Lebanese University Faculty of Economics Sciences and Business Administration

Branch III

# E-Elementary: Online Elementary School Management System

Senior project for Master degree
Specialization: **Business Computer Sciences**

Prepared by

**Najat Walid Ismail**

Jury

Supervisor: **Dr. Mohammad Hani AL JAMAL**

Examiner: **Dr. Adolf ABDALLAH**

**2019-2020**

# Abstract

On the 17th of March 2021 **171,341,667** learners are effected by the closure of schools in response to the Covid-19 pandemic, according to UNESCO. The closer of schools around the world raises the need for online schooling systems that are able to replicate and enhance the school experience of the effected learners, their teachers and the schools' administrations.

This report presents an online school management system for elementary schools that aims to help them offer students, teachers and admins an efficient and comprehensive online school experience. The system allows admins to manage teachers' and students' information, in addition to school accounts and classes. Using this system enables teachers and students to view their classes, teachers can assign homework and students can view their assigned homework in every subject.

Throughout the process of building this early stage solution, **frontend web app development technologies** were used through working with an **application framework** using Angular and **UI libraries** such as Angular Material and Nebular. **backend web app development technologies** were also used through implementing **RESTful web services** using HTTP request through Express and a **runtime environment** using Node.js and its npm library to connect them with the data stored in a **No- Sql Database** using MongoDB**.**

# Resumé

Le 17 mars 2021, 171.341.667 apprenants sont affectés par la fermeture d'écoles en réponse à la pandémie de Covid-19, selon l'UNESCO. La fermeture des écoles dans le monde augmente la nécessité des systèmes d'enseignement en ligne capables de reproduire et d'améliorer l'expérience scolaire de la personne concernée apprenants, leurs enseignants et les administrations des écoles.

Ce rapport présente un système de gestion scolaire en ligne pour les écoles élémentaires qui vise à les aider à offrir aux étudiants, enseignants et administrateurs une expérience scolaire en ligne efficace et complète. Le système permet aux administrateurs de gérer les informations des enseignants et des étudiants, en plus aux comptes scolaires et aux classes. L'utilisation de ce système permet aux enseignants et aux élèves de visualiser leurs cours, les enseignants peuvent attribuer des devoirs et les élèves peuvent consulter les devoirs qui leur ont été attribués dans chaque matière.

Tout au long du processus de création de cette solution initiale pour ce problème, les tèchnologies de développement d'applications Web frontales etait exploré en travaillant avec un cadre d'application u tilisant des bibliothèques Angular et UI utilisant Angular Material et Nebular. Les tèchnologies de développement d'applications Web backend  etait également exploré en implémentant des services Web RESTful à l'aide de la requête HTTP via Express avec un environnement d'exécution en utilisant Node.js et sa bibliothèque npm pour les connecter aux données stockées dans une base de données No-Sql utilisant MongoDB.

# الملخص

في 17 مارس 2021، تأثر 171،341،667 متعلمًا بإغلاق المدارس استجابةً لوباء كوفيد -19، وفقًا لليونسكو. تزيد حاجة المدارس حول العالم إلى أنظمة التعليم عبر الإنترنت القادرة على تكرار وتعزيز التجربة المدرسية للمتعلمين المتأثرين ومعلميهم وإدارات المدارس.

يقدم هذا التقرير نظام إدارة المدارس عبر الإنترنت للمدارس الابتدائية الذي يهدف إلى مساعدتهم على تزويد الطلاب والمدرسين والمسؤولين بتجربة مدرسية فعالة وشاملة عبر الإنترنت، يسمح النظام للمسؤولين بإدارة معلومات المعلمين والطلاب، بالإضافة إلى حسابات المدرسة والصفوف. يتيح استخدام هذا النظام للمدرسين والطلاب مشاهدة فصولهم، ويمكن للمدرسين تعيين واجبات منزلية ويمكن للطلاب مشاهدة واجباتهم المنزلية في كل مادة.

خلال عملية بناء الحل في مراحله الأولية، تم تعلم تقنيات تطوير تطبيقات الويب للواجهة الأمامية من خلال العمل مع إطار عمل للتطبيق باستخدام مكتبات متخصصة بذلك مثل: Angular Nebular ول Angular Material . كما تم تعلم تقنيات تطوير تطبيقات

الويب الخلفية من خلال بناء خدمات الويب RESTful باستخدام طلبات HTTP من خلال Express مع بيئة وقت التشغيل باستخدام Node.jsومكتبة البرمجيات npm لربطها بالبيانات المخزنة في قاعدة بيانات No-Sql باستخدام MongoDB .

# Acknowledgements

I would like to express my special thanks of gratitude to my project supervisor Dr. Mohammad Hani AL JAMAL as well as my examiner Dr. Adolf ABDALLAH who gave me the golden opportunity to do this wonderful project on the topic of online education.

Secondly, I’d like to thank everyone that helped me finish this project within the limited time frame, thank you to Tripoli Entrepreneurs Club – TEC for teaching me the technical programming skills I’ve used to develop this system and thank you to Mostafa TLAIS for helping me out with technical programming tasks in the project.

Finally, to my caring, loving, and supportive mother, my deepest gratitude.

Thank you,

Najat

MIS – M2

# Table of content

## Contents

# General introduction

Most governments around the world have temporarily closed educational institutions in an attempt to contain the spread of the COVID-19 pandemic. According to UNESCO, On the 17th of March 2021, there are 171,341,667 learners affected by the closure of schools around the world, this number makes up 9.8% of total enrolled learners, with 29 country-wide closures.

Education technology is playing an important role in making education possible for those affected in these hard times. According to a report from Grand View Research, the global education technology market size was valued at USD 76.4 billion in 2019 and is expected to grow at a compound annual growth rate (CAGR) of 18.1% from 2020 to 2027 as more people look for learning resources online.

“E-Elementary” is an online school management system catered for elementary schools. This system can be used by elementary schools everywhere. The system helps elementary schools’ administrations to keep the schools’ information in one place and access them through a well-designed application. This project also includes a separate application that can be used by teachers and students to access their classes, it’s important to note that this application is in the earlier stages of its development.

We present the” E-Elementary” school system, its purpose, development approach, the features and functions that it offers and how the system works (its different use cases). This report will also include:

- **Chapter 1:** an overview about existing solutions and systems in the market will be presented in this chapter, as well as project in scope / out scope, adopted development approach and methods, and project development phases / planning and implementation plan.

- **Chapter 2**: this chapter will encompass a deep presentation about the system’s requirements, system main features, system actors, system use cases and an introduction about system design (using UML design Methods).

- **Chapter 3:** a full presentation about system design will be included in this chapter, a detailed content explanation will be added in order to explain database elements and its structure. This includes an explanation about NoSQL database, its advantages, why it was used in the system and how it was used. In addition to an introduction to MongoDB.

- **Chapter 4:** this chapter will be decomposed into two parts. The first part, will be an introduction about technical part of the project that includes the programming languages, frameworks, libraries and platforms used to build the two applications. The second part, will be dedicated to present all system views that make up the two application.

To finish up the report i will discuss some ways to improve it and get it to its full potential! And all relevant references used will be presented in the last page.

# Chapter 1: Project Overview

An overview about existing solutions and systems in the market will be presented in this chapter, as well as project in scope / out scope, adopted development approach and methods, and project development phases / planning and implementation plan.

## 1. Background

In an article released in September 2020, UNESCO cites how the organization has observed that most governments around the world have temporarily closed educational institutions in an attempt to contain the spread of the COVID-19 pandemic and this impacted over 60% of the world's student population. And in its 75th session of the UN Assembly Volkan Bozkir, the president of the session has expressed that "When education is interrupted, it affects everyone", and "all of us pay the price". The coronavirus pandemic has caused huge disruptions; but it has also given an opportunity to contribute in a much needed positive change to global education systems. This contribution is lead by Edtech solutions.

## 2. Existing Solutions

There are many new technologies being used in classrooms today: social networking, online learning, video conferencing, and all these technologies fall under the umbrella of educational technology or Edtech.

### 2.1. Edtech Solutions

Edtech or education technology (a mix between education and technology) refers to the applications developed to provide online solutions and tools in the education field.
Edtech Solutions understand what is needed to develop and use technology effectively in the classroom as well as for school management. These solutions include a wide range of technology such as: learning management systems, digital tools and media, information and communications tech and even schools' IT infrastructure and the devices that are used by students and teachers in

learning.

Edtech solutions aim to facilitate learning and improve schools performance by creating, using and managing new processes and resources. Edtech is playing a significant role in enabling teachers to teach students and enhancing the learning experience of students at distance.

### 2.2. Google Classroom

Google Classroom is great example of a virtual classroom that helps schools manage their classes' information online. Google classroom simplifies online teaching and learning by enabling schools to:

- Add students directly, or share a code or link so the whole class students can join
- Set up a class in minutes and create class work that appear on students' calendars
- Easily communicate with guardians and automatically send them updates

Google classroom help students move forward in their education by enabling them to;

- Store frequently used feedback in their comment banks for fast, personalized responses
- Grade consistently and transparently with rubrics integrated into student work
- Enable originality reports to let students scan their own work for potential plagiarism

Google classroom support communication and strengthen connection between students and teachers by enabling them to;

- Connect with their students from anywhere with a hybrid approach for in-class and virtual classes
- Communicate important announcements to the Stream page
- Enable face-to-face connections with students using Google Meet built into Classroom

## 3. "E-Elementary" Overview

The "E-Elementary" system that's going to be developed in this project is an online school information management system that enables schools to set their teachers, students and classes' information and access them through a web application. Using this application enables schools to set the school's classes, the sections available, and the subjects that are being taught in each one of

them and the students and teachers personal information with the classes they have.
As we have discussed earlier in this report, is composed of 2 applications, one can be used by the school’s administration and the other can be used by both teachers and students. And the backend of the project is composed of 5 web services that store the information of school accounts, teachers, students and classes.

## 3.1. Project in Scope

This project will consist of creating an online elementary school management system. The project will be completed in March, 2021. Modules of the system will include a simple classes’ view, a subjects’ view and a way for teachers to assign homework for the classes. The system will also modules for teachers, students, classes and school accounts information. Teachers and students can sign up to the system if they don’t have a school account and they can sign in with their school accounts.

## 3.2. Project out Scope

This project will not include modules for video conferencing to host online classes, chat, collaborative games, and a comment section. Any tasks associated with the cited modules are irrelevant to the project and will not be implemented.

## 2.1. Project Development Approach

Unlike the more traditional waterfall model, which focuses on a stringent step-by-step process of development stages, the iterative model is best thought of as a cyclical process. After an initial planning phase, a small handful of stages are repeated over and over, with each completion of the cycle incrementally improving and iterating on the software.

The iterative model was the approach taken to develop this system because it allows iterations in every step of the way. This is exactly what the system needs to keep up-to-date with the ongoing changes in the system itself and on the online learning solutions market. In addition to being able to perform consistent and frequent iterations, the iterative model makes it easy to organize tasks from planning and design through to implementation and testing with the ability to go back and make changes within a minimal time frame.

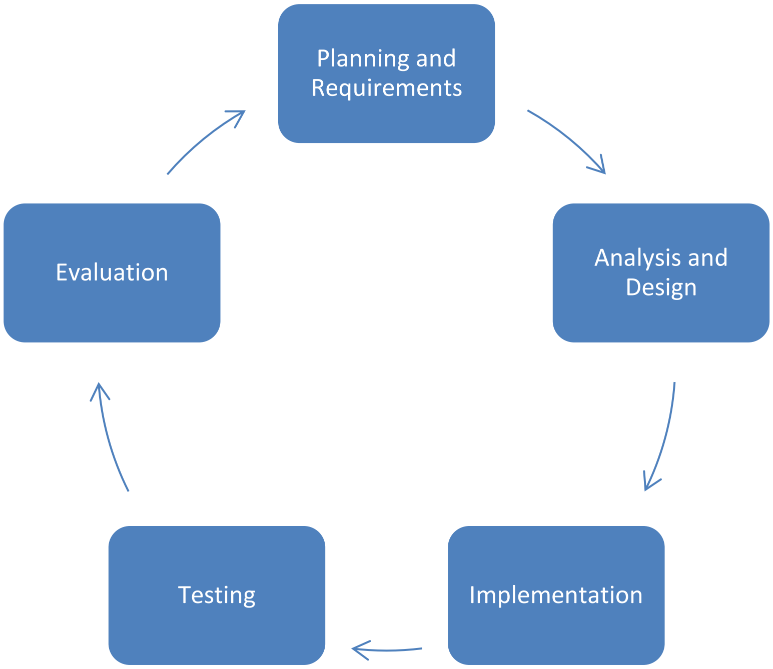


**Figure 1: The Iterative Model**

a) Planning & Requirements:

As with most any development project, the first step is go through an initial planning stage to research existing solutions in the market and map out the business requirements, project scope and out scope. And generally prepare for the upcoming stages of the cycle.

b) Analysis & Design:

Once planning is complete, an analysis is performed to nail down the appropriate business logic, database structure, system users and use cases and user interface design. The database models are developed in this phase and the views are designed. This phase also goes through the technical requirements of the project; the platforms that are going to be used to design, develop and test the project, the programming languages used, the different UI frameworks used for designing its views and the database host that will host the database models.

c) Implementation:

With the planning and analysis out of the way, the actual implementation and coding process can now begin. In this phase, the user interface design will be implemented, the system functionalities and business logic will be coded and the first iteration of the system is presented. This iteration will be revisited and modified to fit the evolving system and business needs.

d) Testing:

Once this current build iteration has been coded and implemented, the next step is to go through a series of testing procedures to identify and locate any bugs and issues and fixing them using the testing methods specified in the planning phase.

e) Evaluation:

Every time the previous stages are completed, the system is examined to identify where the project is at, where it needs to be, what can or should change which will result in another iteration.

## 2.2. Project Features

“E-Elementary” is an online school information management system that doesn’t only work as a virtual classroom but offers elementary schools features to help them manage their students and teachers information. Elementary schools can use this system to enable their administrators, students and teachers, each to:

| **Users** | **Features** |
|---|---|
| Teachers and students | Create a school account |
|  | Sign in to the system |
| Teachers | View classes |
|  | View subjects in each class |
|  | Assign students homework |
| Students | View classes |
|  | View subjects in each class |
|  | View the homework assigned to them in each class |
| Administrators | Manage school accounts |
|  | Manage school classes |
|  | Manage students’ and teacher’ information |

## 2.3. Implementation plan

| # | TASK |
|---|---|
| **1** | **Design phase** |
| 1.1 | Identification of business requirements (features and functions to implement) |
| 1.2 | Creating a use case diagram with all actors to differentiate between their roles and tasks in the system |
| 1.3 | Creating a class diagram to model data and figure out an appropriate structure |
| 1.4 | Building wireframes to identify the user interface design and content of the different views of the system |
| **2** | **Implementation phase** |
| **2.1** | **Admin application implementation** |
| 2.1.1 | Students module |
| 2.1.1.1 | Students information table |
| 2.1.2 | Teachers module |
| 2.1.2.1 | Teachers information table |
| 2.1.3 | Classes module |
| 2.1.3.1 | Classes information table |
| 2.1.4 | Users module |
| 2.1.4.1 | Users information table |
| **2.2** | **Teacher and student application implementation** |
| 2.2.1 | Registration module |
| 2.2.1.1 | Registration form |
| 2.2.2 | Login module |
| 2.2.2.1 | Login form |
| 2.2.3 | Classes module |
| 2.2.3.1 | Classes view |
| 2.2.4 | Subjects module |
| 2.2.4.1 | Subjects view |
| 2.2.5 | Homework module |
| 2.2.5.1 | Homework view |
| **3** | **Testing** |
| 3.1 | Applying a complete scenario in the two applications to identify issues |

| | |
|---|---|
| 3.2 | Testing the RESTful APIs with HTTP request using  Postman |
| 3.3 | Fixing the issues |

# Chapter 2: Project Specification

This chapter will encompass a deep presentation about the system's requirements, system main features, system actors, system use cases and an introduction about system design (using UML design Methods).

