TED UNIVERSITY

CMPE 492 Senior Project 2

A Gamified Training Platform Supported By AI

Final Report

The URL of the project web page:
https://isthisecho.github.io/TermProject/

Yasemin Direk 14506076728
Furkan Erkan 25081572932
Ekrem Görgüç 13472000546

Supervisor: Ulaş Güleç
Jury Members: Haydar Çukurtepe, Emin Kuğu

1. Introduction 3
2. Machine Learning Process 4
2.1. Data Generation 4
2.2. Data Clustering and Labeling 5
2.3. Algorithm Selection and Classification 11
2.4. Results 13
3. Tools and Technologies 15
3.1. Scikit Learn 15
3.2. Yellowbrick: Machine Learning Visualization 16
3.3. Seaborn: Statistical Data Visualization 16
3.4. Matplotlib 16
3.5. Joblib 16
3.6. NumPy 16
3.7. Pandas 16
3.8. Statsmodels API 16
3.9. ML.NET 16
3.10. Model Builder 17
3.12. Font Awesome 17
3.13. Firebase Tools 17
3.14. React Navigation 17
3.15. Bootstrap 18
3.16. React 18
3.17. React PLayer 18
4. User Manual 18
4.1. User Manual For Web Application 18
4.2. User Manual For Mobile Application 24
5. Impact of Engineering Solutions 37
5.1. Economic Impacts 37
5.2. Environmental Impacts 37
5.3. Social Impacts 37
6. References 38

1. Introduction

Nowadays, online education platforms are becoming widespread and used in many areas. Although online education platforms are accessible and widespread, they can cause loss of motivation and productivity due to the decrease in student-teacher interaction. The aim of our project is to make this online education platform more efficient and fun by creating a competitive environment.

The purpose of this project is to provide an education platform supported by artificial intelligence. We aim to provide a system that will recommend to students about the subjects they are failing. Recommendations will be made according to the subjects that students make wrong in the exams.

In addition to the features provided by a standard online education platform, we aim to increase the motivation of the students and their interaction with the course with features such as a leaderboard system, badge system, missing topic suggestion system.

Our project consists of two parts, a web application running on the web side and a mobile application. The web application will be used by the instructors and the mobile application will be used by the students.

On the mobile part of the project, React Native has been used to develop mobile applications on either Android OS and iOS. The idea behind React Native is to write all of the codes with JSX which has embeddable XML-like syntax and React Native CLI will convert them to native codes for Android OS or iOS. As a result, the main advantage is to have a consistent build on different operating systems at the same time.

On the mobile application, After Login process participants would navigate between ‘Home’, ‘Courses’ and ‘Events’ screens. At the end of the day, Participants can view course materials that are already assigned by the Instructor under the specified weeks. Also, They can take the quiz and see their standings with respect to their quiz scores. Finally, Participants would logout from the application if they desire.

On the web part of the project, React has been used to develop web applications on the web apps. As we said, we have created our mobile application with React native and we wanted to continue with the similar ,flexible and easy to use programming language such as React.

To briefly explain what a web application is, We will sum up the pages and functionality. Later on we'll be showing them in the user manual section in this report.

After logging into the system with an approved email and password, We can see the Dashboard screen which includes courses and quizzes. On the left side of the website we can always click to the sidebar for navigating easily between pages. In the Courses page, we can add courses and inside these courses we can create new chapters. These chapters can include pdf's or video's. Also in the quiz screen we can create a quiz. Inside of the quiz we can create new questions. These questions will be multiple choice and we can upload some images to elaborate questions more specifically. Also in this system we can see the verified users in the user's screen. Lastly we have a my profile page which includes settings about changing password or email for specified users.

2. Machine Learning Process

2.1. Data Generation

A data generating process in statistics and empirical sciences is a real-world process that generates the data of interest. Since the system is not used by real users and no real data can be obtained, a data generation phase is required in order to select the appropriate algorithm for the scenario.

Our aim is to divide the students into 3 groups according to their success, and to recommend study materials for the students in the unsuccessful category on the subjects they failed. In order to create the data to be used to achieve this purpose, we considered the data to be obtained from an exam result. When a student takes the quiz, there will be a time period for each question and a chapter to which each question belongs. Since recommendations will be

made on the basis of the chapter, the data we need consists of the correct number of answers the students made on any chapter and the time they spent doing these questions. In other words, for a chapter, we need to create a dataset consisting of the correct number of answers and the answer time. Assuming that there are 300 questions for each chapter, and that each question has a duration of 60 seconds, we created a data which consists of a total of 300 questions and 300 minutes. We implemented a function using Python to create an equal number of 3 clusters of this data with a smooth distribution. In this function, we generated the number of correct answers and the answer time, as well as calculated the average time spent per question, at the intervals we determined, and saved them in a CSV file. We created a data set suitable for our scenario by looking at the data distributions in the graph.

2.2. Data Clustering and Labeling

After the data generation stage, we used the K-Means Clustering Algorithm to separate this data we created according to its clusters and label it later.

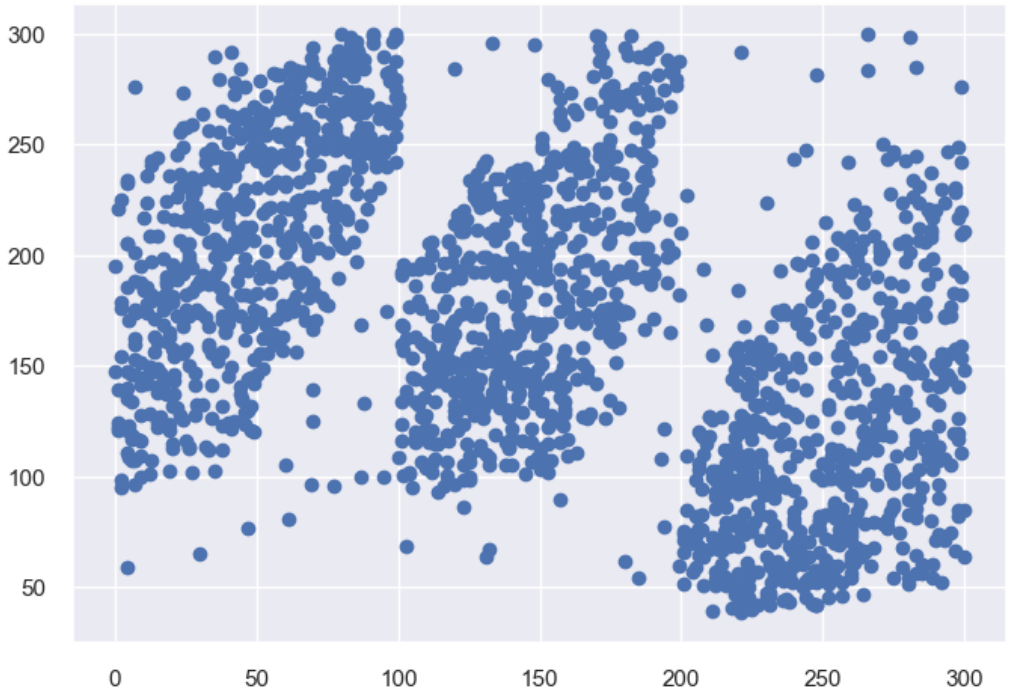


Figure 1: Raw Data

The most often used unsupervised learning algorithm is K-Means clustering. It is employed when we have unlabeled data, or data that does not have defined categories or groups. The algorithm uses an easy or basic method to classify a given data set into a predetermined number of clusters. The K-Means algorithm assigns each data point to one of K groups iteratively based on the features provided. Feature similarity is used to group data points. To arrive at a final result, the K-Means clustering algorithm employs an iterative process. As input, the algorithm requires the number of clusters K and the data set. Each data point in the data set has a collection of features. The algorithm starts with initial estimates for the K centroids. After that, the algorithm iterates between these steps:

- The data is divided into K categories, with k being a fixed number.
- At random, K sites are chosen as cluster centers.
- The Euclidean distance function is used to assign objects to their closest cluster center.
- The centroid or mean of all items in each cluster is calculated.
- Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.

