_period	String	10	Yes
student_id	Int	6	No

- Material table:

Table 3.4 Material table

Field name	Data type	Size	Null
id	Int	6	No
title	String	20	No
content	String	80	No
date	Date		No
staff-id	Int	6	No

- Assignment table:

Table 3.3 Assignment table

Field name	Data type	Size	Null
id	Int	6	No
title	String	20	No
content	String	80	No
date	Date	3	No
staff-id	Int	6	No
deadline	Date	3	No

- Game table:

Table 3.5 Game table

Field name	Data type	Size	Null
id	Int	6	No
title	String	20	No
content	String	80	No
date	Date	3	No
type	String	10	No
staff-id	Int	6	No

3.7. Screens

We created a basic interface for the game using Figma based on the functional requirement that we decided.

- **Landing Page**

A landing page aims to capture information about the kindergarten also it helps users to navigate through the web application pages



تهتم روضة الذكاء العقلي حول العديد من الجوانب لنمو طفلك
مثل :



راحة لطفلك

تنويع الطرق التعليمية واستخدام التكنولوجيا لتحقيق تجارب تعلم متفاعلة وشاملة تعزز التفكير النقدي والابتكار وتمكن الطلاب من اكتساب المهارات اللازمة لمواجهة التحديات المستقبل



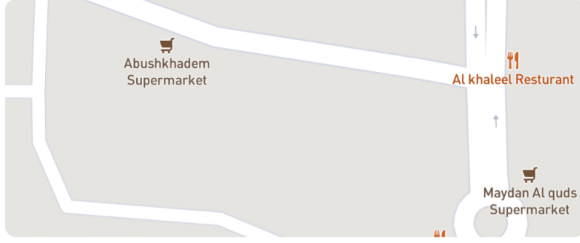
تعليم مرح

عزز الحماس والاندماج في عملية التعلم، حيث يتم توظيف الأنشطة اللببية والتفاعلية لجعل التعلم ممتعا ومثيرا للاهتمام، مما يساهم في تعزيز التفاعل الإيجابي وتحقيق نتائج أفضل للطلاب



تعليم حديث

توفير بيئة مريحة ومحبة، وتقديم الرعاية الشخصية والاهتمام الفردي لكل طفل، وتوفير فرص للتفاعل الاجتماعي واللعب الهادف، مما يساهم في إشباع احتياجاته الجسدية والعاطفية وتعزيز شعوره بالأمان والانتماء في بيئة روضة سعيدة ومحفزة



معلومات التواصل

الاييميل : smart-kids@gmail.com

رقم الهاتف : 5421-322-022

الموقع : الخليل - رأس الجورة



- Sign up page

On the Sign up page, Student enters his/her full name, id number and other required student information.

رقم الهوية	الاسم الرباعي
<input type="text"/>	<input type="text"/>
صورة للطفل	تاريخ الميلاد
No file chosen <input type="button" value="Choose file"/>	dd / mm / yyyy <input type="text"/>
* الجنس	* مع من يعيش الطفل
ذكر <input type="text"/>	الأم والأب <input type="text"/>
* يشاهد التلفاز	* زمرة الدم
نعم <input type="text"/>	+A <input type="text"/>
نقاط القوة	
<input type="text"/>	
نقاط الضعف	
<input type="text"/>	
<input type="button" value="السابق"/>	<input type="button" value="التالي"/>

عنوان البيت	* المواصلات
<input type="text"/>	مواصلات كاملة (ذهابوايابا) <input type="text"/>
* بنام في النهار	موعد المغادرة
نعم <input type="text"/>	-- : -- <input type="text"/>
وقت النوم	* تعرض لحادث
-- : -- <input type="text"/>	نعم <input type="text"/>
اسم الأم	اسم الأب
<input type="text"/>	<input type="text"/>
رقم هاتف الأب	رقم هاتف الأم
<input type="text"/>	<input type="text"/>
رابط صفحة الفيس بوك	رقم هاتف الاخر
<input type="text"/>	<input type="text"/>
مستوى التعليم للأب	مستوى التعليم للأم
<input type="text"/>	<input type="text"/>
<input type="button" value="السابق"/>	<input type="button" value="ارسل"/>

- **Sign in page**

On the Sign in page, the user can access their account by entering the previously created login name and password.



تسجيل الدخول

الايمل

mera@chill.com

كلمة المرور

.....