## 1. Business requirements

### 1.1. System Purpose

The "E-Elementary" online school system aims to help elementary schools store and manage their data online. This system can be used by teachers and students in their day to day school life to access their classes from the comfort of their home and enables school administration to update their schools' most important information regarding their teachers, students and classes.

Briefly put, the E-Elementary school system is a management information system for elementary schools that focusses on these main components:

- Registration of new students and teachers and storing their data
- Updating classes' sections and subjects
- Homework assignment

### 1.2. System Features

The system required consists of two web application which is consistent of 3 components: an admin component, a teacher's component and student's component.

Admin component: This component enables the school's **administration** to:

- View, search, add, edit and delete teachers information
- View, search, add, edit and delete students information
- View, search, add, edit and delete classes information
- View, search, add, edit and delete school accounts information

Teacher component: This component enables the school's **teachers** to:

- Create a school account
- View and select the classes and subjects they teach
- Assign and update homework

Student component: This component enables the school's **students** to:

- Create a school account
- View and select the classes and subjects they have
- View assigned homework

## 2. Use Case description

Use cases are a unified modeling language (UML) diagram. UML is effective for high-level conceptual data modeling, it's a way of visualizing a system and helps anyone understand its components. A use case diagram is a behavioral UML diagrams that describe the requirements of the system and represents its functionalities (what it does). A use case diagram is composed of use cases, actors and communicational links. The use cases represents the system's function, the actors represents the system's users that interact with the use cases and the communication links are the links between the actors and the use cases and represents how those actors are participating in the use cases.

### 2.1. Use Case Details

| Actor name | Description | Use Case |
|---|---|---|
| **Admin** | The admin actor represents the school's administration responsible about all administrative modules in this system, such as users' registration, students and teachers' and classes' information. | **User's registration:**<br>Manage account information.<br>**Student's information:**<br>Manage Personal and academic information.<br>**Teacher's information:**<br>Manage personal information and the classes, sections and subjects that they teach.<br>**Class' information:**<br>Manage classes' information (grades, sections and subjects of each class). |

| | | |
|---|---|---|
| **Teacher** | The teacher actor represents the school’s teachers responsible of the classes’ information modules, such as assigning homework for subjects. | **Register:**<br>Enter username and password to create a teacher school account.<br>**Login:**<br>Enter username and password to access the system as a teacher.<br>**Classes information:**<br>View available classes (grades, sections and subjects).<br>**Homework information:**<br>View and edit homework for their students. |
| **Student** | The Student actor represents the school’s students that view their class’ information and assigned homework. | **Register:**<br>Enter username and password to create a student school account.<br>**Login:**<br>Enter username and password to access the system as a student.<br>**Classes information:**<br>View available classes (grades, sections and subjects).<br>**Homework information:**<br>View the assigned homework for each subject in his/her class. |

## 2.2. Use Case Diagram

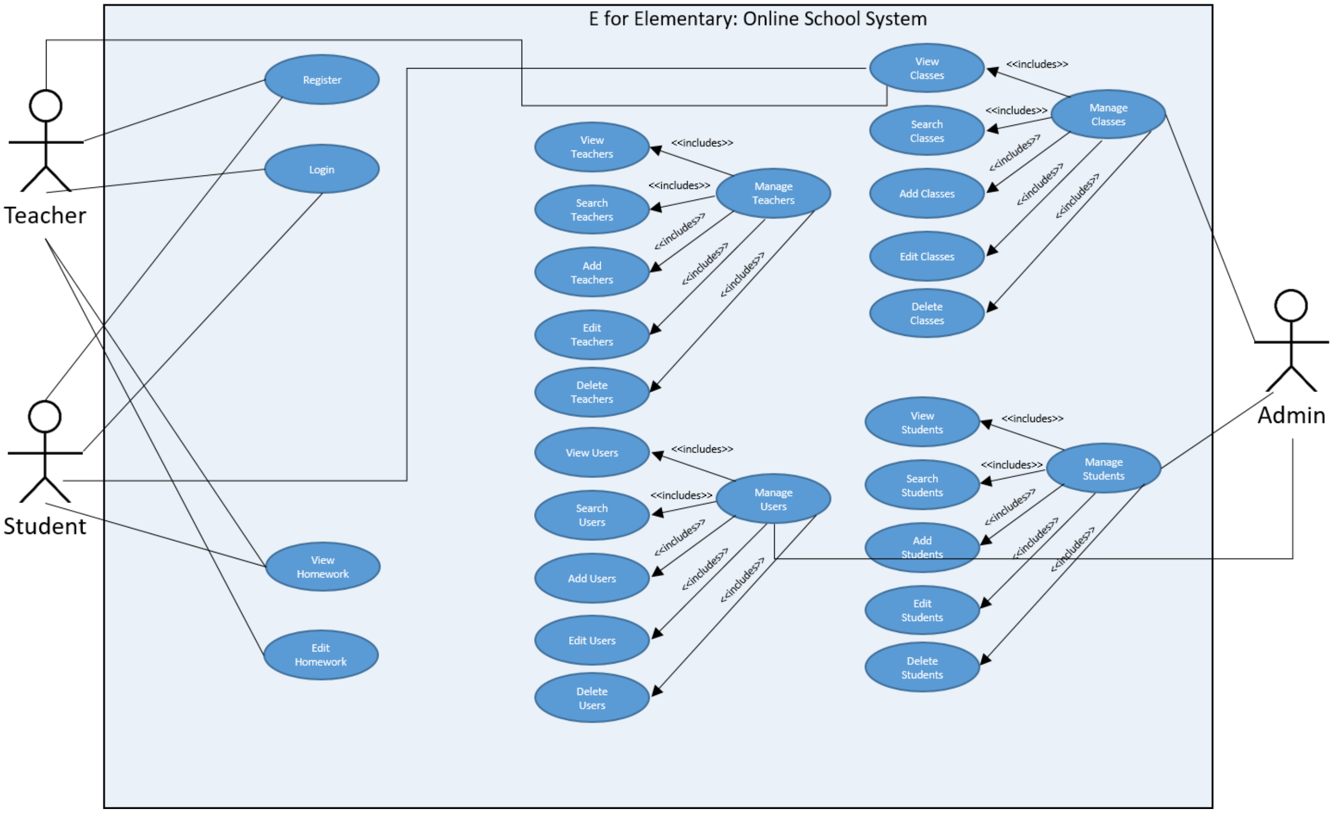


**Figure 2: Use Case Diagram**

# 3. Class Diagram View

A class diagram is another type of UML diagram that is used to model the structure of your database. It is composed of a set of classes that represents each data tables and the relationship between them. A class contains characteristics of an object (attributes) and its behavior (operations). Classes describe objects, while objects are usable instances of classes. Each Object was built from the same set of blueprints and therefore contains the same components (properties and methods) that a class has. And the relationship between those classes represents how they interact with each other.

## 3.1. Class diagram

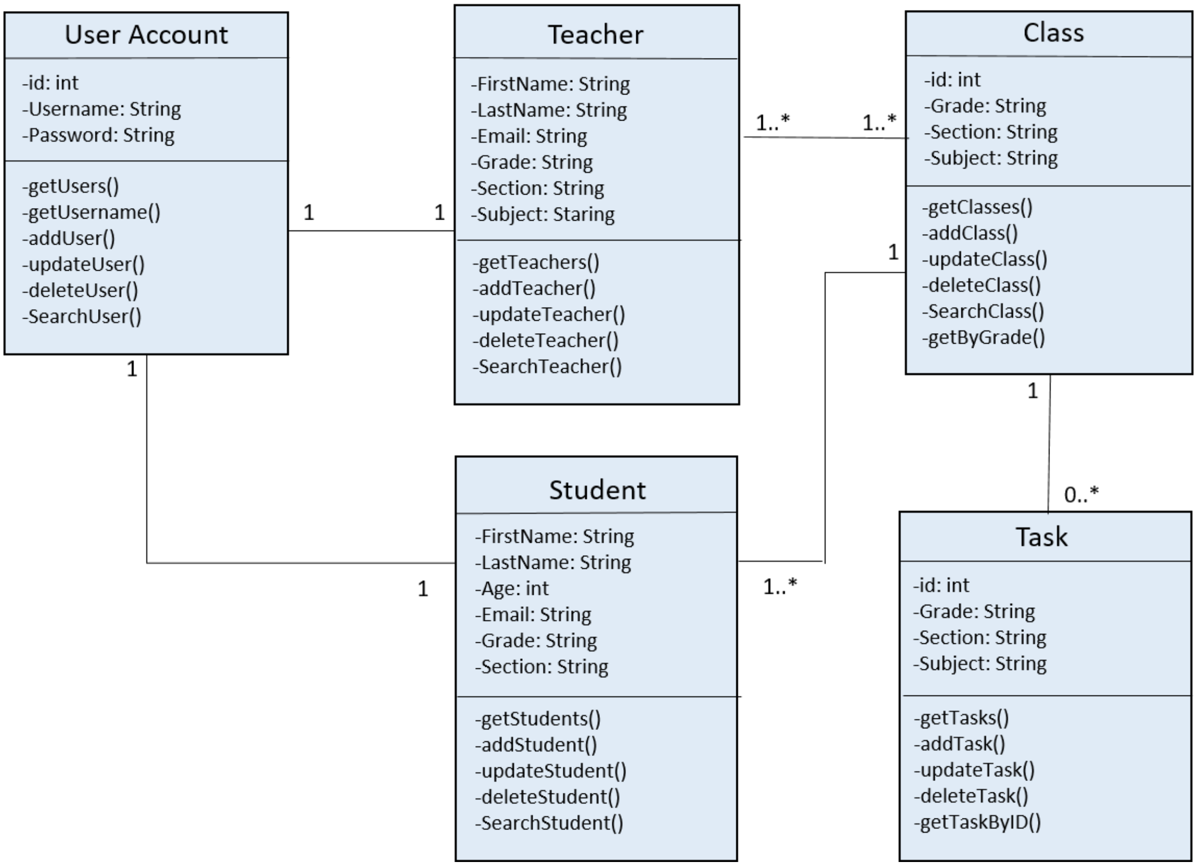


**Figure 3: Class diagram**

## 3.2. Relations and Classes Description

A class diagram consists of classes, attributes and functions. In this section the classes and attributes of the class diagram will be discussed and the relationship between the classes will be explained.

### 3.2.1. Classes and Attributes Description:

The Classes:

**User:** Account determines the users’ accounts registered in the system that can be either student or teacher.

**Class:** determines the classes defined within this system.

**Task:** determines the homework assigned in each class' subject of the system.

**Student:** determines the students registered in the system.

**Teacher:** determines the teachers registered in the system.

The Attributes:

**id:** determines the id of the system/s users, teachers and students.

**FirstName:** determines the first name of the students and teachers of this system.

**LastName:** determines the last name of the students and teachers of this system.

**Username:** determines the school account username of the users of the system.

**Password:** determines the school account password of the users of the system.

**Email:** determines the email of the system/s users, teachers and students.

**Age:** determines the age of the students of the system.

**Grade:** determines the grade of the classes of the system.

**Section:** determines the section available in the classes of the system.

**Subjects:** determines the subjects that are being taught in the classes of the system.

### 3.2.2. Relations Description

- A teacher and a student can create a school account and become a user in the system.
- A student can belong 1 class only and a class can have 1 to many students.
- A teacher can teach 1 to many classes and a class can have 1 to many teachers.
- A task is assigned to a class based on the subjects of that class and a subject can have 0 to many tasks (homework) and a task can be assigned to one subject only.

# Chapter 3: Project Design

A full presentation about system design will be included in this chapter. This includes an explanation about NoSQL database, its advantages, why it was used in the system and how it was used. In addition to an introduction to MongoDB. The database design and how it's implemented the system are going to be explained in details.

## 1. Database

This section presents an explanation about NoSQL, its advatages, why it's used in the system, Mongodb and NoSQL and how it's used in the development of the system.

### 1.1. Introduction to NoSQL Database

The NoSQL database defers from the traditional tables structure of an SQL database. A NoSQL database breaks away from the rows and columns of the relational database model and stores its data in a simplistic way in documents.

### 1.2. NoSQL Advantages

There a lot of advantages that come with taking the NoSQL route, some of these are:

- NoSQL is designed for data that doesn't have a fixed structure, this means that it supports and can process any kind of data.
- Building a NoSQL database allows developers to speed up the data design process and gives you the ability to change its structure whenever needed
- NoSQL databases often store data in a form that is similar to the objects used in applications, reducing the need for translation from the form the data is stored into the form the data takes in the code.
- NoSQL databases easy scales to adapt to huge volumes and complexity of cloud applications.

### 1.3. Why use NoSQL?

The first reason is its short learning curve because NoSQL doesn't require any knowledge in the ins and outs of structuring a system's database in the traditional relational way, with no use of tables, relationships in the implementation of the database. The data is stored in documents as is.

Another reason to why we chose NoSQL is the opportunity it gives to shorten the process of data design and development because of its emphasis on the physical database model that becomes the first and only step to implementing the database.