We implemented the K-Means Clustering algorithm using Python. Python's libraries and modules such as Sklearn, Yellowbrick, Seaborn, Matplotlib, Joblib, NumPy, Pandas and Statsmodel API are used in the implementation of the algorithm, determining the optimal number of clusters and visualizing the data.

To find the number of clusters in the data, we executed the K-Means clustering algorithm for different values of K, and compared the results. The performance of the K-Means algorithm is determined by the value of K. Therefore, we chose the optimal K number, which is 3, which gives the best results. There are several different methods for determining the ideal value of K. We used the two most common techniques, the elbow method and the Silhouette method.

Elbow Method

In K-means clustering, the Elbow Method is used to determine the optimal number of clusters. In the Elbow method, the number of clusters (K) is varied between 1 and 10. WCSS (Within-Cluster Sum of Square) is calculated for each value of K. In a cluster, WCSS is the sum of squared distances between each point and the centroid. The plot appears like an elbow

when we plot the WCSS with the K value. The WCSS value will drop as the number of clusters grows. When we examine the graph, we can see that the graph will change rapidly at one point, forming an elbow shape. The graph begins to move nearly parallel to the X-axis at this point. The optimal K value or number of clusters corresponds to this point.

Figure 2: shows how the elbow method works for our dataset.

Silhouette Method

In terms of how well samples are clustered with other samples that are similar to each other, the Silhouette score is used to evaluate the quality of clusters generated using clustering algorithms like K-Means. The silhouette plot shows how close each point in one cluster is to points in neighboring clusters, and thus allows to visually examine parameters like cluster count. The range of this metric is [-1, 1]. Silhouette coefficients approaching +1 indicate that the sample is separated from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters, while negative values indicate that the samples were incorrectly assigned to the wrong cluster.

The following is the formula for determining the silhouette coefficient for a given data point:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

S(i): The silhouette coefficient of the data point i.

a(i): The average distance between i and all the other data points in the cluster to which i belongs.

b(i): The average distance between i and all clusters to which i does not belong.

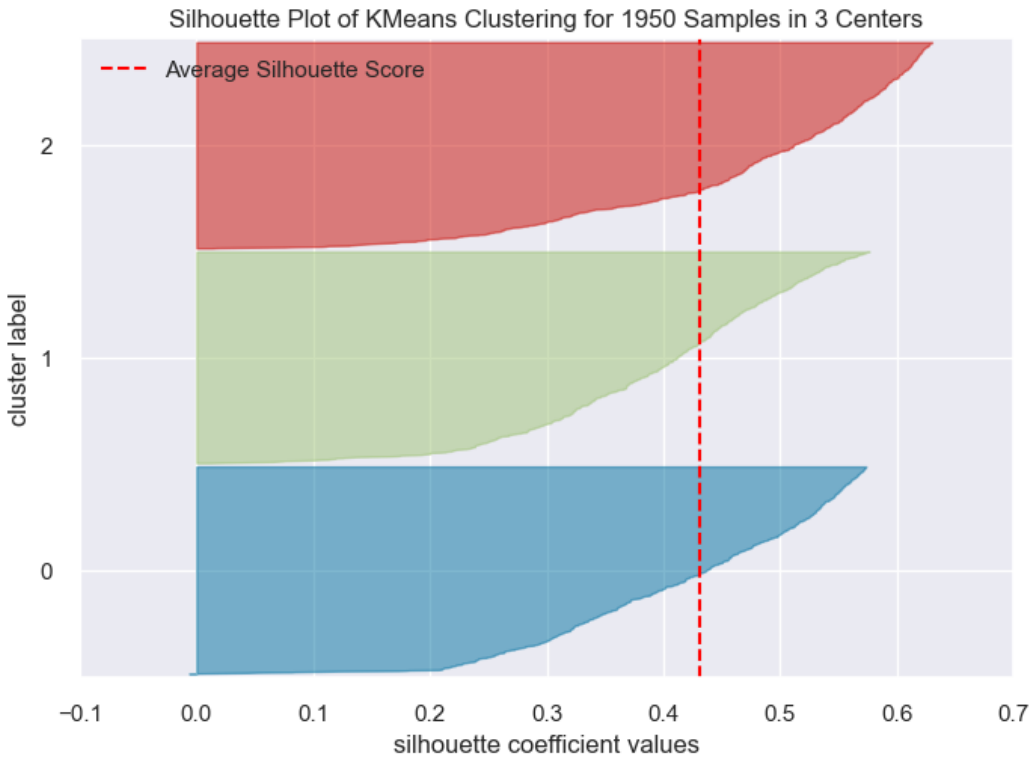


Figure 3: shows how the Silhouette method works for our dataset with k = 3.

Figure 4: shows how the Silhouette method works for our dataset with k = 4.

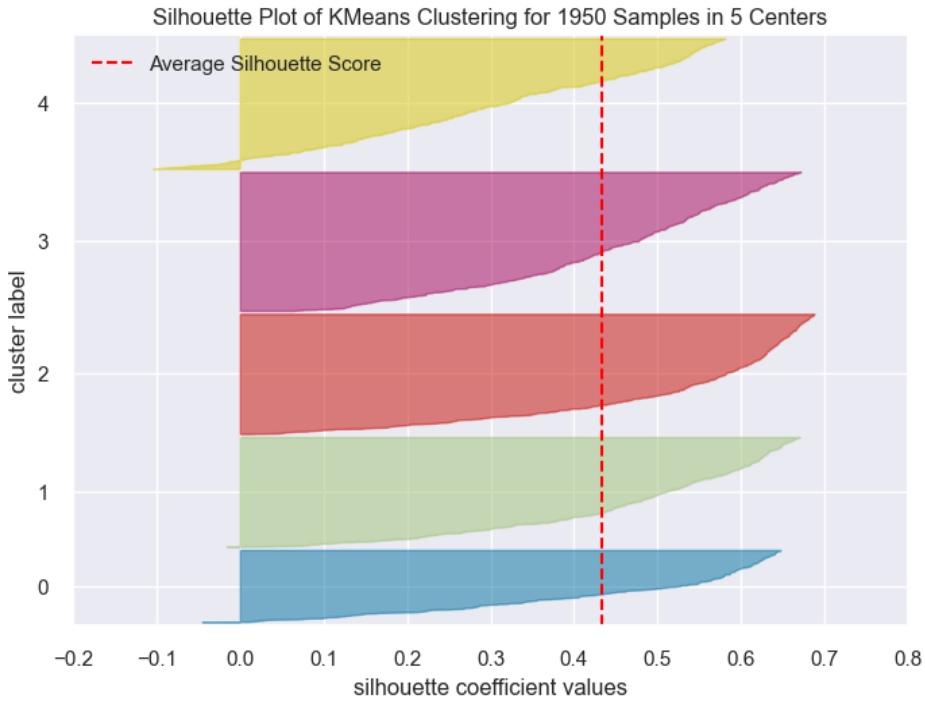
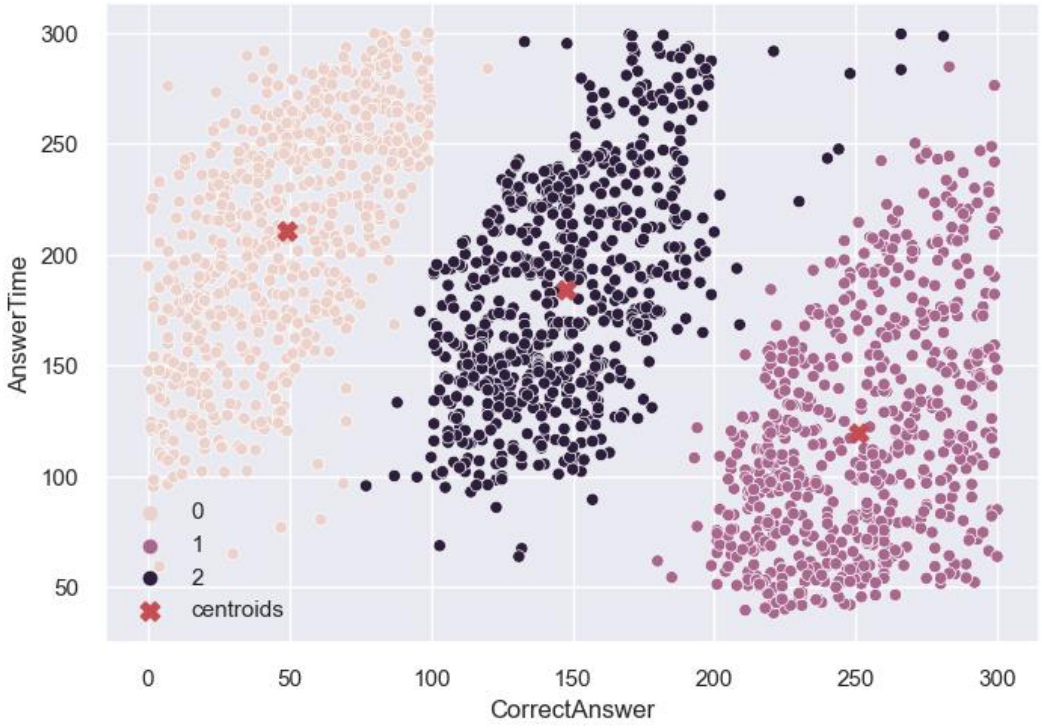