تسجيل الدخول

- **Change password page**

The Password Reset Page provides your applications' users with a way to change their passwords if they cannot log in.



استرجاع الحساب

الايمل

أدخل الإيميل

تأكيد

هل تنكرت كلمة المرور؟ تسجيل الدخول

- **user verification page**

User verification page is to support identifying a user, authorize and then determine whether a user can take specific actions



تأكيد الرمز

الرمز

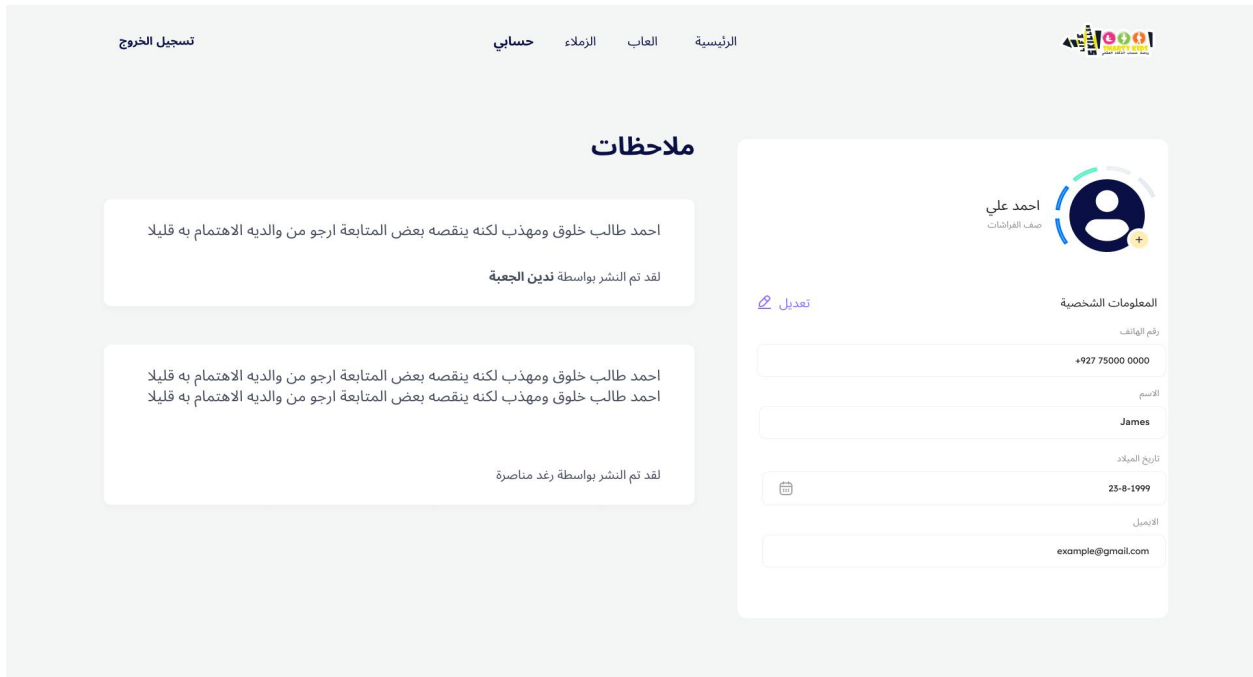
ادخل الرمز

تأكيد

هل أدخلت بريدا خاطئا؟ [عودة](#)
هل تود إلغاء العملية؟ [إلغاء](#)

- **Profile page**

The user profile page displays the data related to a particular user



- **All students Page**

All student page display all student in a the kindergarten.



- **Add material page**

Add material page helps the teacher to add new material by giving the title, type and content of it.



- **Material Page**

Material page display all material added by the teacher.