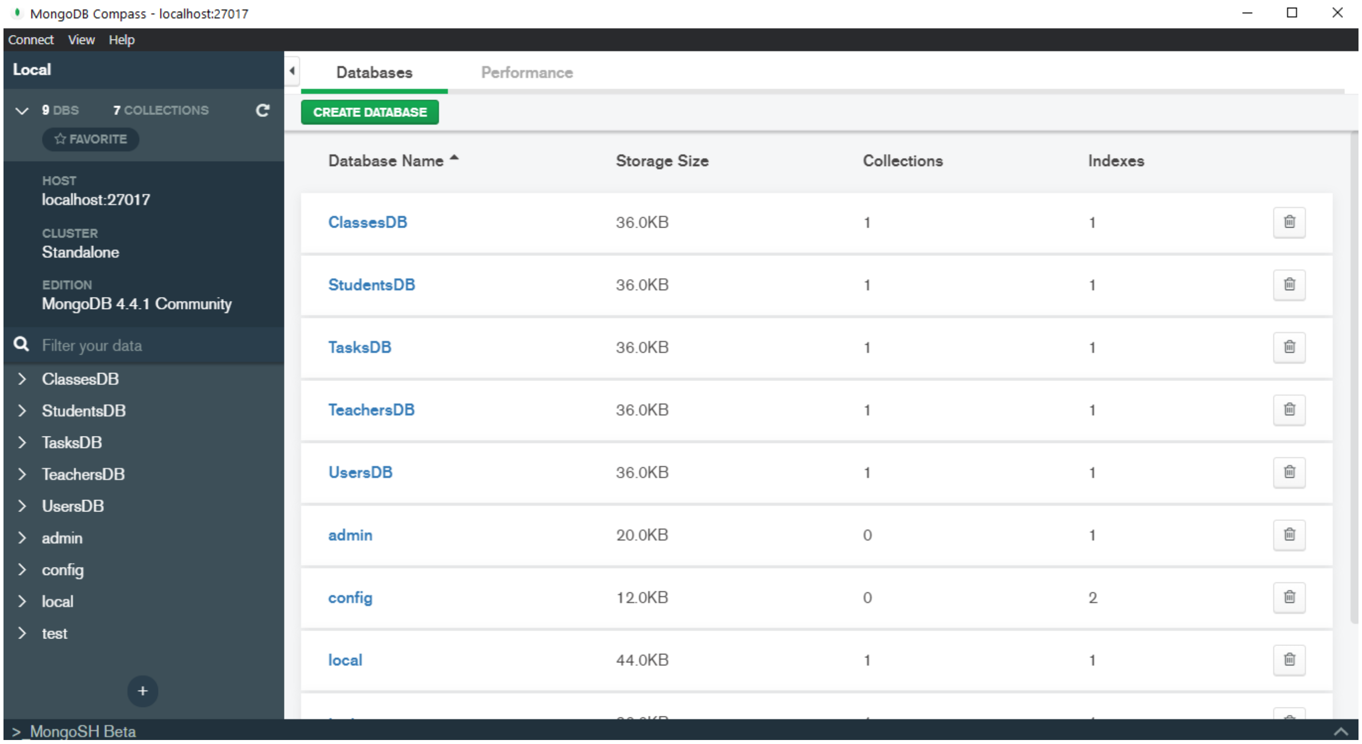
## 1.4. MongoDB and NoSQL

MongoDB is a NoSQL Database. MongoDB is an open-source NoSQL database. MongoDB is a document-based database. MongoDB stores data in JSON documents, these documents can difer in size and content.

Each MongoDB database contains collections and which in turn contains documents, this means that MongoDB is a schema-less database which flexible than traditional database tables.

## 1.5. E-Elementary and NoSQL

The system's data is distributed on 5 separate databases and stored in 5 collections (each database has one collection): the UsersDB, TeachersDB, StudentsDB, TasksDB and ClassesDB:



HOST
localhost:27017

CLUSTER
Standalone

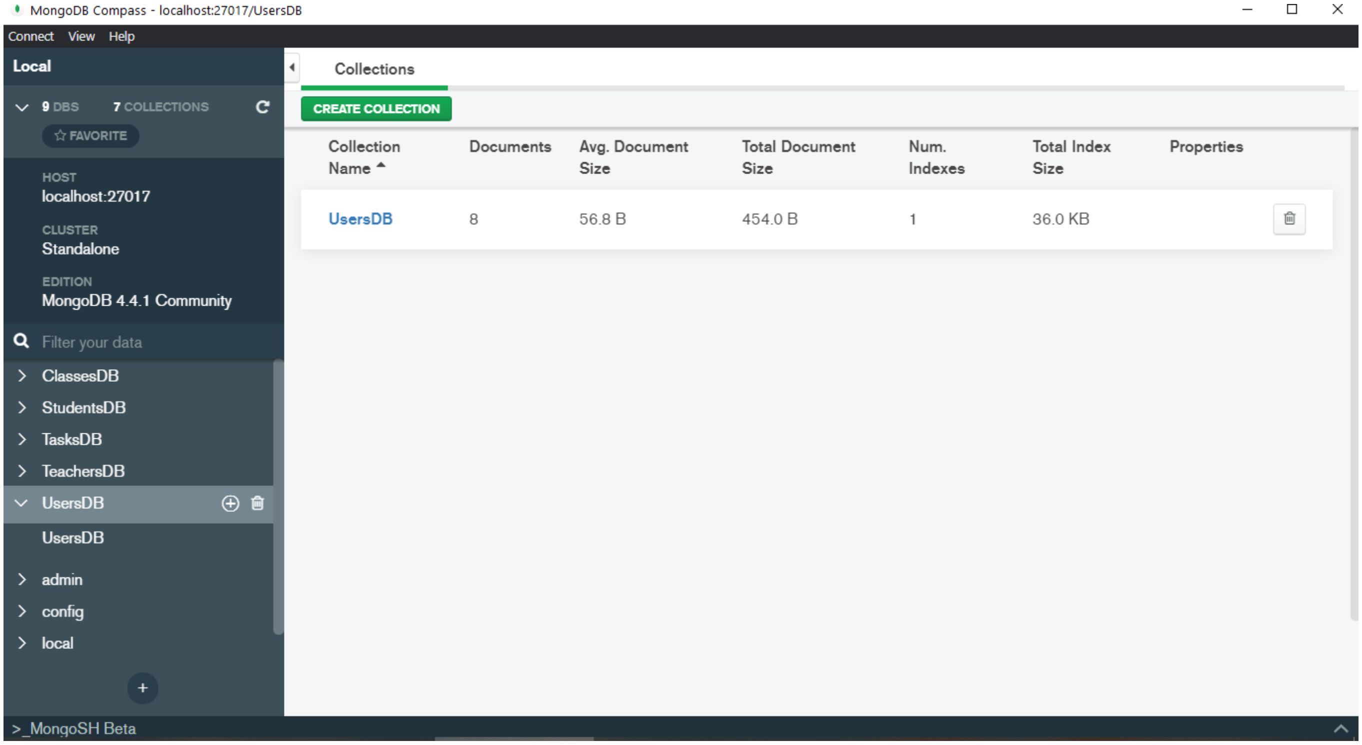
EDITION
MongoDB 4.4.1 Community

| Database Name ▲ | Storage Size | Collections | Indexes |
|---|---|---|---|
| ClassesDB | 36.0KB | 1 | 1 |
| StudentsDB | 36.0KB | 1 | 1 |
| TasksDB | 36.0KB | 1 | 1 |
| TeachersDB | 36.0KB | 1 | 1 |
| UsersDB | 36.0KB | 1 | 1 |
| admin | 20.0KB | 0 | 1 |
| config | 12.0KB | 0 | 2 |
| local | 44.0KB | 1 | 1 |

Figure 4: MongoDB Compass: System Database

a) UsersDB:

The users’ data is stored in the UsersDB collection in the UssersDB database:



HOST
localhost:27017

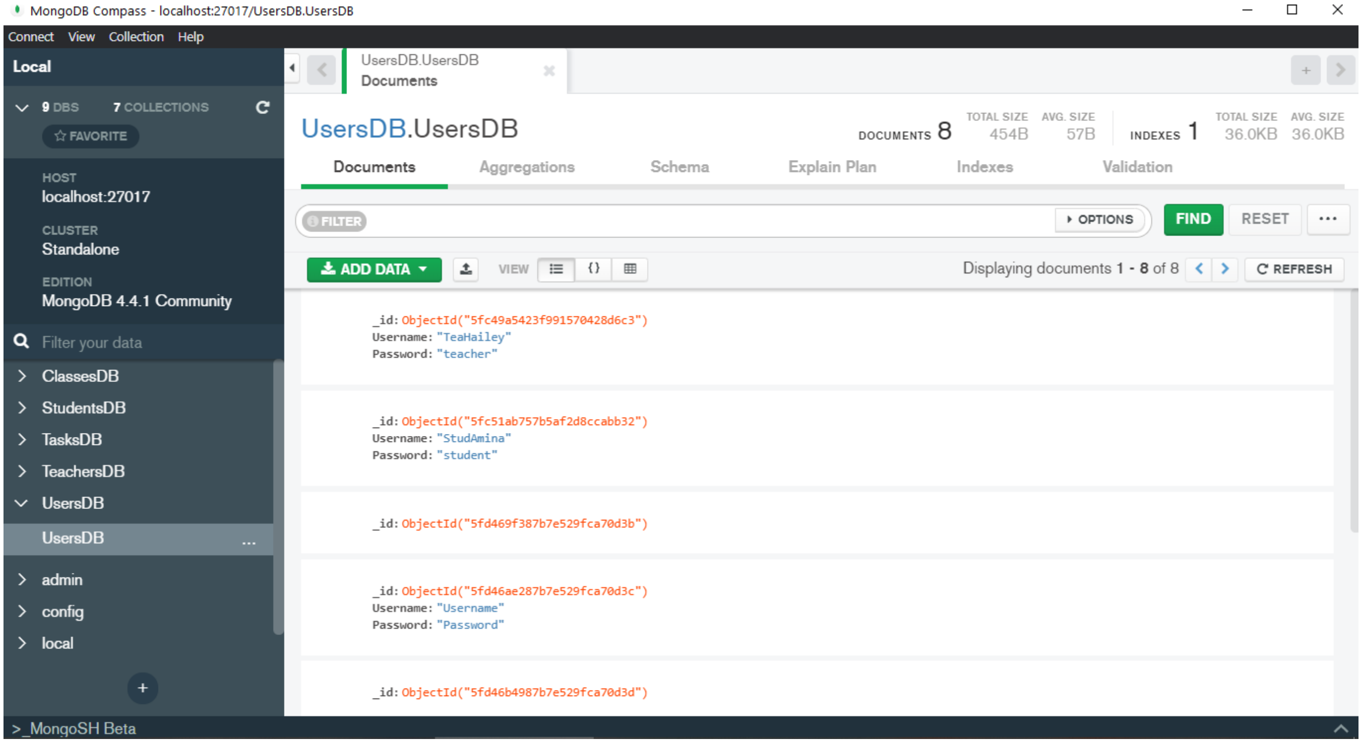
CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

| Collection Name | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|---|---|---|---|---|---|---|
| UsersDB | 8 | 56.8 B | 454.0 B | 1 | 36.0 KB | |

Figure 5: MongoDB Compass: UsersDB Database

In the UsersDB collection you can find the data of the registered school accounts in form of Json documents:



HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 4.4.1 Community

UsersDB.UsersDB — DOCUMENTS 8, TOTAL SIZE 454B, AVG. SIZE 57B — INDEXES 1, TOTAL SIZE 36.0KB, AVG. SIZE 36.0KB

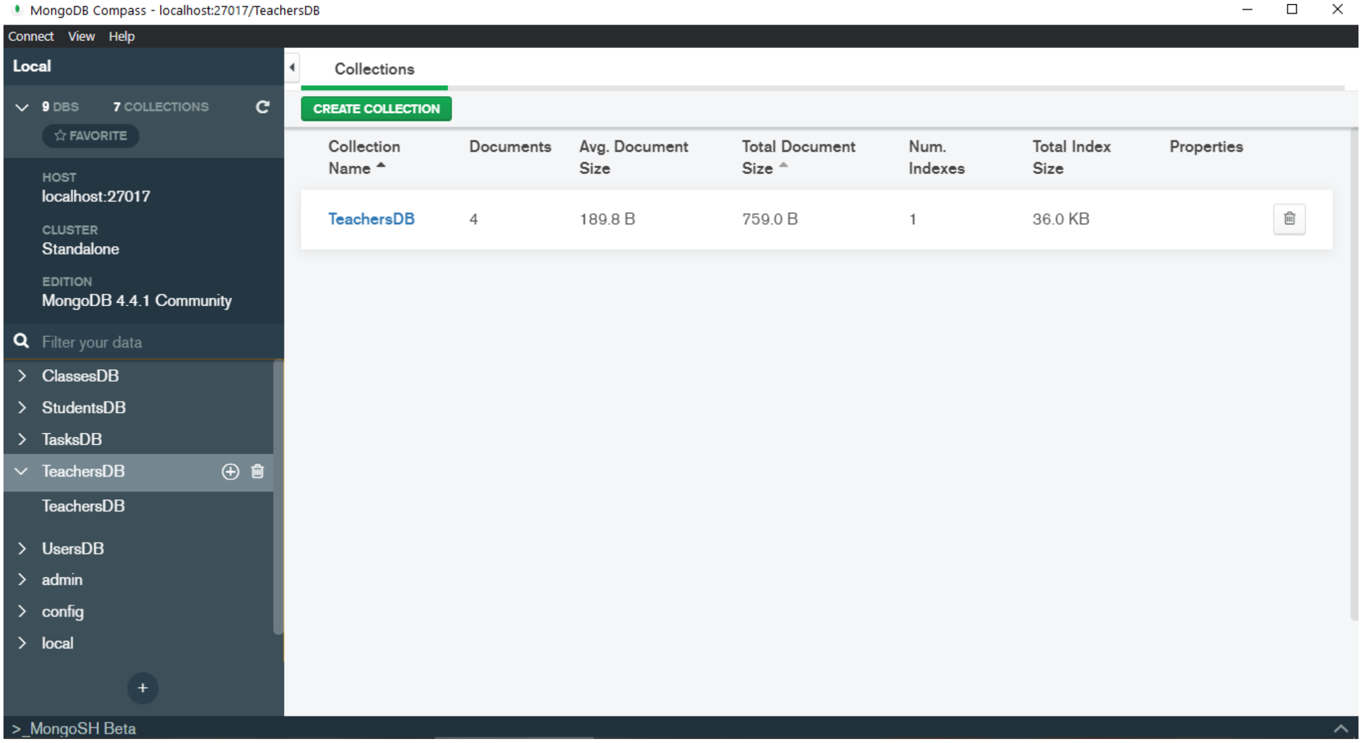
Displaying documents 1 - 8 of 8

```
_id: ObjectId("5fc49a5423f991570428d6c3")
Username: "TeaHailey"
Password: "teacher"

_id: ObjectId("5fc51ab757b5af2d8ccabb32")
Username: "StudAmina"
Password: "student"

_id: ObjectId("5fd469f387b7e529fca70d3b")

_id: ObjectId("5fd46ae287b7e529fca70d3c")
Username: "Username"
Password: "Password"

_id: ObjectId("5fd46b4987b7e529fca70d3d")
```

**Figure 6: MongoDB Compass: UsersDB Collection**

b) TeachersDB:

The teachers' data is stored in the TeachersDB collection in the TeachersDB database:



HOST localhost:27017

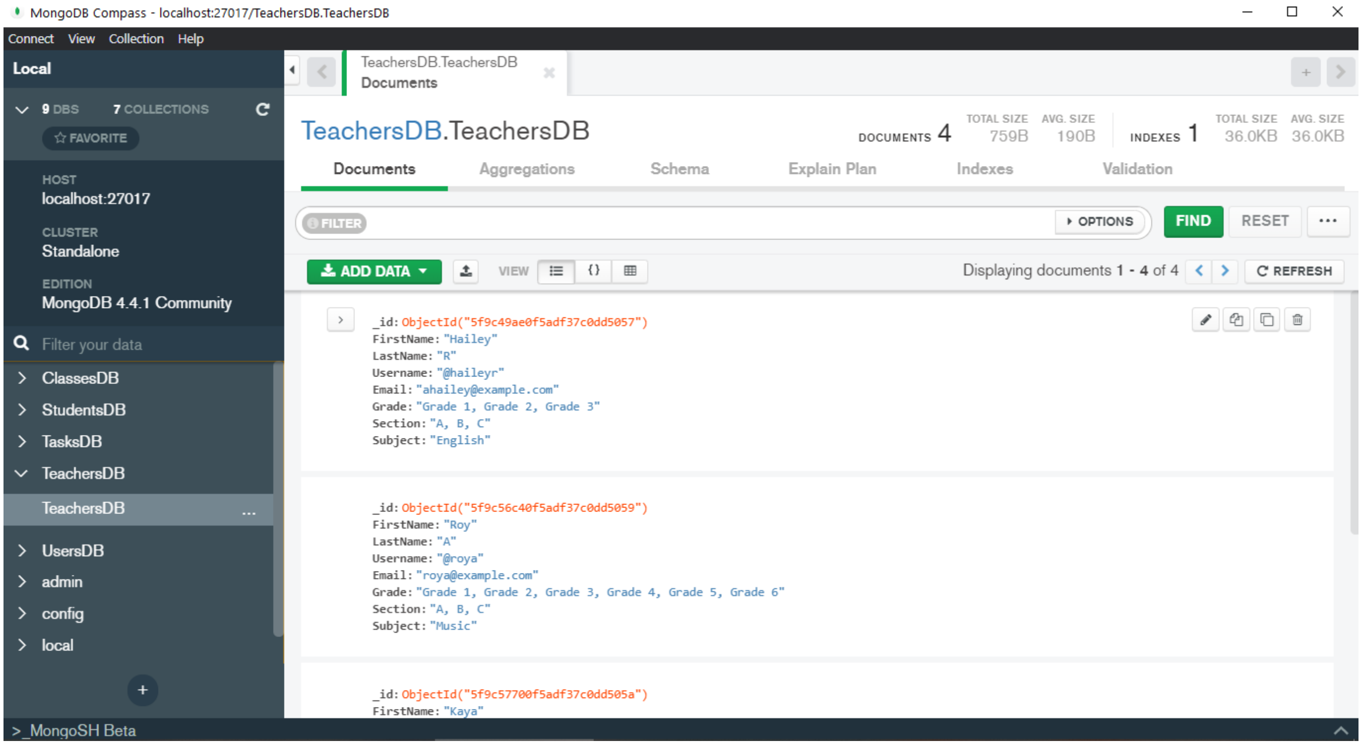
CLUSTER Standalone

EDITION MongoDB 4.4.1 Community

| Collection Name ▲ | Documents | Avg. Document Size | Total Document Size ▲ | Num. Indexes | Total Index Size | Properties |
|---|---|---|---|---|---|---|
| TeachersDB | 4 | 189.8 B | 759.0 B | 1 | 36.0 KB | |

**Figure 7: MongoDB Compass: TeachersDB Database**

In the TeachersDB collection you can find the data of the school's teachers in form of Json documents:

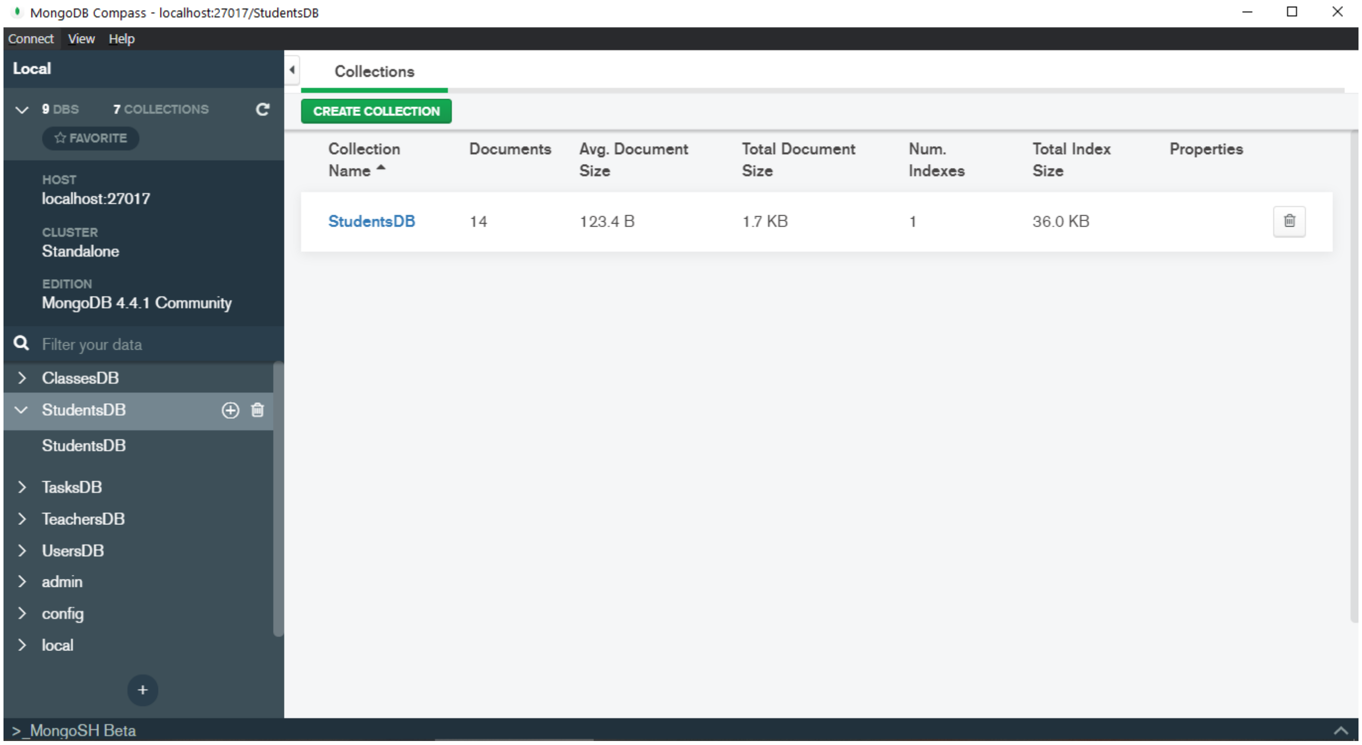


```
_id: ObjectId("5f9c49ae0f5adf37c0dd5057")
FirstName: "Hailey"
LastName: "R"
Username: "@haileyr"
Email: "ahailey@example.com"
Grade: "Grade 1, Grade 2, Grade 3"
Section: "A, B, C"
Subject: "English"

_id: ObjectId("5f9c56c40f5adf37c0dd5059")
FirstName: "Roy"
LastName: "A"
Username: "@roya"
Email: "roya@example.com"
Grade: "Grade 1, Grade 2, Grade 3, Grade 4, Grade 5, Grade 6"
Section: "A, B, C"
Subject: "Music"

_id: ObjectId("5f9c57700f5adf37c0dd505a")
FirstName: "Kaya"
```

**Figure 8: MongoDB Compass: TeachersDB Collection**

c) StudentsDB:

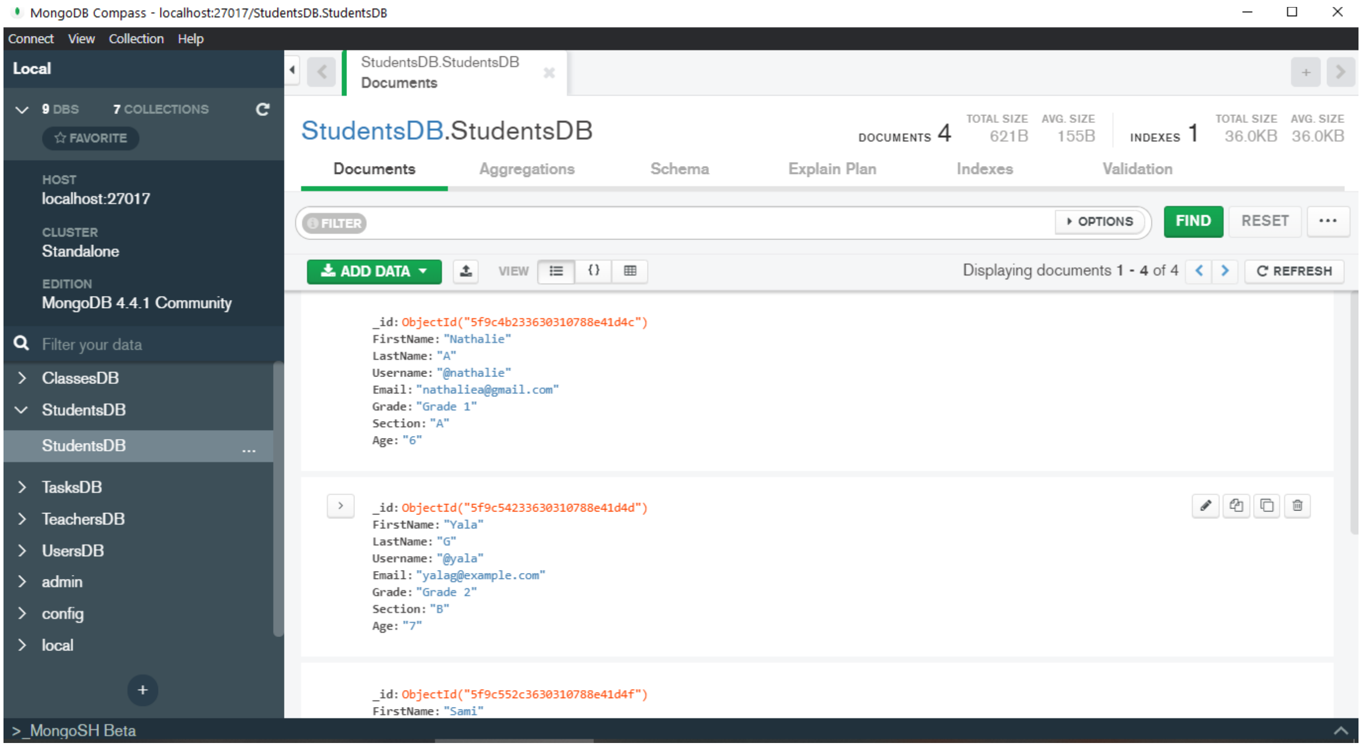
The students' data is stored in the StudentsDB collection in the StudentsDB database:



| Collection Name | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|---|---|---|---|---|---|---|
| StudentsDB | 14 | 123.4 B | 1.7 KB | 1 | 36.0 KB | |

Figure 9: MongoDB Compass: StudentsDB Database

In the StudentsDB collection you can find the data of the school’s students in form of Json documents:

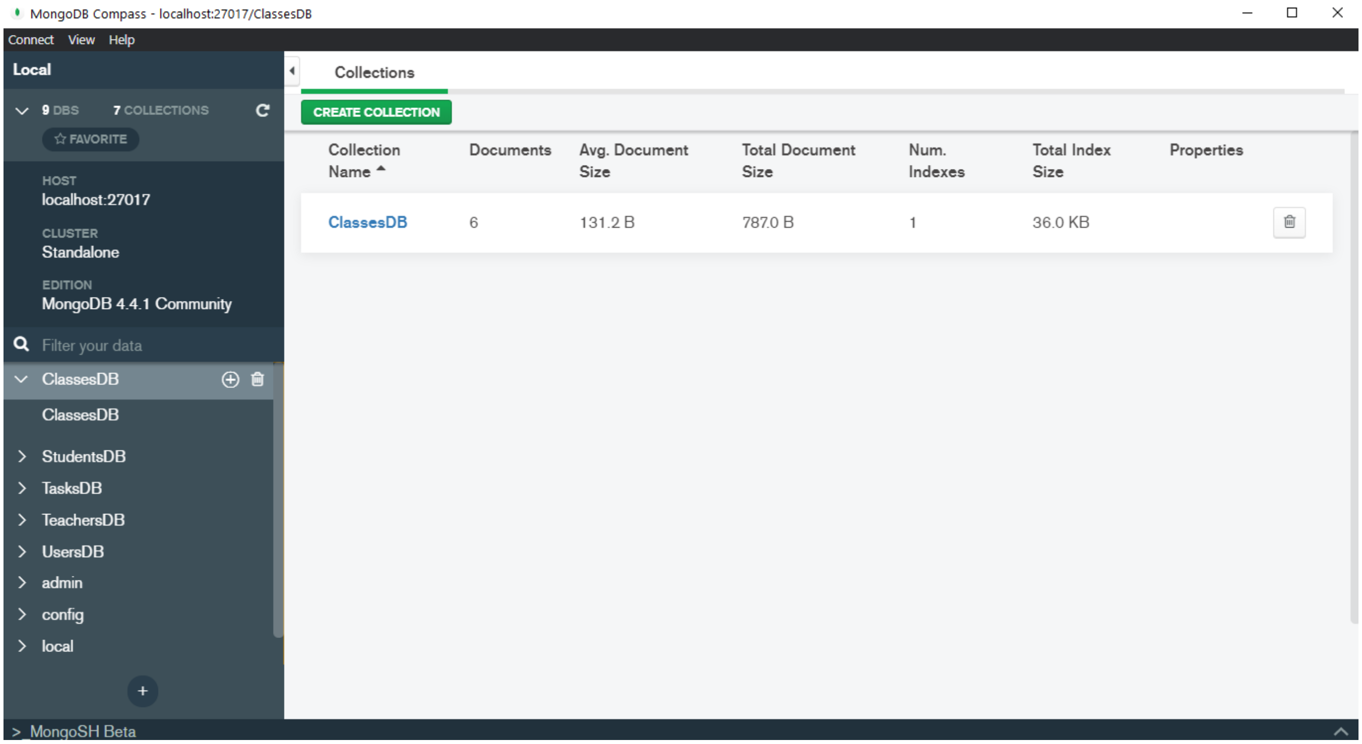


```
_id: ObjectId("5f9c4b233630310788e41d4c")
FirstName: "Nathalie"
LastName: "A"
Username: "@nathalie"
Email: "nathaliea@gmail.com"
Grade: "Grade 1"
Section: "A"
Age: "6"

_id: ObjectId("5f9c54233630310788e41d4d")
FirstName: "Yala"
LastName: "G"
Username: "@yala"
Email: "yalag@example.com"
Grade: "Grade 2"
Section: "B"
Age: "7"

_id: ObjectId("5f9c552c3630310788e41d4f")
FirstName: "Sami"
```

Figure 10: MongoDB Compass: StudentsDB Collection

d) ClassesDB:

The Classes' data is stored in the ClassesDB collection in the ClassesDB database:



HOST
localhost:27017

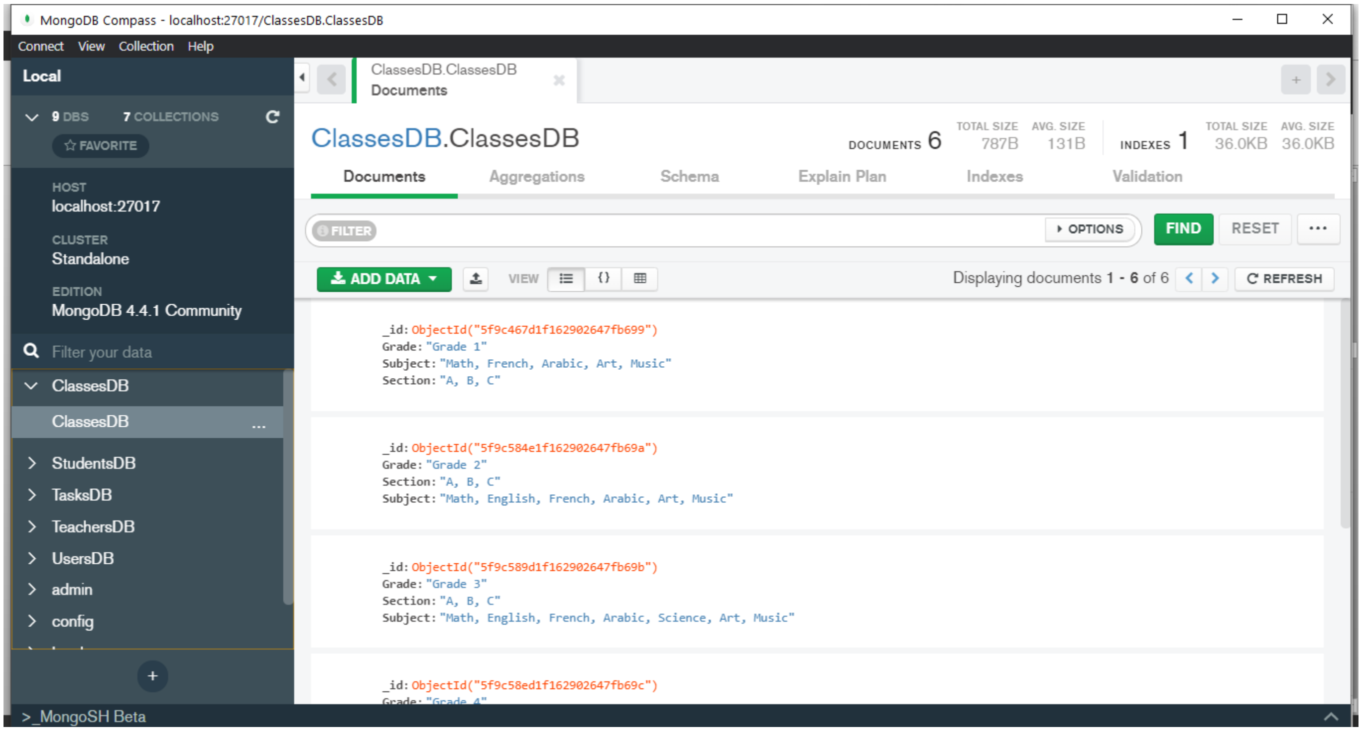
CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

| Collection Name ▲ | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|---|---|---|---|---|---|---|
| ClassesDB | 6 | 131.2 B | 787.0 B | 1 | 36.0 KB | |

Figure 11: MongoDB Compass: ClassesDB Database

In the ClassesDB collection you can find the data of the school's classes in form of Json documents:



HOST
localhost:27017

CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

DOCUMENTS 6 TOTAL SIZE 787B AVG. SIZE 131B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

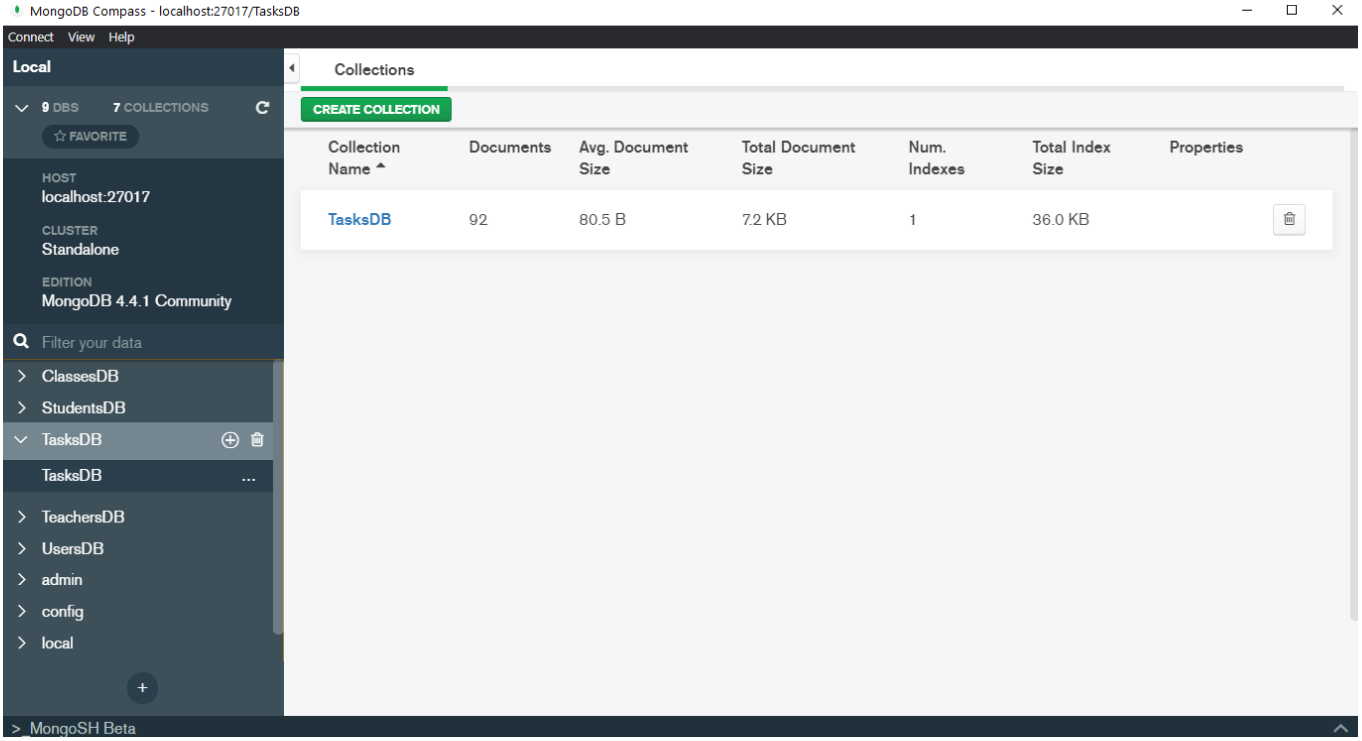
Displaying documents 1 - 6 of 6

```
_id: ObjectId("5f9c467d1f162902647fb699")
Grade: "Grade 1"
Subject: "Math, French, Arabic, Art, Music"
Section: "A, B, C"

_id: ObjectId("5f9c584e1f162902647fb69a")
Grade: "Grade 2"
Section: "A, B, C"
Subject: "Math, English, French, Arabic, Art, Music"

_id: ObjectId("5f9c589d1f162902647fb69b")
Grade: "Grade 3"
Section: "A, B, C"
Subject: "Math, English, French, Arabic, Science, Art, Music"

_id: ObjectId("5f9c58ed1f162902647fb69c")
Grade: "Grade 4"
```

Figure 12: MongoDB Compass: ClassesDB Collection

e) TasksDB:

The data about the homework and the subjects they are assigned in is stored in the TasksDB collection in the TasksDB database:



HOST
localhost:27017

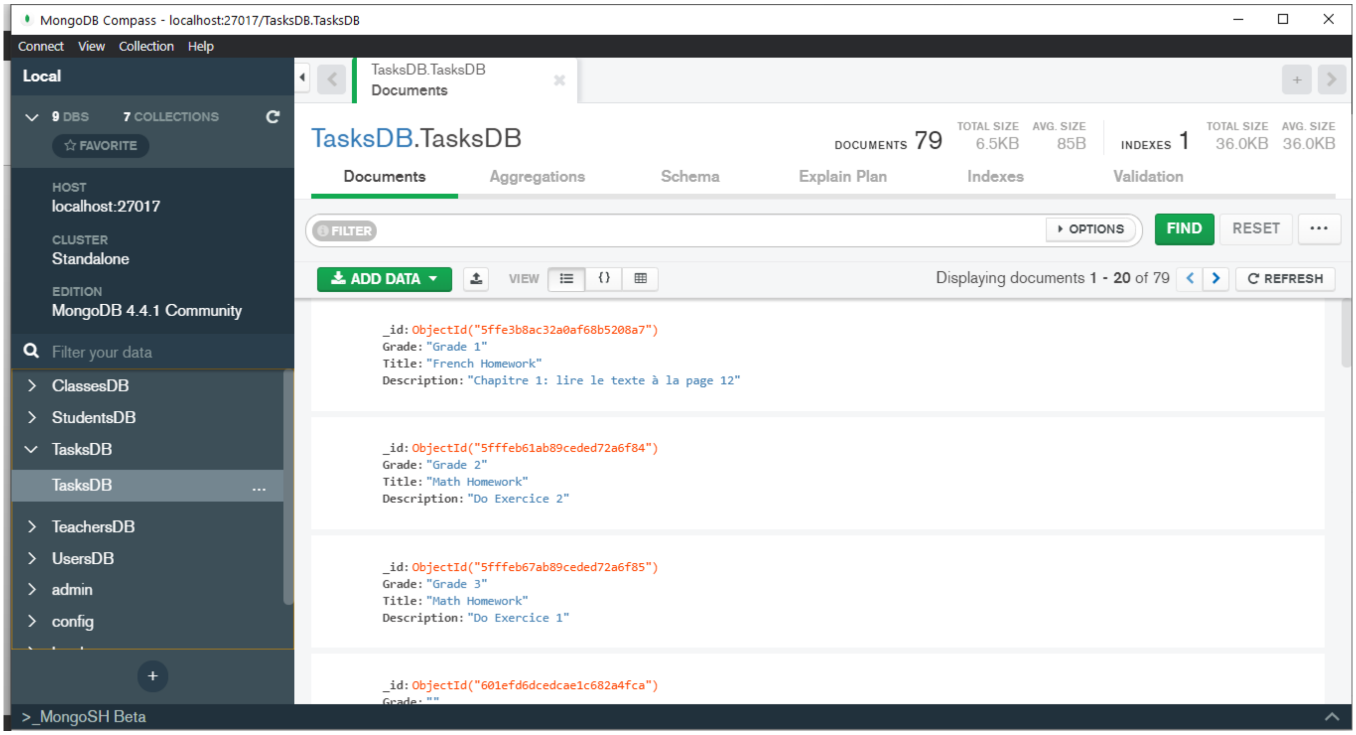
CLUSTER
Standalone

EDITION
MongoDB 4.4.1 Community

| Collection Name | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|---|---|---|---|---|---|---|
| TasksDB | 92 | 80.5 B | 7.2 KB | 1 | 36.0 KB | |

Figure 13: MongoDB Compass: TasksDB Database

In the TasksDB collection you can find the data of the school's classes' assigned homework in each subject in form of Json documents:



```
_id: ObjectId("5ffe3b8ac32a0af68b5208a7")
Grade: "Grade 1"
Title: "French Homework"
Description: "Chapitre 1: lire le texte à la page 12"

_id: ObjectId("5fffeb61ab89ceded72a6f84")
Grade: "Grade 2"
Title: "Math Homework"
Description: "Do Exercice 2"

_id: ObjectId("5fffeb67ab89ceded72a6f85")
Grade: "Grade 3"
Title: "Math Homework"
Description: "Do Exercice 1"

_id: ObjectId("601efd6dcedcae1c682a4fca")
Grade: ""
```

Figure 14: MongoDB Compass: TasksDB Database

# Chapter 4: Implementation View

This chapter will be decomposed into 2 parts. The first part, will be an introduction about technical part of the project that includes the programming languages, frameworks, libraries and platforms used to build the two applications. The second part, will be dedicated to present all system views that make up the two application.

## 1. Platform and methods installation

### 1.1. Plateform and library used

**Microsoft Visual Studio Code**

Microsoft Visual Studio Code is a lightweight yet powerful open source code editor. It offers a cross-platform IDE (integrated development environment) that supports 37 programming languages that can be used in:

- Data analysis: Python and R.
- Web application programming: Javascript, Typescript, HTML and CSS.
- Backend: Json and XM.
- macOS desktop application programming: Swift and objective-C.
- Windows desktop application programming: C++, VB.net and C# (supported through extensions).

This platform can be downloaded on Windows, Linux and macOS. VSCode also offers a runtime environment where you can run Node.js and connect to git.

**MongoDB**

MongoDb is a NoSQL database that replaces the traditional way to store data in tables by a modern document-oriented approach. Instead of having your information stored in tables and setting relationships between them, you have your information in collection that contain sets of documents and function which is the equivalent of relational database tables. MongoDB can be used on the web through MongoDB Atlas or locally on your desktop.

### Node.js

Node.js is a runtime environment that allows software developers to launch both the frontend and backend of web apps using JavaScript on the server. You can run your application using Node.js on macOS, Microsoft Windows, and Linux. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

### Express

Express or express.js is a Node.js backend framework. Express is used on top of Node.js and is what enables developers to build REST APIs because it gives them all the tooling they need to actually build HTTP servers. Express handles all the interactions between the frontend and the database, ensuring a smooth transfer of data to the end user.

### Angular

Angular is a frontend Javascript framework that uses Typescript and HTML to generate web applications. Angular helps you build interactive and dynamic single page applications (SPAs) with its features that include:

- Templating
- Two ways binding
- RESTFUL APIs

Angular has a unique application architecture, an app is made of components declared in the NgModule and each component is composed of three files an HTML file that handles the UI, a CSS file for styling and a Typescript file that handles the application's data.

### Nebular

Nebular is an Angular 10 customizable UI library. Nebular helps you design a modern dashboard by offering you 4 visual themes to choose from and more than 35 Angular built in UI components. You can install and add Nebular to your web application using the Angular CLI by running: ng add @nebular/theme.

In my admin application I've used Nebular's Ngx-admin admin based template which is the most popular and trusted Angular open source dashboard template is out there. Used by hundreds of thousands developers worldwide and Fortune 500 companies*. Some of the UI components that this template includes:

- **Tables:** ng-table
- **Charts:** ng-chart

### Angular Material

Angular Material is an Angular UI component library that helps you design modern web pages with a material design, the same design style that is used by Google. You can install and add Angular Material to your web application using the Angular CLI by running: ng add @angular/material.

Angular Material offers you all the UI components you need in its modern material design style. Some of its components that I've used in my teacher and student application are:

- **Cards:** mat-card
- **Buttons:** mat-button
- **Lists:** mat-list

### Postman

Postman is an API client that is used to develop and test your APIs through HTTP requests:  GET, POST, PUT, FETCH and DELETE.Using Postman will help you develop your RESTful APIs faster with its easy to use interface and features all you have to do is plug the route into the address bar, select the HTTP request you want to execute and press Send.