Figure 5: shows how the Silhouette method works for our dataset with k = 5.

We see that the silhouette score is higher when the number of clusters is 3 and 5. According to the silhouette method, we need to choose between 3 or 5 for the k value. By combining the results of the Silhouette and Elbow methods, we see that the optimal number of clusters for our data is 3.

After implementing the K-Means Clustering algorithm with k = 3, we obtained the following clusters:



After dividing our data into clusters, we renamed the clusters as Successful, Average and Unsuccessful, considering the number of correct answers and the answer times.

Finally, we saved the data that we labeled with K-Means Clustering to be used in the classification phase.

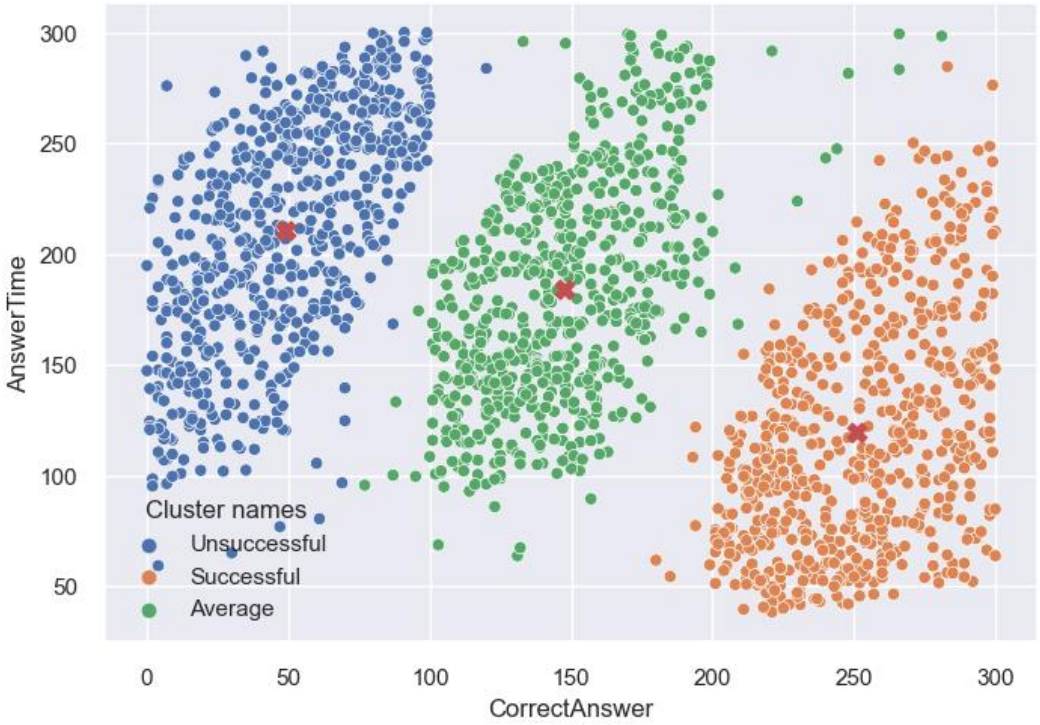


Figure 6: Clustered Data

2.3. Algorithm Selection and Classification

At the classification step, we can use the data we labeled by using K-Means Clustering algorithm. Since our data consists of 3 classes, we searched for algorithms suitable for the multiclass classification scenario. As we will do the Classification using ML.NET, we have listed the most suitable algorithms for our data using Model Builder.

One-Versus-All

This is an approach that builds multiple binary classifiers by doing a binary classification algorithm for each class in the dataset. After that, predictions are made by running these binary classifiers and selecting the one with the highest confidence score.

This algorithm works with all of ML.NET's binary classifiers. For multi-class scenarios, a few binary classifiers already have implementations, so users can pick between the two based on the scenario.

Pairwise Coupling

In this technique, a binary classification algorithm is trained on each pair of classes. The pairings are not in any particular order, but they are made with replacement: so, if there were three classes, 0, 1, 2, we would train classifiers for the pairs (0,0), (0,1), (0,2), (1,1), (1,2), and (1,1). (2,2). An input data point is regarded as a positive example for each binary classifier if it belongs to one of the two classes in the pair, and a negative example otherwise. The probabilities for each pair of classes are calculated at prediction time as the probability of being in either class of the pair given the data, and the final prediction probabilities out of that per class are calculated given the probability that an example is in any given pair.

SGD Calibrated

In order to implement the One-Versus-All and Pairwise Coupling algorithms, we used the SGD Calibrated binary classification algorithm.

The Stochastic Gradient Descent (SGD) is a prominent stochastic optimization process that can be used to achieve state-of-the-art performance in a variety of machine learning tasks. This trainer uses the Stochastic Gradient Descent algorithm for binary classification, which allows for multi-threading without any locking. Stochastic Gradient Descent produces a nearly optimal rate of convergence when the associated optimization issue is sparse.

Naive Bayes

The probabilistic classifier Naive Bayes can be used to solve multiclass problems. Based on the sample count for each feature combination group, the conditional probability for a sample belonging to a class can be computed using Bayes' theorem. However, the Naive Bayes Classifier is only practical if the number of features and the possible values for each feature are relatively small. Even if features of a class may be dependent on one another, it assumes independence among their presence. This multi-class trainer takes "binary" float feature values: feature values larger than zero are considered true, whereas feature values less than or equal to 0 are considered false.

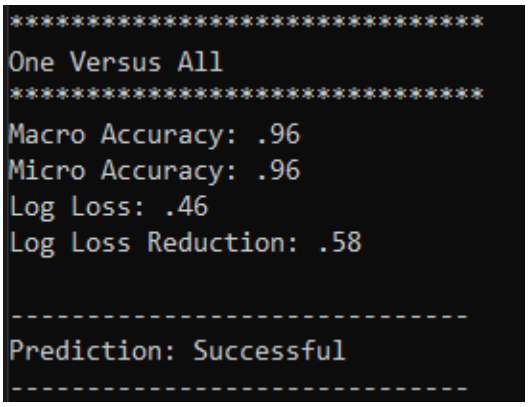
When we look at the use cases of ML.NET multiclass classification algorithms, we see that One-Versus-All and Pairwise Coupling algorithms are suitable for our scenario. However, we see that the Naive Bayes algorithm is not suitable for our dataset and feature values. In order to see if we chose an algorithm suitable for our scenario, we applied all 3 algorithms and compared our results.