الرئيسية العايب الزملاء حسابي تسجيل الخروج

المهام

مشاهدة فيديو الانترنت 11-2-2022
رؤية المنشور

تحضير الحروف 11-2-2022
رؤية المنشور

مشاهدة فيديو الانترنت 11-2-2022
رؤية المنشور

مشاهدة فيديو الانترنت 11-2-2022
رؤية المنشور

مشاهدة فيديو الانترنت 11-2-2022
رؤية المنشور

إضافة منشور

المنشورات

مشاهدة فيديو الانترنت والسلحفاة

اعزائي الطلاب يجب عليكم مشاهدة هذا الفيديو قبل يوم الاحد القادم

لقد تم النشر بواسطة ميرا قنيبي لقد تمت مشاهدته

اعلان لفعالية يوم الام

تعلمن الروضة عن عمل فعالية يوم الام يوم الخميس

لقد تم النشر بواسطة ميرا قنيبي وضع علامة مشاهدة

تحضير الحروف

اعزائي الطلاب يجب عليكم التدرب على الحروف وشكرا (:

Chapter 4: Software Demonstration

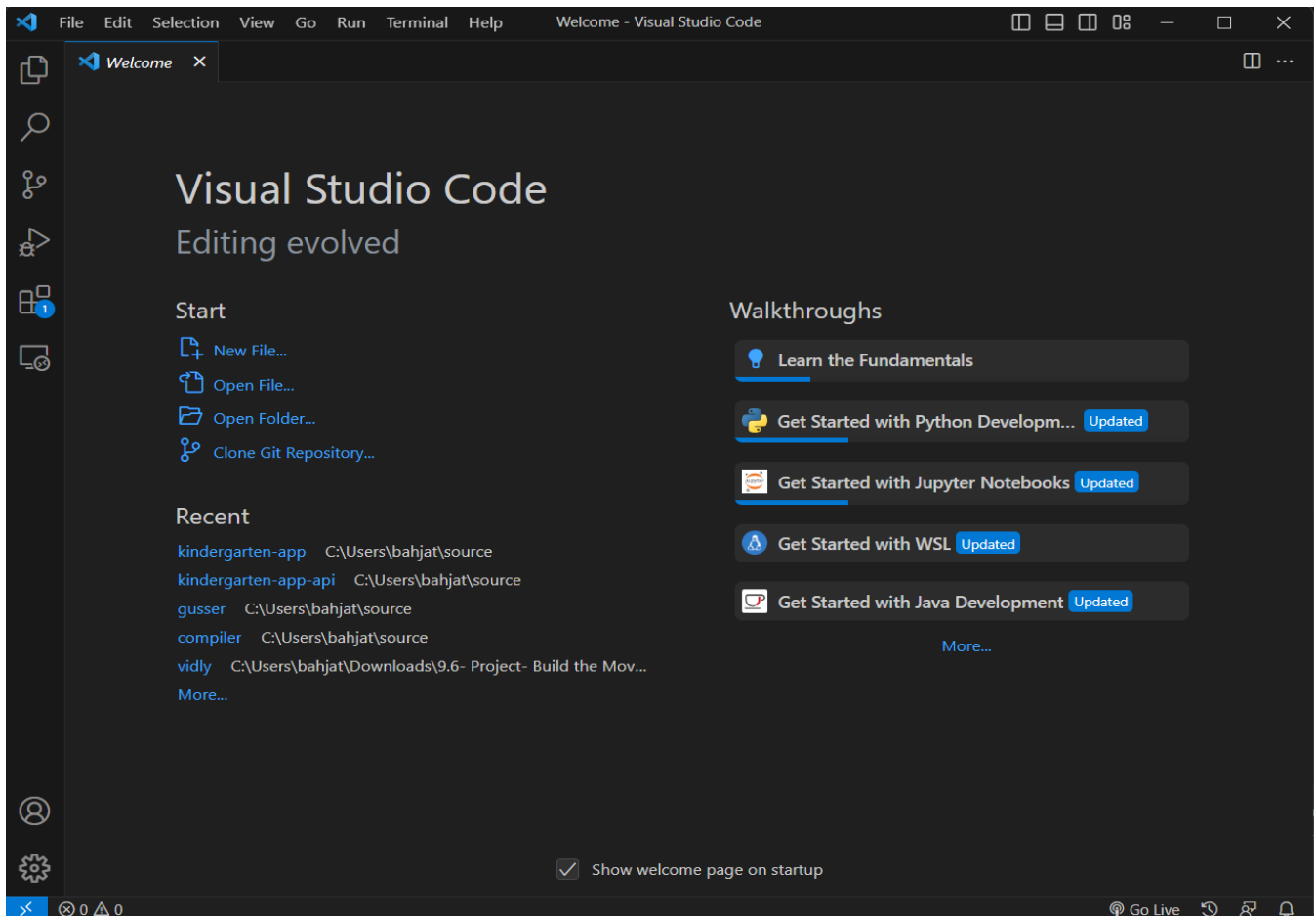
4.1. Overview

This chapter will discuss the technologies that were used in building the web application. Also some of the implementation details and issues that faced the project during the implementation.