## 1.2. The methods of installation and activation

### 1.2.1. Download and install .MongoDB

To install MongoDB 4.4 Community Edition on your PC, go to the mongoDB website and download the MIS file .msi.

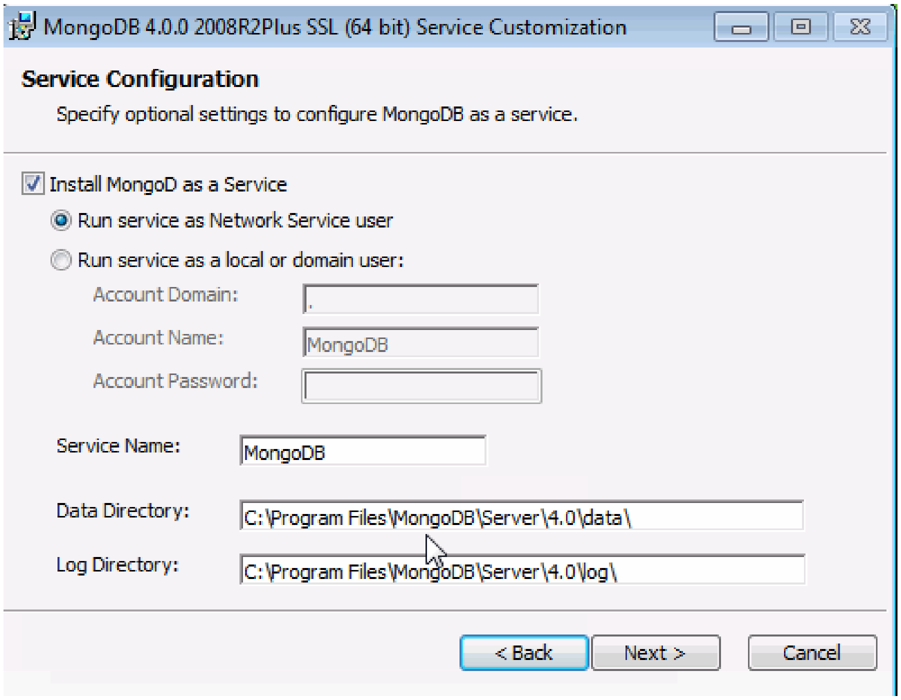
- **Choose Setup Type**

You can download the complete version or you can customize MongoDb by choosing the custom version. Choose the Complete version for now.

- **Service Configuration**

The following installs and configures MongoDB as a Windows service. If you don’t want to install it as a service uou can uncheck the Install MongoDB as a Service checkbox.

**Figure 13: MongoDB Compass: TasksDB Database** **Figure 13: MongoDB Compass: TasksDB Database**



MongoDB 4.0.0 2008R2Plus SSL (64 bit) Service Customization

**Service Configuration**

Specify optional settings to configure MongoDB as a service.

☑ Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

Account Domain: .

Account Name: MongoDB

Account Password:

Service Name: MongoDB

Data Directory: C:\Program Files\MongoDB\Server\4.0\data\

Log Directory: C:\Program Files\MongoDB\Server\4.0\log\

Figure 15: MongoDB Installation Dialog Box

- **Install MongoDB Compass**

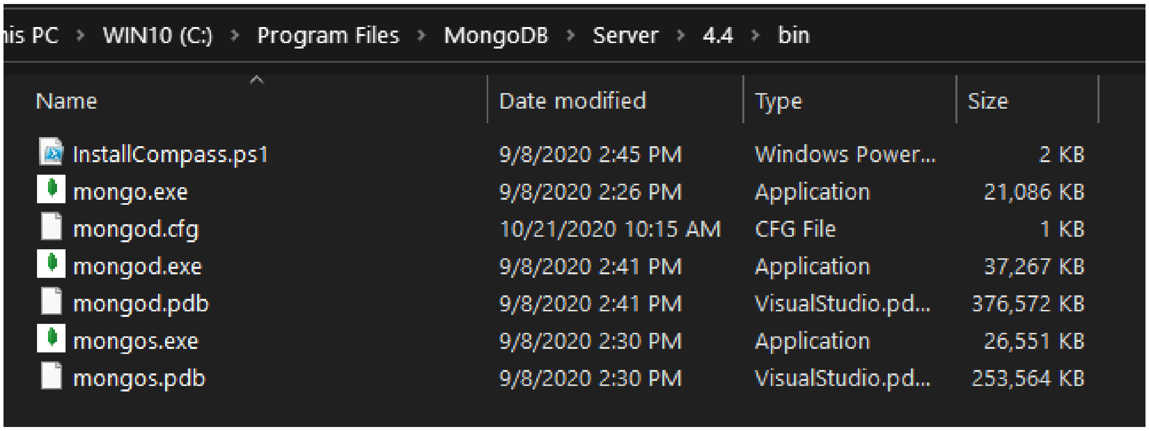
Optional. To have the wizard install MongoDB Compass, select Install MongoDB Compass (Default). I recomoend installing Compass bcause it gives you a GUI view of your database.

- **Finally, click Install.**

### 1.2.2. Open MongoDB

- **Find MongoDB**

Go to: C:\Program Files\MongoDB\Server\4.4\bin\



is PC › WIN10 (C:) › Program Files › MongoDB › Server › 4.4 › bin

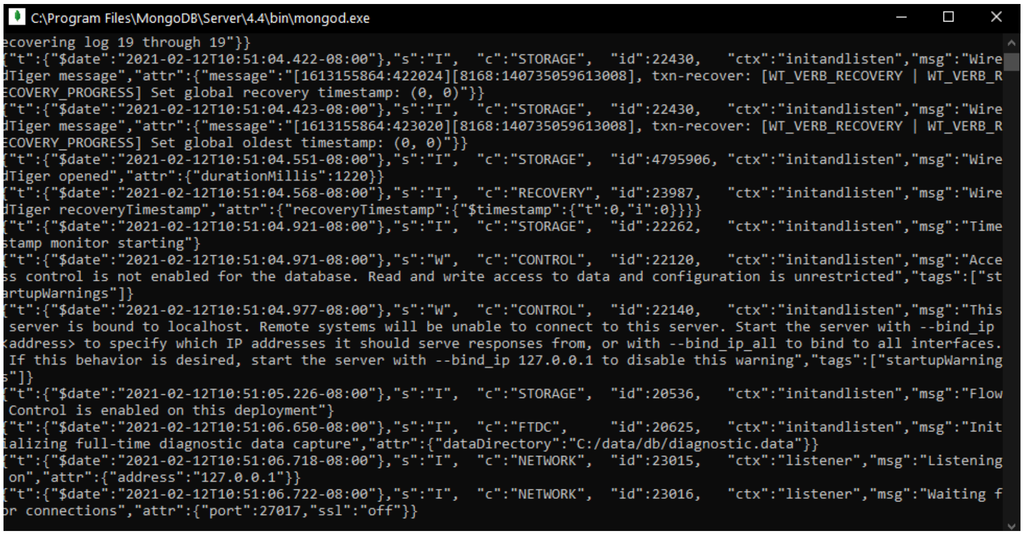
| Name | Date modified | Type | Size |
|---|---|---|---|
| InstallCompass.ps1 | 9/8/2020 2:45 PM | Windows Power... | 2 KB |
| mongo.exe | 9/8/2020 2:26 PM | Application | 21,086 KB |
| mongod.cfg | 10/21/2020 10:15 AM | CFG File | 1 KB |
| mongod.exe | 9/8/2020 2:41 PM | Application | 37,267 KB |
| mongod.pdb | 9/8/2020 2:41 PM | VisualStudio.pd... | 376,572 KB |
| mongos.exe | 9/8/2020 2:30 PM | Application | 26,551 KB |
| mongos.pdb | 9/8/2020 2:30 PM | VisualStudio.pd... | 253,564 KB |

Figure 16: MongoDB Instance

If you only installed the executable and did not install MongoDB as a Windows service, you must manually start the MongoDB instance.

- ➢ **Start the MongoDB background process**

Run the mongod.exe file as administrator. This is necessary to do first before running MongoDB.

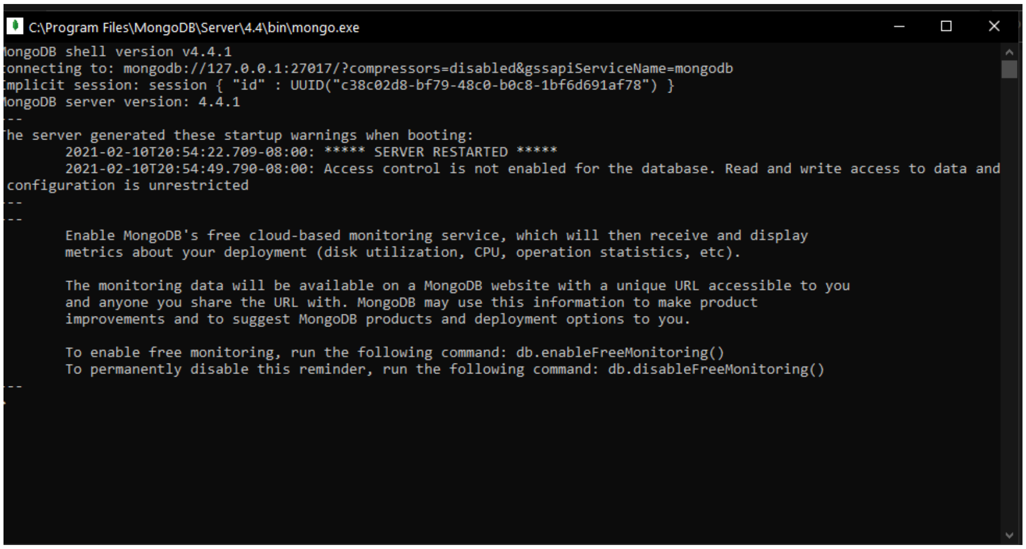


```
ecovering log 19 through 19"}}
{"t":{"$date":"2021-02-12T10:51:04.422-08:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"Wire
dTiger message","attr":{"message":"[1613155864:422024][8168:140735059613008], txn-recover: [WT_VERB_RECOVERY | WT_VERB_R
ECOVERY_PROGRESS] Set global recovery timestamp: (0, 0)"}}
{"t":{"$date":"2021-02-12T10:51:04.423-08:00"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":"initandlisten","msg":"Wire
dTiger message","attr":{"message":"[1613155864:423020][8168:140735059613008], txn-recover: [WT_VERB_RECOVERY | WT_VERB_R
ECOVERY_PROGRESS] Set global oldest timestamp: (0, 0)"}}
{"t":{"$date":"2021-02-12T10:51:04.551-08:00"},"s":"I",  "c":"STORAGE",  "id":4795906, "ctx":"initandlisten","msg":"Wire
dTiger opened","attr":{"durationMillis":1220}}
{"t":{"$date":"2021-02-12T10:51:04.568-08:00"},"s":"I",  "c":"RECOVERY", "id":23987,   "ctx":"initandlisten","msg":"Wire
dTiger recoveryTimestamp","attr":{"recoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2021-02-12T10:51:04.921-08:00"},"s":"I",  "c":"STORAGE",  "id":22262,   "ctx":"initandlisten","msg":"Time
stamp monitor starting"}
{"t":{"$date":"2021-02-12T10:51:04.971-08:00"},"s":"W",  "c":"CONTROL",  "id":22120,   "ctx":"initandlisten","msg":"Acce
ss control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}
{"t":{"$date":"2021-02-12T10:51:04.977-08:00"},"s":"W",  "c":"CONTROL",  "id":22140,   "ctx":"initandlisten","msg":"This
 server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip
<address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces.
 If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarning
s"]}
{"t":{"$date":"2021-02-12T10:51:05.226-08:00"},"s":"I",  "c":"STORAGE",  "id":20536,   "ctx":"initandlisten","msg":"Flow
 Control is enabled on this deployment"}
{"t":{"$date":"2021-02-12T10:51:06.650-08:00"},"s":"I",  "c":"FTDC",     "id":20625,   "ctx":"initandlisten","msg":"Init
ializing full-time diagnostic data capture","attr":{"dataDirectory":"C:/data/db/diagnostic.data"}}
{"t":{"$date":"2021-02-12T10:51:06.718-08:00"},"s":"I",  "c":"NETWORK",  "id":23015,   "ctx":"listener","msg":"Listening
 on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2021-02-12T10:51:06.722-08:00"},"s":"I",  "c":"NETWORK",  "id":23016,   "ctx":"listener","msg":"Waiting f
or connections","attr":{"port":27017,"ssl":"off"}}
```

Figure 17: MongoDB Background Process Command Line: mongod.exe

- ➢ **Start the MongoDB**

After running the mongod.exe file, it’s time to run mongo.exe and start a MongoDB instance.



```
ongoDB shell version v4.4.1
onnecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
mplicit session: session { "id" : UUID("c38c02d8-bf79-48c0-b0c8-1bf6d691af78") }
ongoDB server version: 4.4.1
--
he server generated these startup warnings when booting:
       2021-02-10T20:54:22.709-08:00: ***** SERVER RESTARTED *****
       2021-02-10T20:54:49.790-08:00: Access control is not enabled for the database. Read and write access to data and
 configuration is unrestricted
--
--
       Enable MongoDB's free cloud-based monitoring service, which will then receive and display
       metrics about your deployment (disk utilization, CPU, operation statistics, etc).

       The monitoring data will be available on a MongoDB website with a unique URL accessible to you
       and anyone you share the URL with. MongoDB may use this information to make product
       improvements and to suggest MongoDB products and deployment options to you.

       To enable free monitoring, run the following command: db.enableFreeMonitoring()
       To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
--
```

Figure 18: MongoDB Command Line: mmongo.exe

### ➢ Create a collection

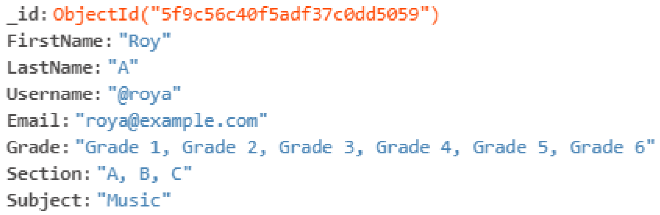
Now you have a MongoDB database, you can create a collection in the database by using the use COLLECTION NAME command.