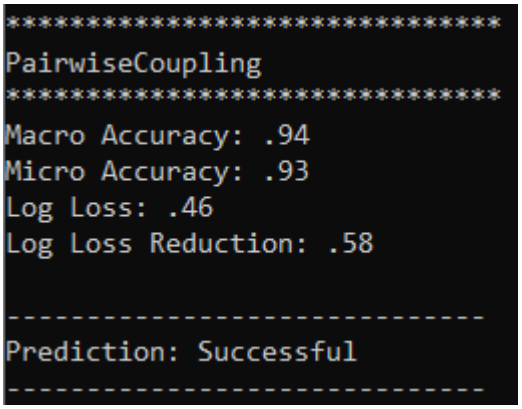
2.4. Results

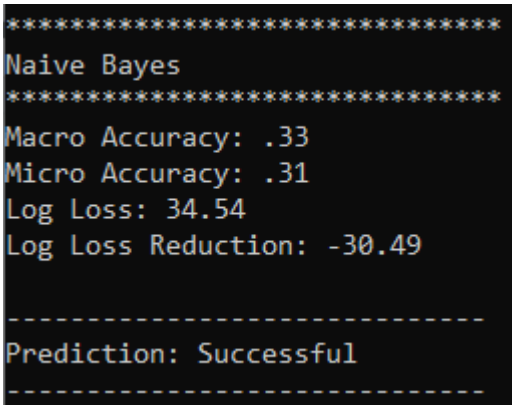
Results for a student data sample in the Successful category

Number of correct answers: 250

Total answer time: 150



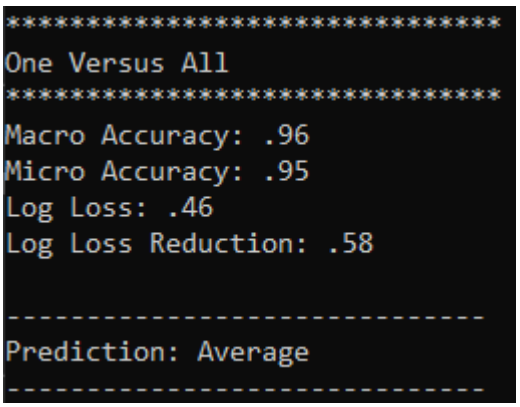




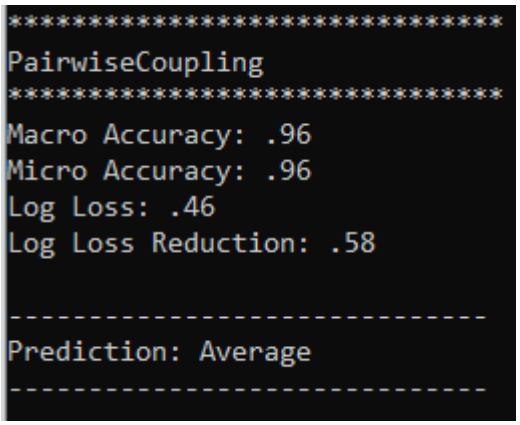
Results for a student data sample in the Average category

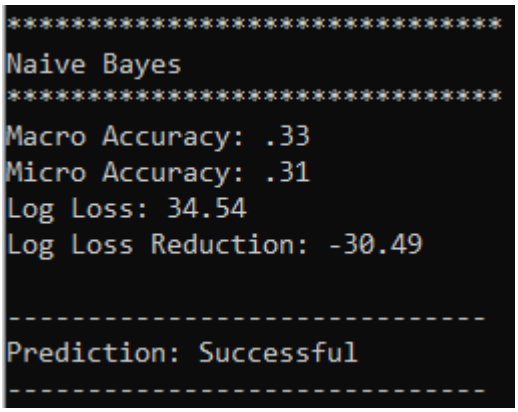
Number of correct answers: 170

Total answer time: 210



`

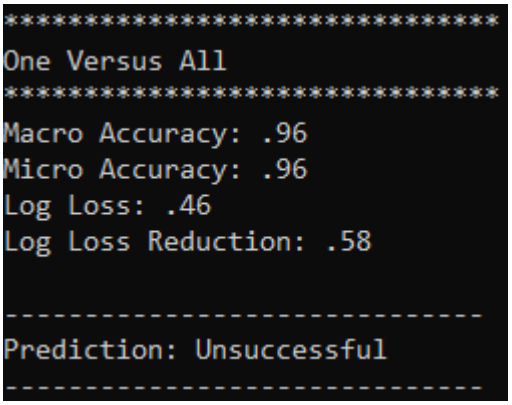


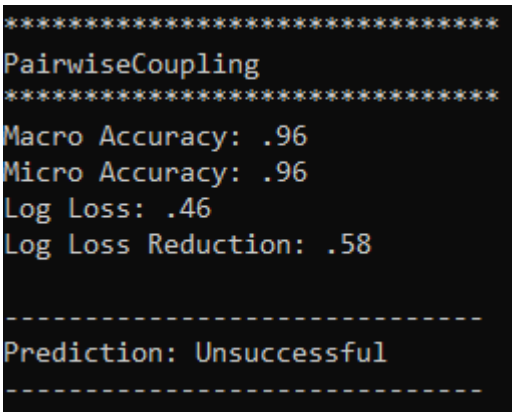


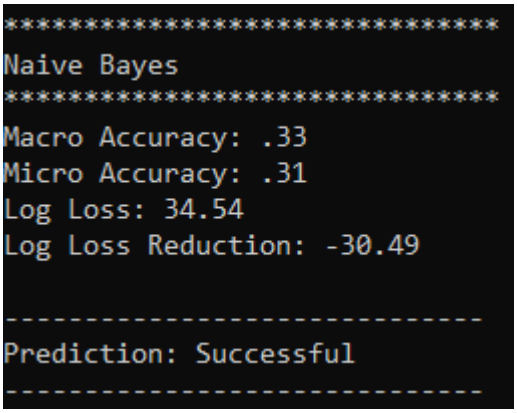
Results for a student data sample in the Unsuccessful category

Number of correct answers: 60

Total answer time: 140







Micro Accuracy: Micro-average Accuracy calculates the average metric by combining the contributions of all classes. It is the percentage of instances correctly predicted. The micro-average makes no distinction between classes. Essentially, each sample-class pair contributes the same amount to the accuracy metric. The closer to 1.00, the better.

Macro Accuracy: Macro-average Accuracy is the average accuracy at the class level. The macro-accuracy is the average of the accuracy for each class. Every class, in essence, contributes equally to the accuracy metric. Minority groups are accorded the same weight as bigger groups. The macro-average measure gives each class the same weight regardless of how many instances of that class are in the dataset. The closer to 1.00, the better.

Log loss: The performance of a classification model with a prediction input of a probability value between 0.00 and 1.00 is measured using logarithmic loss. As the expected likelihood differs from the actual label, log-loss grows. The closer to 0.00, the better.

Log Loss Reduction: The advantage of the classifier over a random prediction is referred to as logarithmic loss reduction. Ranges from -inf and 1.00, where 1.00 is perfect predictions and 0.00 indicates mean predictions.

When we compare the results of all three models, we find that Naive Bayes has the lowest accuracy and in some cases misclassifies the sample data.

We see that One-vs-All and Pairwise Coupling algorithms classify the samples correctly and their accuracy is higher than Naive Bayes.

We use the One-vs-All algorithm in our final model because the One-vs-All algorithm often gives higher accuracy.

3. Tools and Technologies

3.1. Scikit Learn

Scikit-learn is a Python-based machine learning library that is available for free. It includes various classification, regression and clustering algorithms such as support-vector machines, random forests, gradient boosting, k-means, and DBSCAN, and is designed to work with the Python numerical and scientific libraries NumPy and SciPy [1].

