

Barmej

Team Members

- **Haider:**
 - Strengths: back-end, algorithms, front-end back-end connections
 - Weaknesses: GUI
- **Kanza:**
 - Strengths: algorithms, back-end structuring
 - Weaknesses: GUIs and front-end
- **Laila:**
 - Strengths: algorithms, extensibility
 - Weaknesses: front-end/GUIs
- **Moustafa:**
 - Strengths: databases, back-end connections, GUI
 - Weaknesses: security, graphical design, javascript animation

Overview

This platform is a teaching and learning platform that allows its users, (students, educators, and administrators) to access computer science coursework, make computer science lessons that are accessible to everyone and keep track of how the specific educator and student body is performing. This platform is targeted towards businesses and organizations along with schools that have employees/students whose native language is not English and is who have reading proficiency in either Arabic or Urdu. This includes businesses, organizations, and schools of the refugee community which belong to Syria and other parts of the Arab world. The student version of this is the highlight of the project, it includes an IDE, which will support one programming language like Java or Python and will be used to teach the students or the employees basic concepts of programming. Moreover, additional features like syntax highlighting in the IDE will allow students to identify their code mistakes and understand them properly in their own language.

This platform also allows educators a lot of freedom in designing the courses of their choice and running them as they see fit and allows them to be successful in doing so by introducing a

student tracker, a video platform which can be used to access online resources which can be provided to students and will help them in their learning. Finally, being one of its kind this platform also allows the administrators to access the progress of the educators in teaching courses and control student enrollment.

The functionality of each of the separate clients will be dealt with on a separate web page which will be hosted on the backend. All the pages will be dynamic and will include a Student/educator search for the administrator, a student/coursework search for the educators and coursework only search for students. All of the web pages will be in Arabic / Urdu and the website will support IDEs in these languages. Lastly, in the given time frame the team decided that it would be best to go with an IDE that allows the user to see error messages in their native language.

Features

Our platform has 3 key users: educators, students, and administrators. Each of the three users accesses a different set of features on our website as follows:

- Educator:
 - Lesson Building Tools
 - This allows educators to make and preview lessons, as well as save their progress in drafts. If time allows, this might be made into a drag and drop interface in order to allow for increased flexibility, although this feature is not an integral part of our project.
 - This feature also allows the teacher to build a test suite for student code.
 - Student Tracker
 - Allows the course creator to view the work of students taking their course and track their progress
- Student:
 - Integrated Development Environment
 - Includes syntax highlighting, error messaging
 - Compiler for one programming language
 - Progress tracking page
 - Keeps track of all the courses they are enrolled in as well as how much of the course they have completed
 - Lesson pages
 - These integrate videos, text, and the IDE/compiler
- Administrator:
 - Course Selection and Assignment
 - Includes the ability to search through courses by keyword
 - Lets administrators subscribe and unsubscribe from courses
 - Student Tracker + Manager

- Lets the administrator keep track of which/how many students are enrolled in certain courses
- Used for enrolling students in courses

All 3 users will have the following features, although the actual functionality might differ:

- Sign up/sign in
- Profile editing

Our platform will also support multiple languages; the ones we are building in are Arabic and Urdu.

Clients

This platform is intended for three groups of users: students, educators/teachers, and institution administrators. The following three categories explain in greater detail what the experience for each group of users will look like:

1. **STUDENTS:** Each student will start by creating their own student account. Students will be prevented from potentially signing up as an educator or administrator by not having access to educator/administrator codes specific to the particular teacher or institution, which the latter would need to create their accounts. Once a student account is created, their account can be linked to specific courses by their institution administrator. The student will then be able to see all the courses they are enrolled in on a potential 'dashboard' that would act as the homepage of their accounts. The student will also be able to navigate to individual courses and complete course tasks. In addition, they will be able to view their saved progress, and also interact with the web-IDE and text and video components of the courses. Students will not have the authority to enroll/register for courses. This will be handled by the administrators.
2. **EDUCATORS/TEACHERS:** Each educator will start by creating their educator account which they will be able to do after being given some sort of unique identifying ID. The educators will interact with the platform by having an easy-to-use interface which makes designing courses easy for teachers with varying levels of technical background. Educators will be able to include text, pictures or videos to their courses and also put up questions or assignments. Their task will mainly involve designing and creating courses, and they can also track the progress of the students enrolled in their courses.
3. **ADMINISTRATORS:** Yet again, each administrator will start by creating their administrator account for their particular institution, a process that will be overlooked by another form of unique identification. Administrator access of the platform will involve an

ability to view the different courses available, the ability to add courses to the list of courses used by that administration, and also assigning of different students in the institution to certain courses. Administrators will not have the authority to create new courses or edit existing ones.

Database

One of the major components of our website is the database. We would need to have a database of all the users, educators, and admins that have accounts on our website. Moreover, we would need to save the data for each course in our database. Each course would have lessons, videos, assignments, etc, all of which would be stored somewhere on the server-side and access points to these files shall be stored in our database. Queries that we will need to implement include getting all classes at any institution, getting all students in a class, getting all courses for any educator, getting all classes for a student, getting all course details for a student, etc. Therefore, we will need the following schemes in our database.

1. user

```
`id`: integer PRIMARY KEY
`username`: text UNIQUE
`first_name`: text NOT NULL
`middle_name`: text
`last_name`: text NOT NULL
`email_address`: text NOT NULL
`password_hash`: text NOT NULL
`user_type`: integer NOT NULL
               : : where 1 is a learner, 2 is an admin, and 3 is an educator
`photo_url`: text NOT NULL
`institution_id`: integer FOREIGN KEY institution(id)
`payment_information`: text
```

2. class

```
`course_id`: integer FOREIGN KEY course(id)
`institution_id`: integer FOREIGN KEY institution(id)
```

3. course

```
`id`: integer PRIMARY KEY
`educator_id`: integer FOREIGN KEY user(id)
`num_pages`: integer NOT NULL >= 0
`title`: text NOT NULL
`coding_language`: text
`level`: text
```

`'thumbnail_url': text`

4. course_page

`'id': integer PRIMARY KEY`

`'course_id': integer FOREIGN KEY course(id)`

`'page_url': text`

5. enrolled_courses

`'student_id': integer FOREIGN KEY user(id)`

`'course_id': integer FOREIGN KEY course(id)`

6. institution

`'id': integer PRIMARY KEY`

`'name': text NOT NULL`

`'registration_number': text NOT NULL`

`'address': text`

`'type': text`