### ➢ Insert documents

Now you have a MongoDB collection, you can insert your documents using the db.collection.insert() command, pass it the COLLECTION NAME and add your json document the fields  you need to have with their values, for example:

To add a teacher in the E-Elementary school system you run:

db.TeachersDB.insert({FirstName:"Roy",LastName:"A", Username:"@roya", Email:"roya@example.com",Class:"Grade 1, Grade 2, Grade 3, Grade4, Grade 5, Grade 6",Section:”A, B, C”, Subject:"Music"})



```
_id: ObjectId("5f9c56c40f5adf37c0dd5059")
FirstName: "Roy"
LastName: "A"
Username: "@roya"
Email: "roya@example.com"
Grade: "Grade 1, Grade 2, Grade 3, Grade 4, Grade 5, Grade 6"
Section: "A, B, C"
Subject: "Music"
```

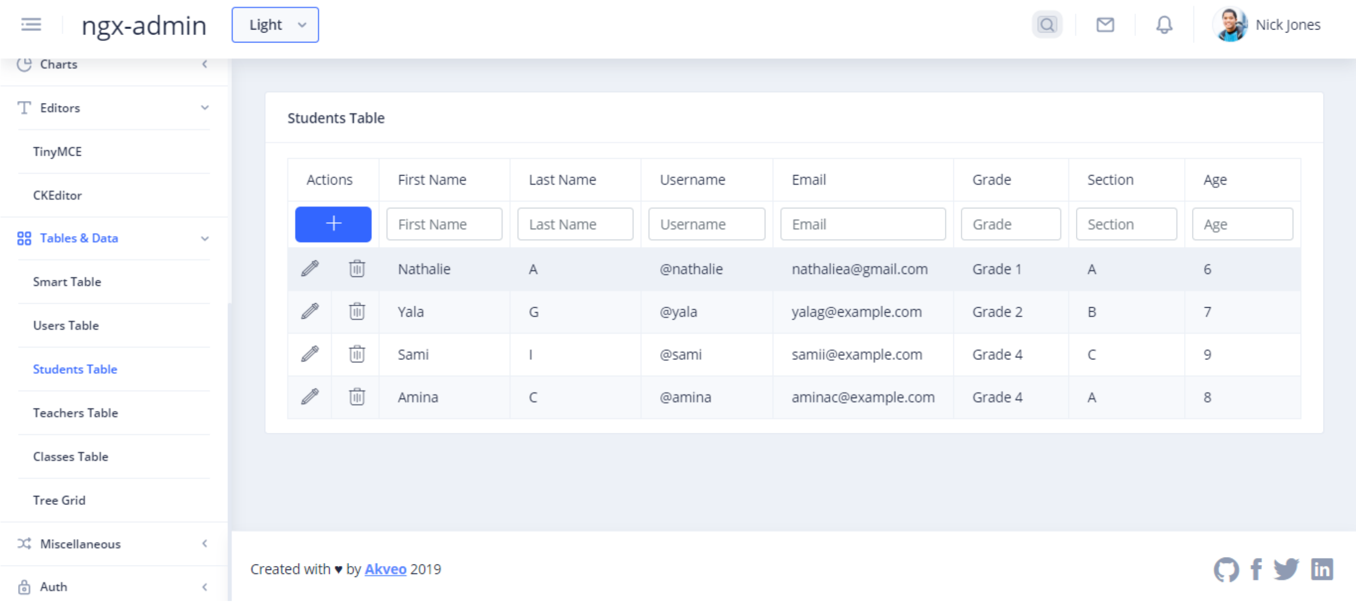
Figure 19: MongoDB Data Document

It’s important to note that MongoDB auto generate an id to each document.

# 2. Applications Views

## 2.1. Admin Application

### 2.1.1. Students Information Module



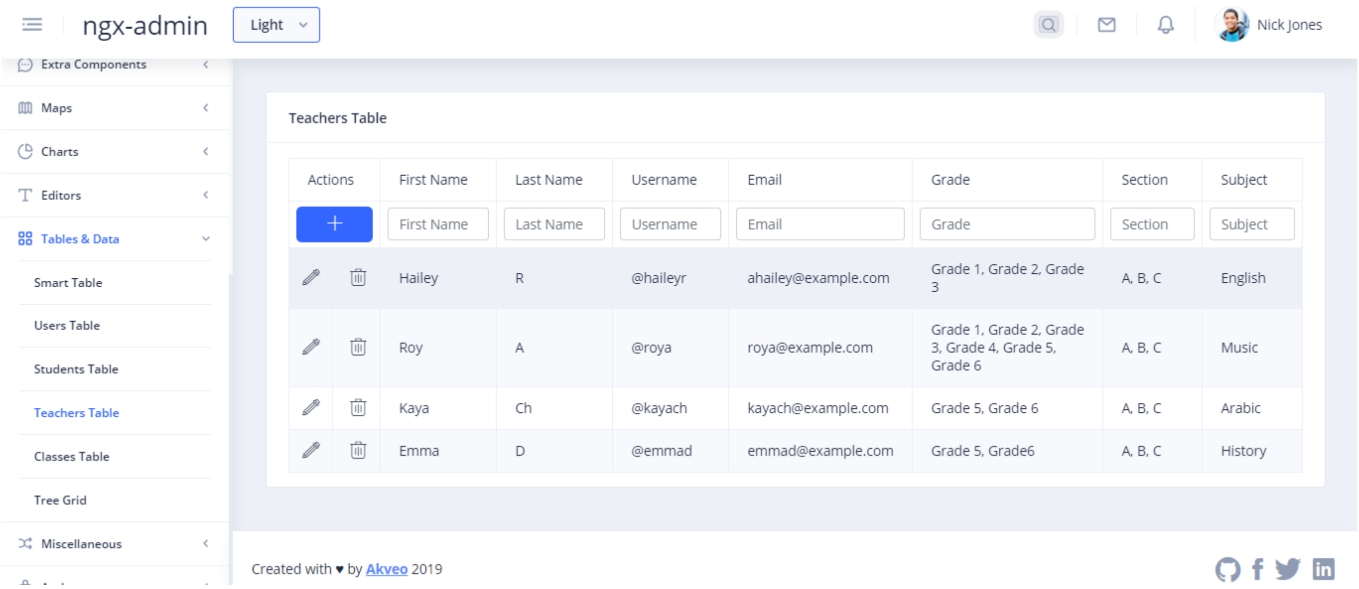
Students Table

| Actions | First Name | Last Name | Username | Email | Grade | Section | Age |
|---|---|---|---|---|---|---|---|
| + | First Name | Last Name | Username | Email | Grade | Section | Age |
|  | Nathalie | A | @nathalie | nathaliea@gmail.com | Grade 1 | A | 6 |
|  | Yala | G | @yala | yalag@example.com | Grade 2 | B | 7 |
|  | Sami | I | @sami | samii@example.com | Grade 4 | C | 9 |
|  | Amina | C | @amina | aminac@example.com | Grade 4 | A | 8 |

**Figure 20: Students Table**

In the Students Table page the admin can view all the current students registered in the system, search a student, add a new student, update a student’s information and delete a student from the system.

### 2.1.2. Teachers Information Module



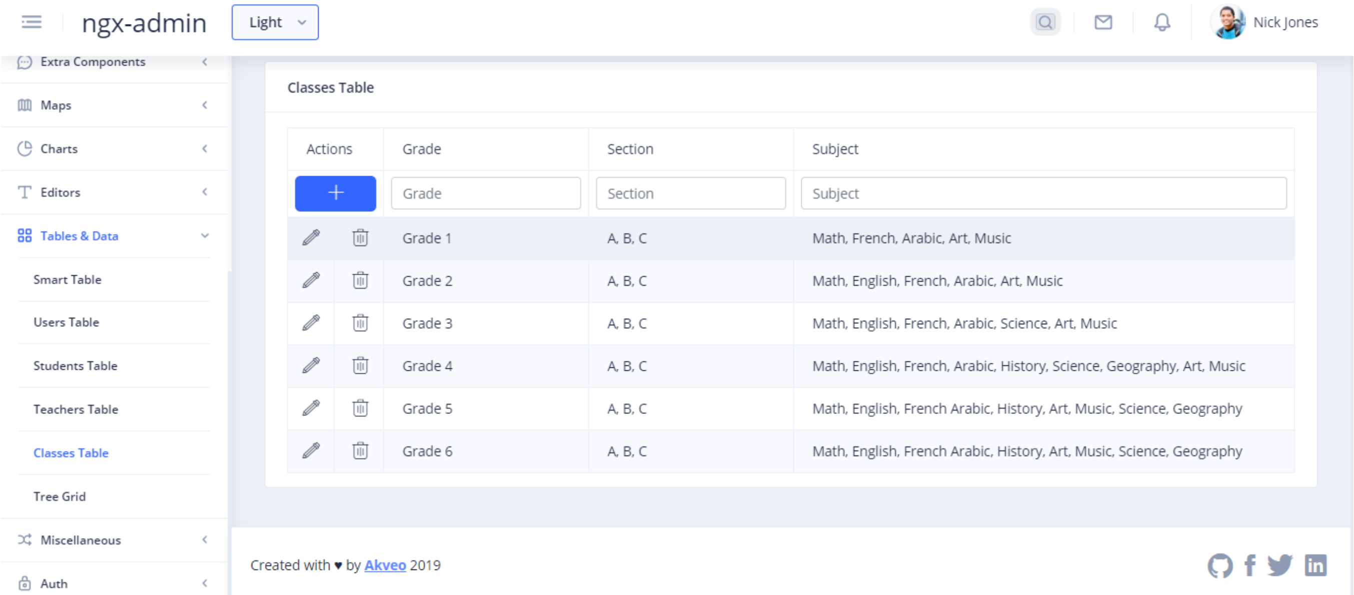
Teachers Table

| Actions | First Name | Last Name | Username | Email | Grade | Section | Subject |
|---|---|---|---|---|---|---|---|
| + | First Name | Last Name | Username | Email | Grade | Section | Subject |
| ✎ 🗑 | Hailey | R | @haileyr | ahailey@example.com | Grade 1, Grade 2, Grade 3 | A, B, C | English |
| ✎ 🗑 | Roy | A | @roya | roya@example.com | Grade 1, Grade 2, Grade 3, Grade 4, Grade 5, Grade 6 | A, B, C | Music |
| ✎ 🗑 | Kaya | Ch | @kayach | kayach@example.com | Grade 5, Grade 6 | A, B, C | Arabic |
| ✎ 🗑 | Emma | D | @emmad | emmad@example.com | Grade 5, Grade6 | A, B, C | History |

**Figure 21: Teachers Table**

In the Teachers Table page the admin can view all the current teachers registered in the system, search a teacher, add a new teacher, update a teacher's information and delete a student from the system.

### 2.1.3. Classes Information Module



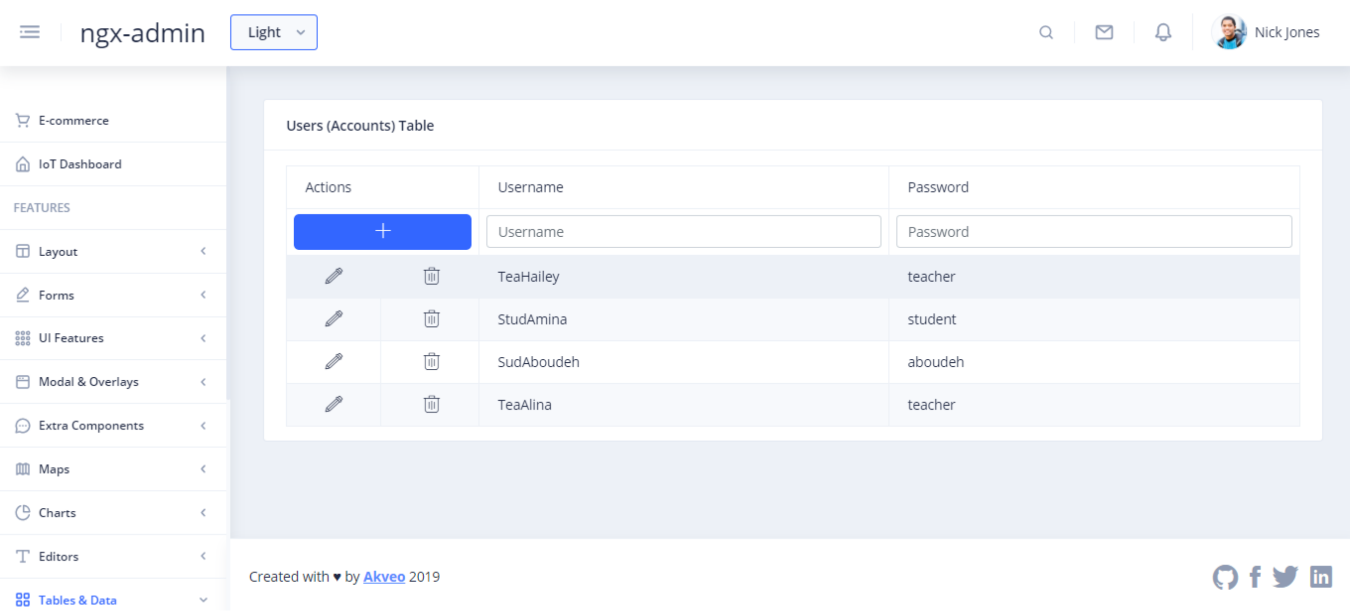
Classes Table

| Actions | Grade | Section | Subject |
|---|---|---|---|
| + | Grade | Section | Subject |
| ✎ 🗑 | Grade 1 | A, B, C | Math, French, Arabic, Art, Music |
| ✎ 🗑 | Grade 2 | A, B, C | Math, English, French, Arabic, Art, Music |
| ✎ 🗑 | Grade 3 | A, B, C | Math, English, French, Arabic, Science, Art, Music |
| ✎ 🗑 | Grade 4 | A, B, C | Math, English, French, Arabic, History, Science, Geography, Art, Music |
| ✎ 🗑 | Grade 5 | A, B, C | Math, English, French Arabic, History, Art, Music, Science, Geography |
| ✎ 🗑 | Grade 6 | A, B, C | Math, English, French Arabic, History, Art, Music, Science, Geography |

**Figure 22: Classes Table**

In the Classes Table page the admin can view all the current classes registered in the system, search a class, add a new class, update a class' information and delete a class from the system.

### 2.1.4. Users (accounts) Information Module



Users (Accounts) Table

| Actions | Username | Password |
|---|---|---|
| + | Username | Password |
| | TeaHailey | teacher |
| | StudAmina | student |
| | SudAboudeh | aboudeh |
| | TeaAlina | teacher |

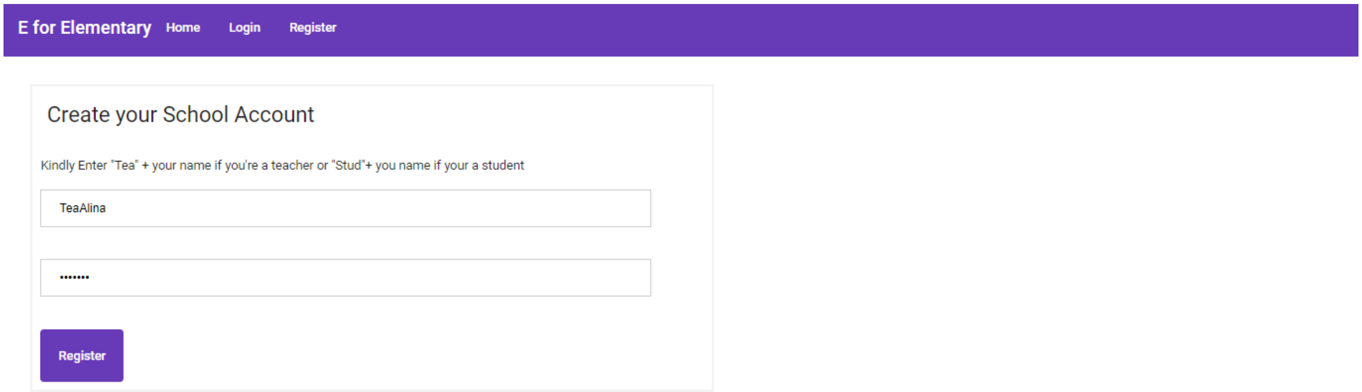
Created with ♥ by Akveo 2019

**Figure 23: Users Table**

In the Users (Accounts) Table page the admin can view all the current users acconts registered in the system, search an account, add a new account, update an account’s information and delete a account from the system.