3.2. Yellowbrick: Machine Learning Visualization

Yellowbrick is a pure Python open source project that adds visual analysis and diagnostic capabilities to the scikit-learn API. The Yellowbrick API additionally wraps matplotlib to create publication-ready figures and interactive data explorations while retaining fine-grained control over figures [2].

3.3. Seaborn: Statistical Data Visualization

Seaborn is a matplotlib-based Python data visualization package. It has a high-level interface for creating visually attractive and useful statistics visuals [3].

3.4. Matplotlib

Matplotlib is a Python package that allows you to create static, animated, and interactive visualizations [4].

3.5. Joblib

Joblib is a set of tools for lightweight pipelining in Python. Joblib provides specific optimizations for numpy arrays and is built to be fast and robust on huge data [5].

3.6. NumPy

NumPy is a Python-based open source project that aims to make numerical computing easier. It was established in 2005, based on the Numeric and Numarray libraries' earlier work [6].

3.7. Pandas

Pandas is an open source data analysis and manipulation tool based on the Python programming language that is quick, powerful, versatile, and simple to use [7].

3.8. Statsmodels API

Statsmodels is a Python module that includes classes and functions for estimating a variety of statistical models, executing statistical tests, and exploring statistical data [8].

3.9. ML.NET

ML.NET allows you to incorporate machine learning into .NET applications, whether online or offline. You can use this feature to make automatic predictions based on the data provided to your application. Machine learning applications create predictions based on patterns in the data rather than having to be explicitly programmed [9].

3.10. Model Builder

Model Builder is a user interface tool that allows developers to create, train, and deploy custom machine learning models in .NET applications. It is a Visual Studio extension that generates the proper model depending on the developers' individual requirements using the ML.NET command-line interface and ML.NET AutoML. Model Builder can produce a variety of predictions, known as scenarios, based on the data they receive. Text classification, image classification, value prediction, and recommendation are all common scenarios [10].

3.11. Expo

Expo stands for stable and reliable development for React Native as CLI. The most crucial part of mobile development is to be able to build an app on either Android and iOs development environments.

Expo behaves as a middle layer for both Android SDK and iOS SDK. By this way, react native has been run native-like smoothness and ease. The code has been written once, and runs on every mobile platform [11].

3.12. Font Awesome

Font Awesome stands for a library which provides SVG icons and fonts for our web and mobile applications [12].

3.13. Firebase Tools

Firebase Tools CLI has been used to open embedded youtube players and its playlists. Also, To be able to open office files on our mobile app, Firebase Tools was the only option to take because there is no other option which is compatible with both Android and iOS development environment [13].

3.14. React Navigation

React Navigation stands for a skeleton of drawer based navigation on mobile applications. Single drawer composed of three different main stacks which represent ‘Home’, ‘Courses’ and ‘Events’ Screens.

Performing back button interactions executing as popping the main stacks from ongoing drawer interaction [14].

3.15. Bootstrap

Bootstrap is the css framework for built-in buttons, modals and some other functionalities. We have used some of the buttons and modal templates from Bootstrap 5.

3.16. React

React is the javascript library for creating user interfaces. We have created a website entirely as a react application.

3.17. React PLayer

React player is the npm tool where you can embed the video player in your website and by giving the url or local target file you can play it in your website with this extension.

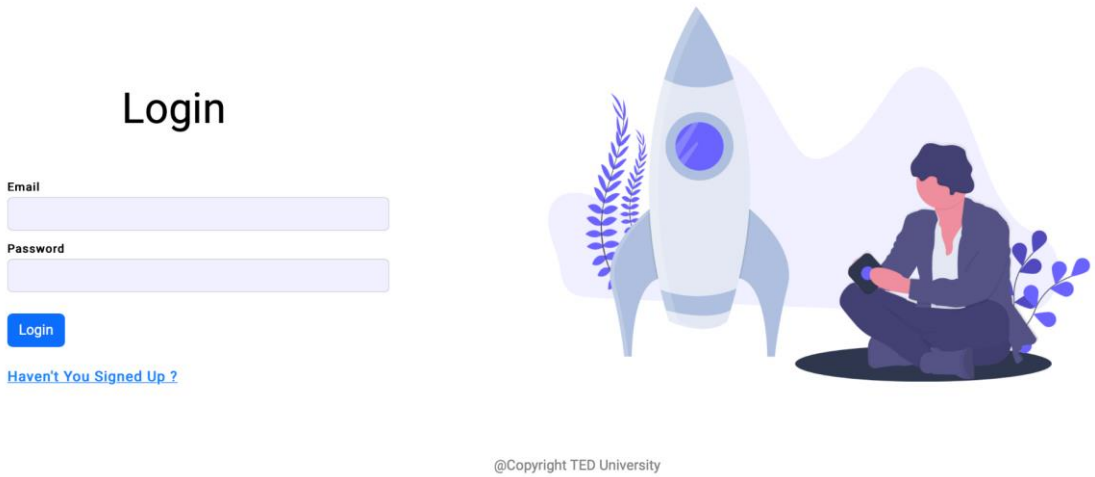
4. User Manual

4.1. User Manual For Web Application

The following images are screenshots from the actual admin website application for explaining the process of how to use the website properly.

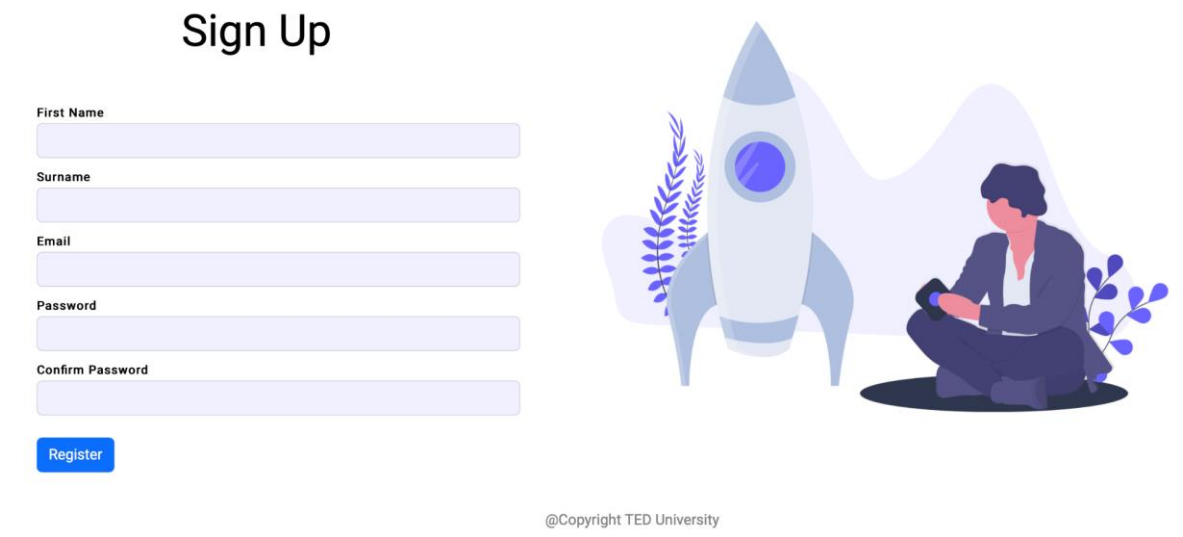
Login Screen

Login page is the opening page for our application. User needs to enter their registered Email and Password and if it matches with our data it will redirect you to the homepage. Otherwise it will show a popup about an email or password being wrong.