4.2. Technology And Tools Used

■ Visual Studio Code

It is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.[1]



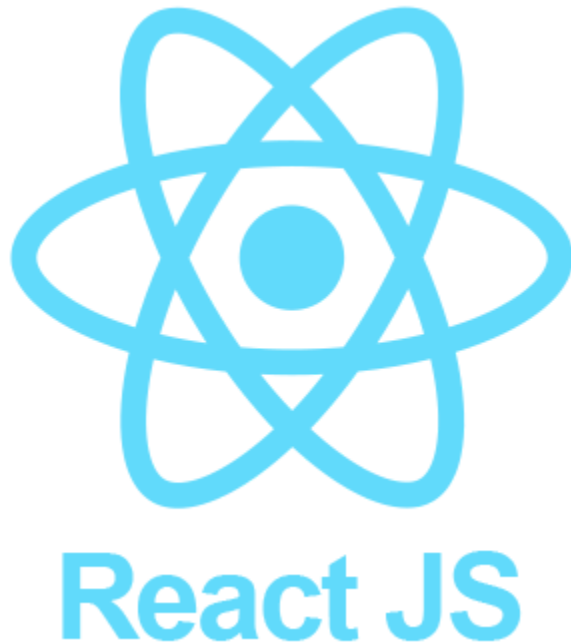
- **GitHub**

It is a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration.[18] Also It offers the distributed version control and source code management functionality of Git. GitHub and Git were as important as other technologies that we used. Each update in the project was pushed on github with a message that represents the changes made. So we can go back for any version of the project we want.[2]



- **React js**

It is a free and open-source JavaScript framework and library developed by Meta . It's used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript. React can be used to develop single-page, mobile, or server-rendered applications with frameworks.[3]



- **Nodejs**

It is an open source, cross-platform runtime environment for developing server-side and networking applications and is. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.[4]



■ MongoDB

It is an open-source, cross-platform, and distributed document-based database designed for ease of application development and scaling. It is a NoSQL database developed by MongoDB Inc[5]



- **Postman**

Postman is a popular collaboration platform and API development tool used by developers and teams to simplify and streamline the process of designing, building, testing, and documenting APIs (Application Programming Interfaces). It provides a user-friendly interface that allows developers to create, send, and receive HTTP requests and analyze the responses.[6]



POSTMAN

4.3. Implementation Details

Our project focuses on developing an application that facilitates communication between parents and teachers while enhancing the educational experience for children. We have chosen to implement the application using ReactJS for the frontend and Node.js for the backend.

Implementation can be divided into three main parts as follow:

1. Project Structure and Organization

- Create separate directories for the frontend (ReactJS) and backend (Node.js) code.
- Set up a version control system like Git to track changes and collaborate with the team.

2. Frontend Development (ReactJS)

In the frontend development phase, we structure the project using a component-based architecture following best practices. We create reusable components for user registration, login, and the home page, ensuring a modular codebase. Adhering to clean code principles, we use meaningful and descriptive names for components, variables, and functions. We maintain a consistent naming convention throughout the frontend codebase to improve code readability and maintainability. The user interface is designed using HTML, CSS, and ReactJS components.

3. Backend Development (Node.js)

Moving to backend development, we use Node.js to build a robust and scalable API. We define the necessary endpoints for user registration, login, and data retrieval, ensuring secure and efficient communication between the frontend and backend. Adhering to clean code practices and naming conventions, we create clear and concise route handlers, models, and controllers. We implement user authentication and authorization using techniques such as JSON Web Tokens (JWT) to ensure secure access to the application's features and protect sensitive data. Integration with MongoDB database system, allows us to store and retrieve user information, student progress, and

kindergarten announcements. We establish consistent naming conventions for database documents to ensure proper organization and maintainability.

4.4. Implementation Issues

During the development of your project, you may encounter several implementation challenges. One of the primary areas that can pose difficulties is authentication. Building a secure authentication system, including password hashing and session management with JSON Web Tokens (JWT) technologies, requires careful attention to security best practices. Additionally, ensuring proper data validation and error handling can be complex. Implementing robust validation mechanisms and providing meaningful error messages on both the frontend and backend are crucial for data integrity and a smooth user experience.