## 2.2. Teacher and Student Application

### 2.2.1. Registration Module



Create your School Account

Kindly Enter "Tea" + your name if you're a teacher or "Stud"+ you name if your a student

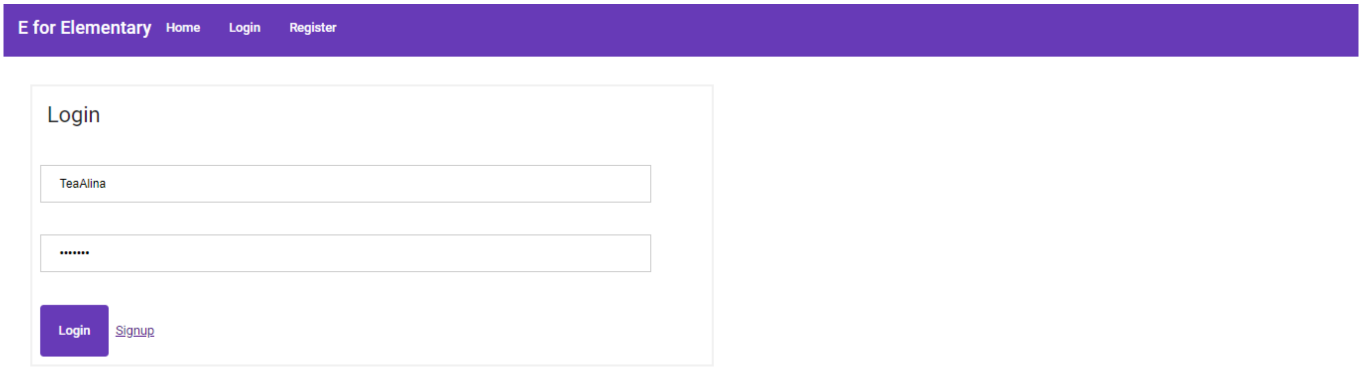
TeaAlina

•••••••

**Figure 24: Signup Form**

In the Create your school Account form in the Register page school’s teachers and students can register to the system with a teacher account as a teacher (TeaAlina) or a student account as student (SudAboudeh).

### 2.2.2. Login Module



Login

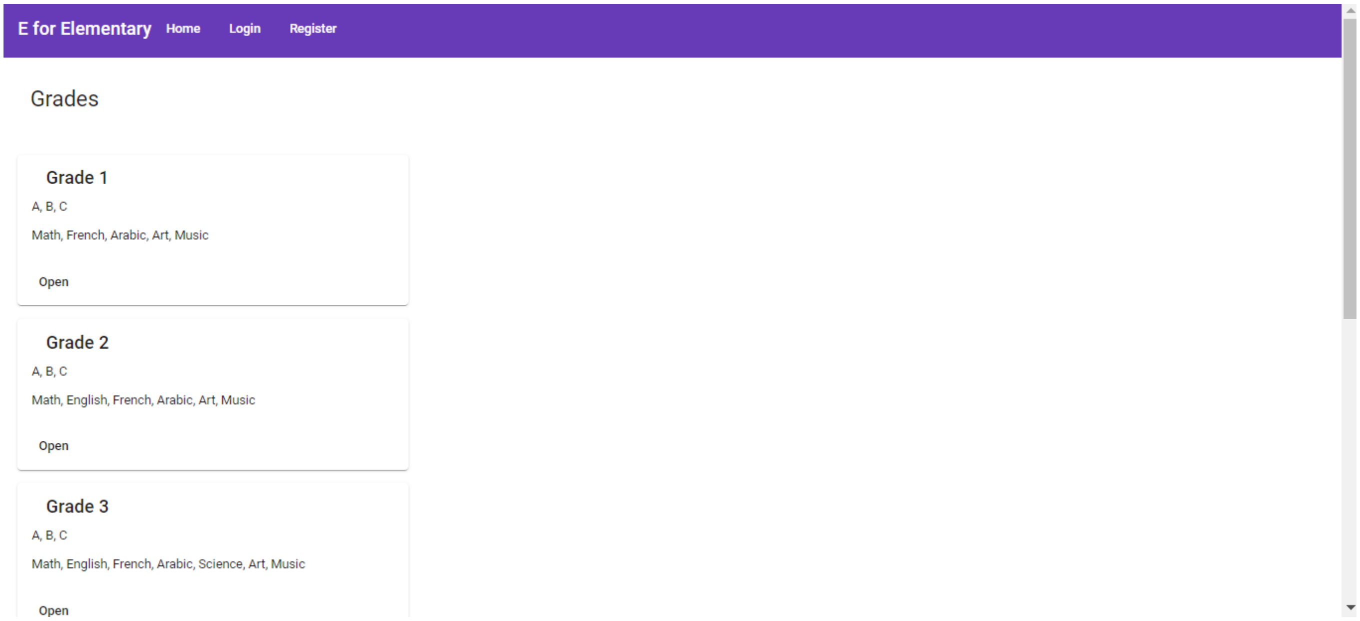
TeaAlina

•••••••

**Figure 25: Login Form**

In the Login form in the Register page school’s teachers and students can login to the system with their registered account and navigate it.

### 2.2.3. Grades Module



Grades

Grade 1
A, B, C
Math, French, Arabic, Art, Music
Open

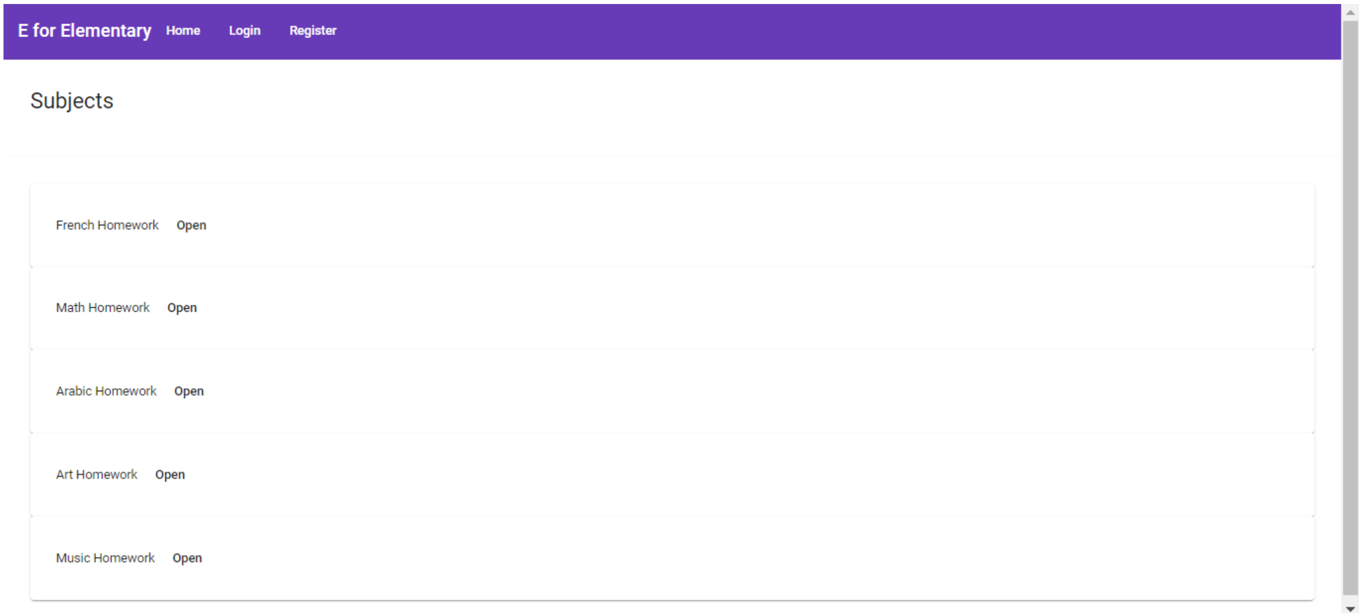
Grade 2
A, B, C
Math, English, French, Arabic, Art, Music
Open

Grade 3
A, B, C
Math, English, French, Arabic, Science, Art, Music
Open

Figure 26: Home Page

In the Grades pager (which is the Home page of the application) teachers and students can view the different grades of the school. Each grade is displayed with info about its sections and subjects. Teachers and students can find their class and open it.

### 2.2.4. Subjects Module



Subjects

French Homework Open

Math Homework Open

Arabic Homework Open

Art Homework Open

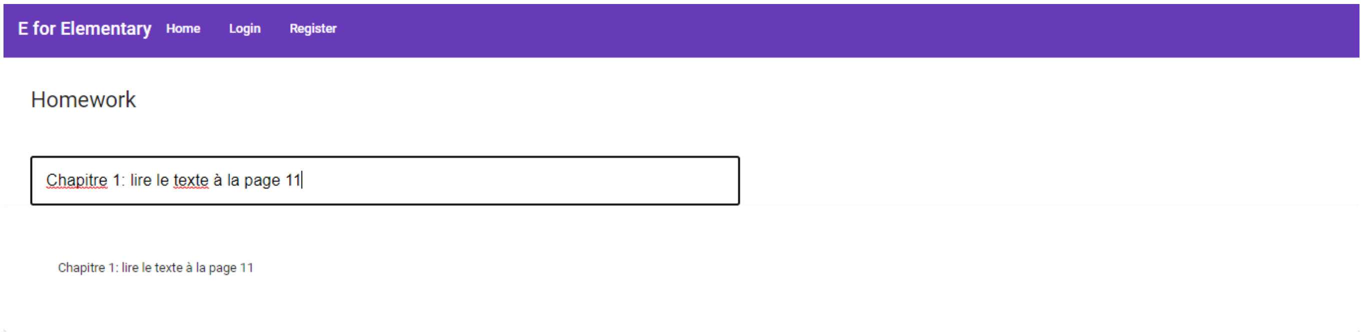
Music Homework Open

Figure 27: Subjects Page

In the Subjects page teachers and students can view the different subjects of each grade of the school. Teachers and students can find these subjects and open th to edit some homework (as ateacher) or view the updated homework (as a student).

### 2.2.5. Homework module

The view for the teachers:



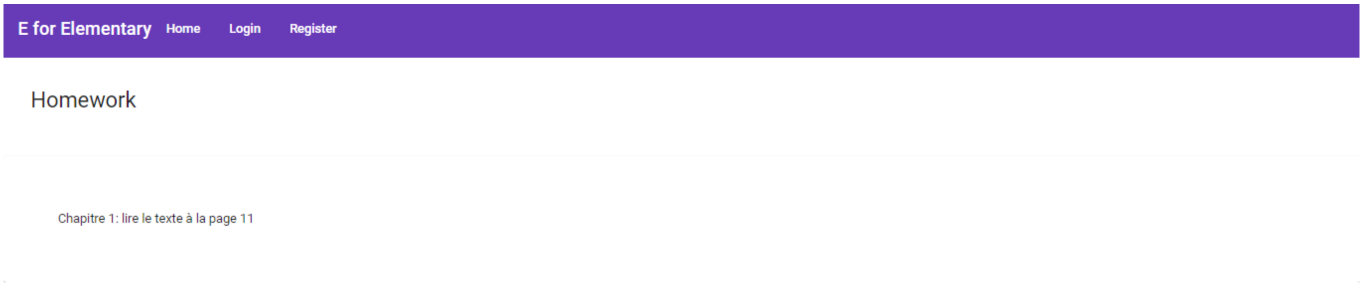
Homework

Chapitre 1: lire le texte à la page 11

Chapitre 1: lire le texte à la page 11

Figure 28: Homework Page fior Teachers

The view for the students:



Homework

Chapitre 1: lire le texte à la page 11

**Figure 29: Homework Page for Students**

In the Homework page teachers and students can view the different homework assignments of each grade subject. Teachers are able to update these homework assignment while students can only view the updates.

# Conclusion

We've presented in this report my final year project, an online school system called "E-Elementary" that cadres to elementary schools and is composed of two web applications, the admin application that helps schools' administrations store and manage their schools information and a teacher and student application that is used by teachers and students to access their classes and assignments of each class.

The report was composed of 4 chapters that explained all aspects of the system:

- **Chapter 1:** an overview about existing solutions and systems in the market was presented in this chapter, as well as project in scope / out scope, adopted development approach and methods, and project development phases / planning and implementation plan.

- **Chapter 2**: this chapter encompasses a deep presentation about the system's requirements, system main features, system actors, system use cases and an introduction about system design (using UML design Methods).

- **Chapter 3:** a full presentation about system design was included in this chapter, a detailed content explanation was added in order to explain database elements and its structure. This includes an explanation about NoSQL database, its advantages, why it was used in the system and how it was used. In addition to an introduction to MongoDB.

- **Chapter 4:** this chapter was decomposed into two parts. The first part, included an introduction about technical part of the project that includes the programming languages, frameworks, libraries and platforms used to build the two applications. The second part, was dedicated to present all system views that make up the two application.

# Perspective

The “E-Elementary” project was inspired by Google Classroom a leading online school information management system. And it included the development of the two web applications with the admin application being more sophisticated than the teacher and student application. The two applications can improve and reach their full potentials in many aspects:

Regarding the functions and features of the admin application:

- The application dashboard should include charts that showcase the data dependencies.

Regarding the functions and features of the teacher and student application:

- Teachers should be able to add many homework assignments for each subject.
- Students should be able to view the class they are registered in and not the whole list of classes.
- Students should be able to view the classes they teach in and not the whole list of classes.
- Teachers should be able to upload their courses (pdf, PowerPoint, videos..)
- Students should be able to comment on their homework assignments to signal that they have finished the homework or to ask questions.

To expend on the typical online school information management system’s features and functions:

- The application can include a video conferencing module to host online classes.
- The application can include a chat module.
- The application should cadre more to kids by offering them a playground like experience through implementing some collaborative games.

# References

Google Classroom: https://classroom.google.com/u/0/

UNESCO: https://en.unesco.org/covid19/educationresponse

MongoDB Documentation: https://docs.mongodb.com/

Angular Documentation: https://angular.io/docs

Angular Material Documentation: https://material.angular.io/

Nebular Documentation https://akveo.github.io/nebular/

Npm Documentation: https://docs.npmjs.com/

Postman Documentation: https://learning.postman.com/docs/getting-started/introduction/

Stack Overflow: https://stackoverflow.com/

Airbake.io: https://airbrake.io/blog/sdlc/iterative-model