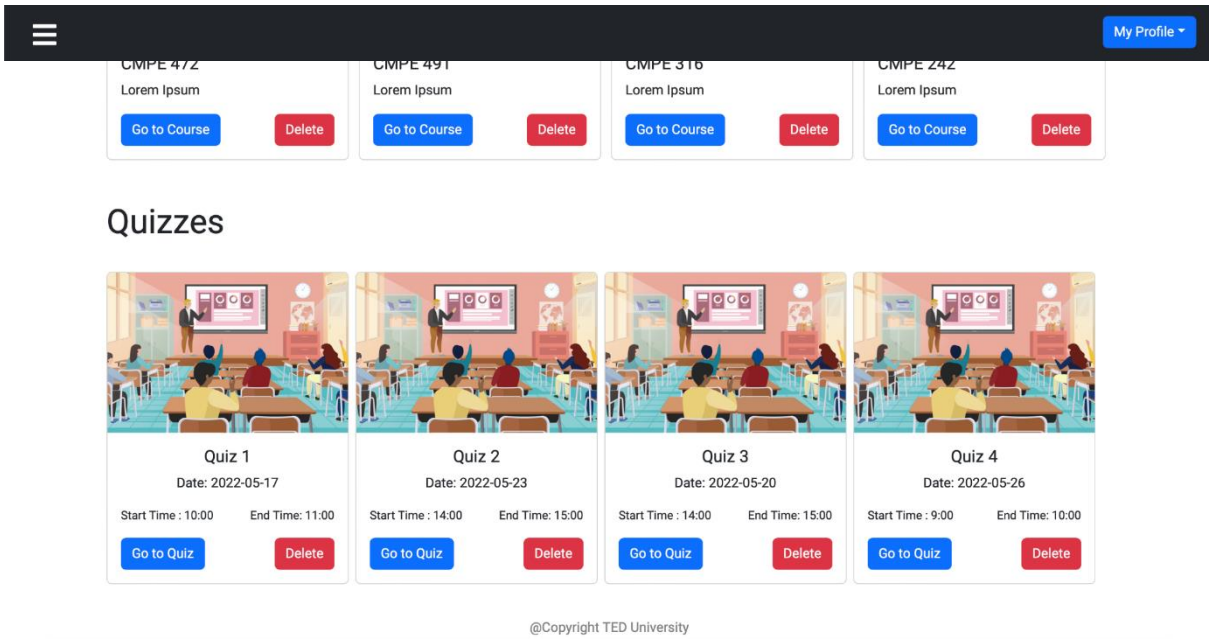
Signup Screen

Signup Page is the account creation page. If you dont have an account you can fill in the blank and you can get a registered account for this application.



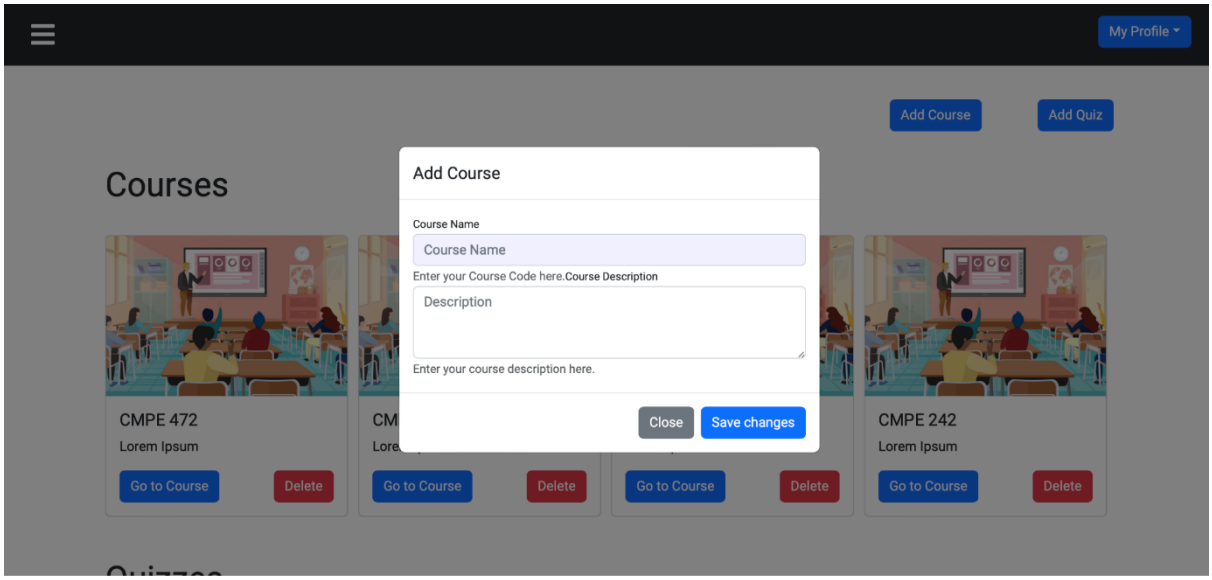
Homepage Screen

In this screen Admin will be able to see the created courses and quizzes. Also they can create a new course or quiz. Or they can delete existing courses and quizzes.



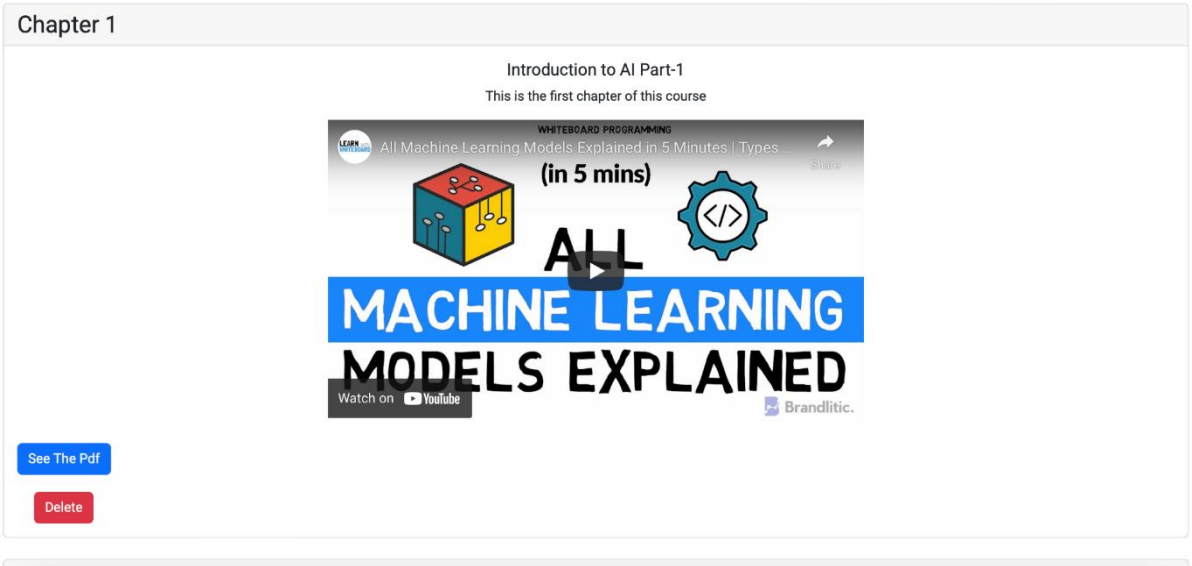
Add Course Page

In this page users will be able to create the course template by naming the course and their description. Furthermore they will be able to add chapters in this course in the next sections.