Chapter 5: Testing

5.1. Overview

This chapter will discuss the unit testing for backend APIs and functional requirements testing. To ensure that our app is working as expected and helps manage new changes in specification or implementation.

There are different types of testing for different test approaches:

- Functional testing.
- Unit testing.

5.2. Functional Requirements Testing

Functional testing is necessary to know whether the system is working as it should be or not.

Test Case	Scenario	Input Data	Expected Output	Actual Output	Test Result
Register Student	Student enter his/her full name, id number and other required student information	Full name, Id num, Copy of id card, image, Level of education, Other student	Student added successful	Student added successful	pass

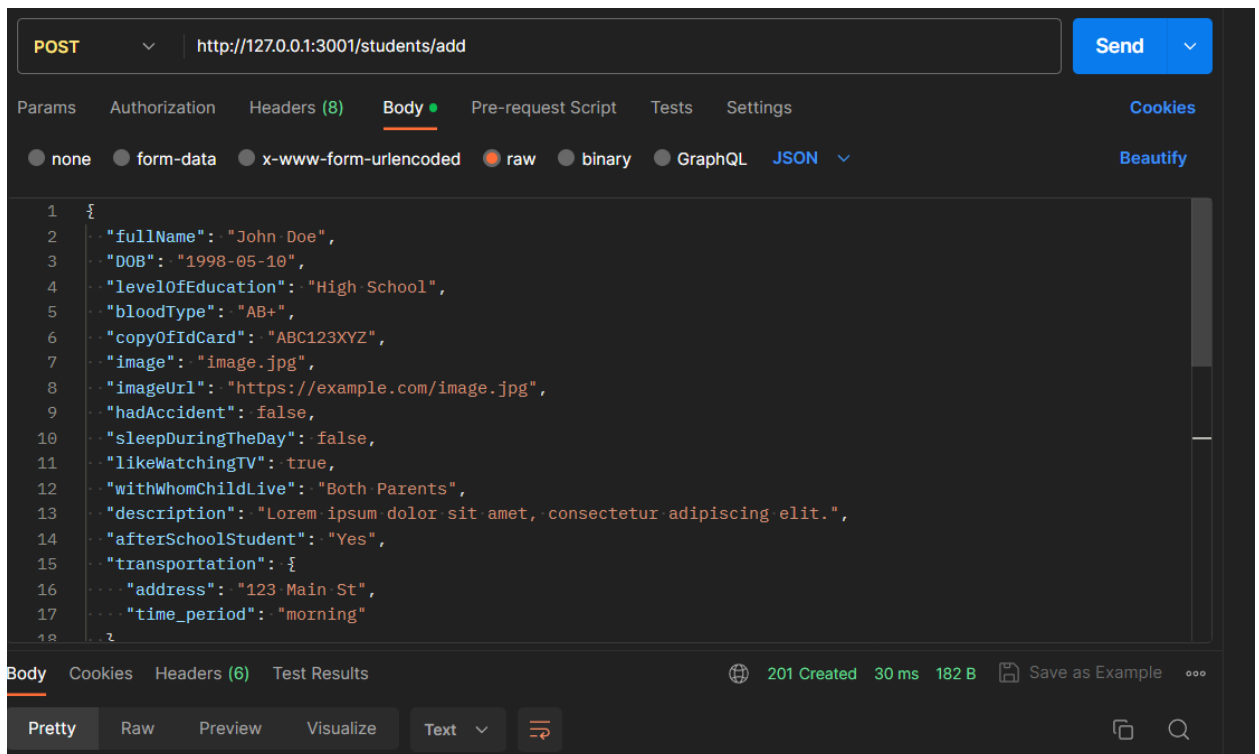
	Student enter his/her full name, id number and other required student information but his/her name is already registered in the database	Full name, Id num, Copy of id card, image, Level of education, Other student	You have been already register	You have been already register	pass
Give feedback about kindergarten	Through the student page, the parent choice give feedback option and write note on the pop up dialog	Feedback about kindergarten	Thanks! We Appreciate your feedback	Thanks! We appreciate your feedback	pass
Give feedback about student	From the student list, the teacher chooses a comment option and starts typing.	Feedback about the student	Your feedback is added successfully	Your feedback is added successfully	pass
Add new post in the main page	From the main page choice add icon	New announcement or assignment	new post created	new post created	pass
Confirm registration	Admin confirm registered student	Confirm action	Student would be able to register using his/her username and generated password	Student sign in using his/her username and generated password	pass
Reject student	Admin reject registered student	Reject action	Student would be deleted from database	Student is deleted by id from database	pass

