Developer Manual

Pool

T1-havuz

Abdurrahman Bilal Kar 22003569

Ahmet Memduh Tutuş 22003153

Berkay Ayçiçek 22003111

Fırat Utku Gül 22003105

Arda Kırcı 22002031

12.05.2024

A high-level description. What does the system do, and why would a user want to use it?

Pool is a comprehensive course management system designed to simplify and streamline course management processes within educational institutions. It offers a wide range of features aimed at students, teachers, and administrators to enhance efficiency, communication, and productivity.

Users of the Pool system can benefit from:

1. **Automated Administrative Tasks:** Pool automates routine administrative tasks such as exam scheduling, assignment submission, grading, and event scheduling, reducing the burden on administrators and allowing them to focus on more strategic initiatives.
2. **Enhanced Communication:** The system provides communication tools such as discussion forums, messaging, and event notifications, facilitating seamless communication between students, teachers, and administrators.
3. **Efficient Resource Sharing:** Pool offers collaboration tools and a centralized repository for sharing resources, allowing users to easily access and collaborate on course materials.
4. **Performance Tracking:** Users can track attendance, grades, and performance data, enabling teachers and administrators to monitor student progress and identify areas for improvement.
5. **Custom Reporting:** The system allows for the creation of custom reports and provides performance analytics and trend reporting, empowering administrators to make data-driven decisions.
6. **Versatility and Adaptability:** Pool supports multiple user types and offers features such as exam archives, instructor evaluations, and project team management, catering to the diverse needs of educational institutions.

How to install it?

Clone the repository, and you will be good to go. However, remember that you need to have Flutter, Node.js, and MongoDB installed on your system, along with the libraries they use. The libraries can be installed using the following guide:

Also, you need to import some JSON files to MongoDB to use the system.

How to run it?

Navigate to the Backend directory from the project's root directory. Go: “npm start” to start the backend. Then, navigate back to the root and the Frontend directory. In the frontend directory, go “flutter run -d Chrome,” and the frontend will start.

You can just really go to your browser and go to the address “http://92.205.57.115/login” and enter your credentials.

What are works in progress and known bugs?

Chat is not yet real-time. We are debugging the socket.io

Forum, which sometimes needs double-clicking.

Some views will be better

The forget password screen does not work yet

How to report a bug?

via Git Hub. Go to issues and create an issue with a title and a proper explanation.

How is the source code organized, what is the layout, and what is where?

The app adopts the MVC approach. The root directory has two main directories, Backend and Frontend. The backend directory has three main directories: routes, controllers, and models. These are self-explanatory. Additionally, there is a middlewares directory containing jwt, role checker, and file handler. Another additional directory is utils, which includes a web socket.

How to build it and use it?

Flutter build can be gotten by going “flutter build web” in the Frontend directory. The backend can be run wherever it is moved, and it is enough to go “npm run” in the backend directory.

Who is responsible for what?

Arda.

Bilal contributed significantly to the development of the chat, forum, event, notification, and ABET modules within the system. His expertise in JSON Web Tokens (JWT) authentication and file handling using Multer ensured secure user authentication and efficient file uploads. Bilal also played a crucial role in establishing the general MVC structure of the application, ensuring a cohesive and organized codebase.

Fırat's contributions focused on the section and admin page functionalities, emphasizing user authentication and role-based access control. He implemented features specific to instructors and students, ensuring each user type had access to relevant functionalities. Additionally, Fırat managed the file handling aspect and implemented assignment and course management features, enhancing the overall user experience.

Berkay's expertise in attendance tracking, grading systems, debugging, and data validation significantly improved the functionality and reliability of the system. He implemented robust validation mechanisms to ensure data integrity and developed comprehensive analysis tools to provide insights into student performance. Additionally, Berkay managed the questions module, including question weights and archive functionalities, contributing to the system's efficiency and effectiveness.

Memduh's contributions were instrumental in developing the chat, forum, and many front-end components of the system. He designed and implemented most front-end features, including homepages, app bars, and calendar functionalities, ensuring an

intuitive and visually appealing user interface. Memduh also contributed to the ABET module, enhancing the system's compliance with accreditation standards.