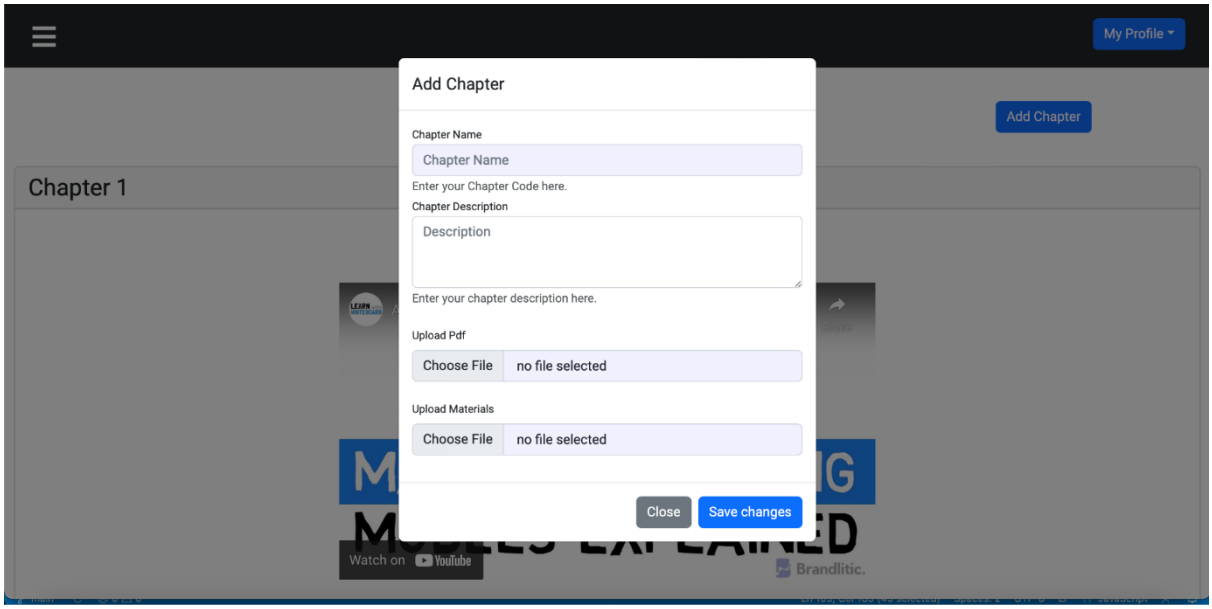
Chapters Page

In this page users will be able to see the chapters that are related with the clicked course. Then you can view the pdf about the chapters and also watch the videos about it.



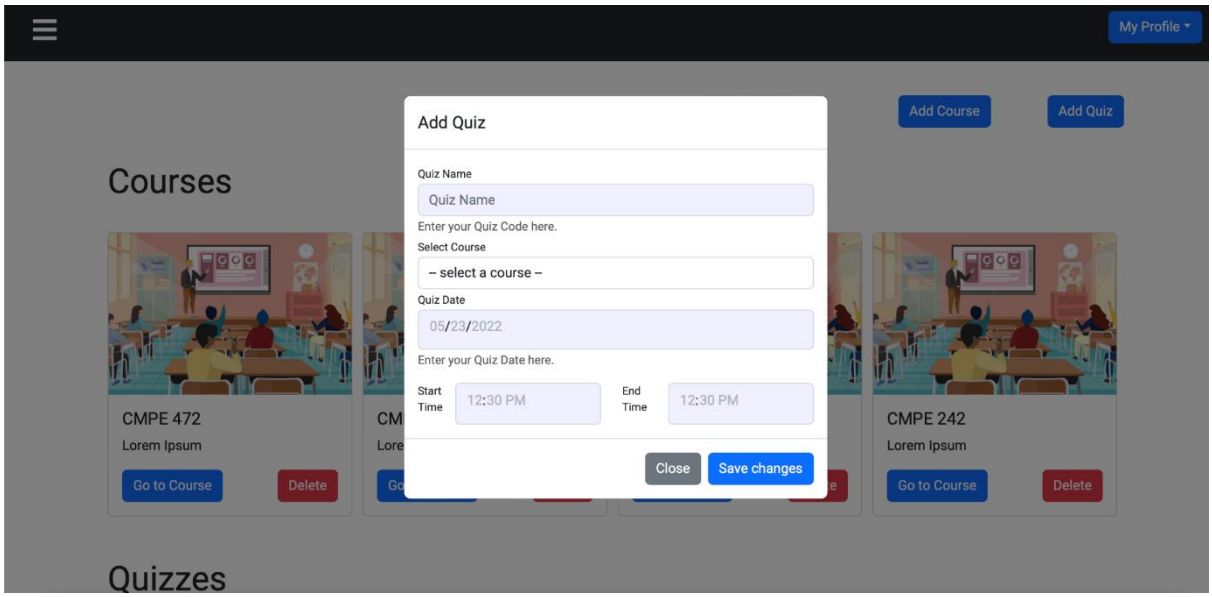
Add Chapter Page

In this page users will be able to add chapters to related courses. They can name the chapter, add description and also they can add some materials about it whether it is video or document.



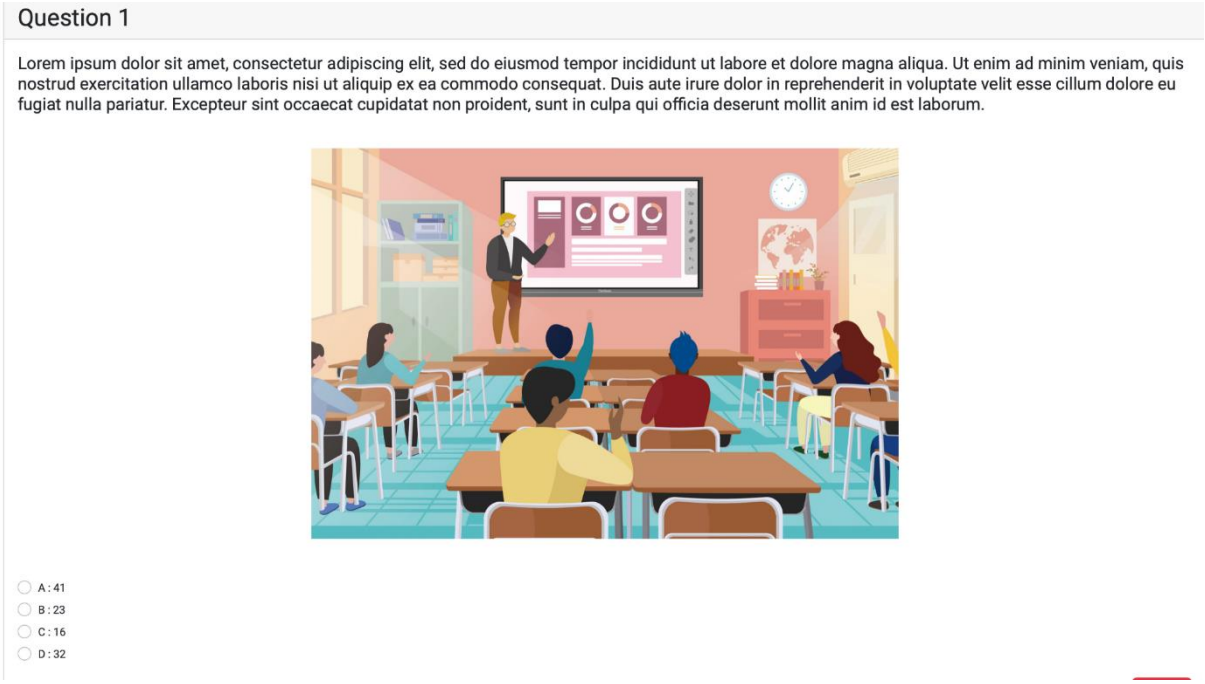
Add Quiz Page

In this page users will be able to create a quiz with specified name , date , start time and end time. After that they can go into the quiz and add questions.