5.3. Unit Testing

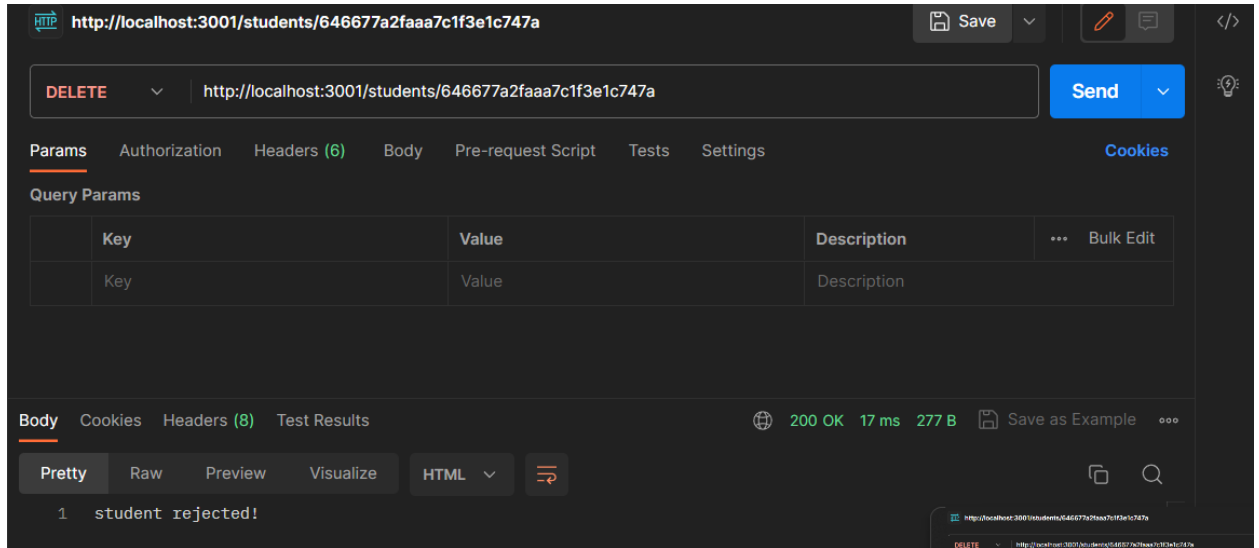
One of the most popular and important testing types is unit testing. Unit testing is basically testing if a unit or component of the system is working as expected. If no input is

required, you just call the component. If the input is required, you give it an input to determine the output. In the context of REST API, a unit is a single endpoint request, a unit test depends on what you want to test, and the response based on the request sent. We test all of our APIs using the postman testing tool.

- Test **Post** new student:



- Test **Delete** student



Chapter 6: Conclusion and Future Work

6.1. Conclusion

In the end, we have developed a web application with the interactive programming language with the intention of streamlining the registration process for students in the kindergarten and monitoring their growth and progress in learning as well as their behavior there. We also included some games that were pertinent to the kindergarten curriculum.

6.2. Future Work

In the near future, we will improve the web application by including some educational features by adding more games, and an AI chatbot to offer users immediate support and guidance.

References

[1]“Visual Studio Code.” Wikipedia, https://en.wikipedia.org/wiki/Visual_Studio_Code. Accessed 14 April 2023.

[2]“What Is GitHub? A Beginner's Introduction to GitHub.” Kinsta, 30 April 2023, <https://kinsta.com/knowledgebase/what-is-github/>. Accessed 1 May 2023.

[3]”ReactJs” Blog.hubspot, <https://blog.hubspot.com/website/react-js> Accessed 10 May 2023.

[4]”NodeJs” Tutorialspoint, https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm Accessed 15 May 2023.

[5]”MongoDB”, Tutorialsteacher, <https://www.tutorialsteacher.com/mongodb/what-is-mongodb> Accessed 25 April 2023.

[6]”Postman” Postman, <https://www.postman.com/> Accessed 20 May 2023.