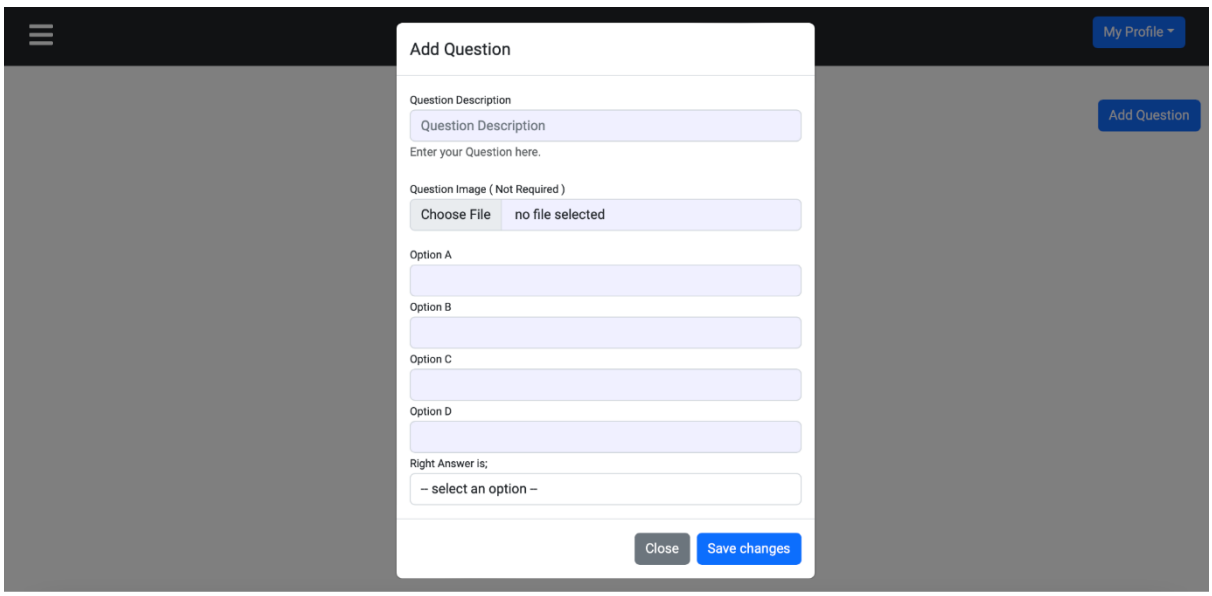
Questions Screen Page

In this page users will be able to see created questions that are related to the specified quiz. also they can create new questions by pressing the button named add question.



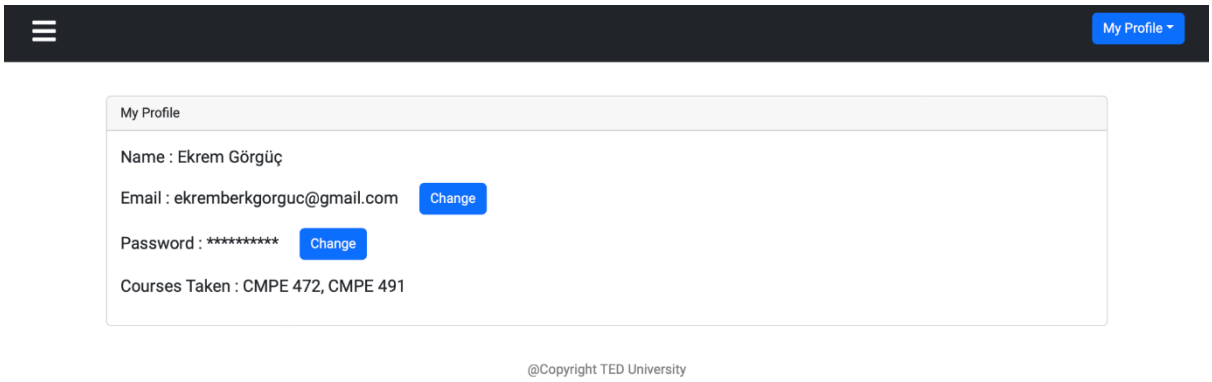
Add Question Page

In this page users will be able to create a question that is related to the specified quiz. They can add questions. Also they can customize the question by adding some images.



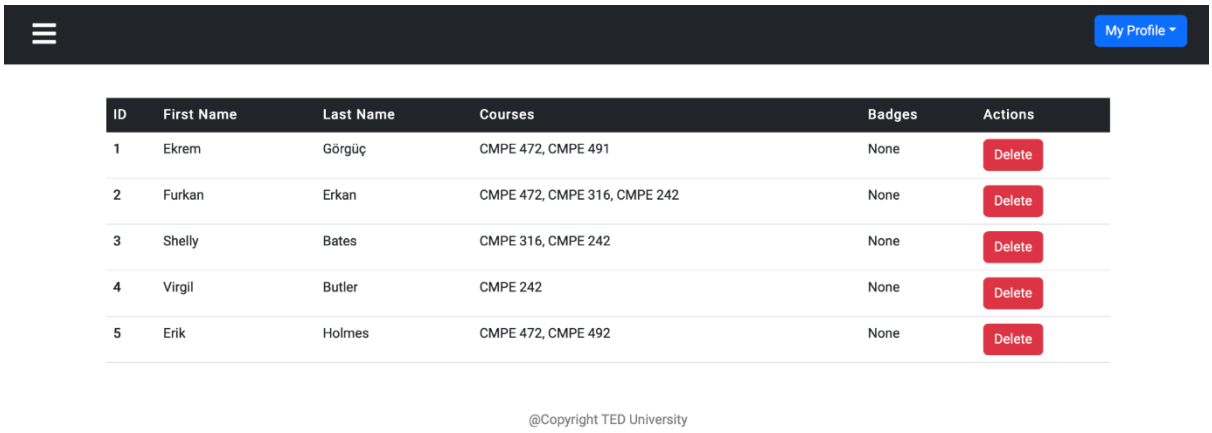
My Profile Screen

In this page users will be able to see their account details. And they can change their passwords or emails by pressing the change button.



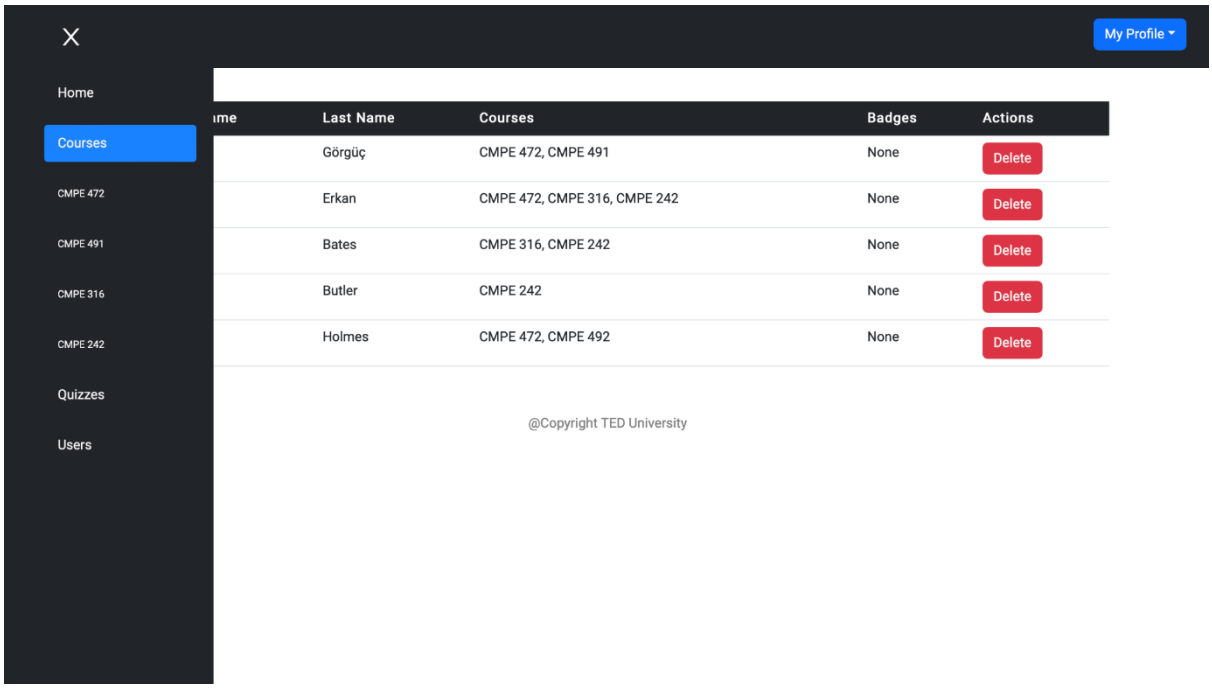
Users Screen

In this page users will be able to see registered accounts in the system. They can delete the person if they want it to.



Sidebar Screen

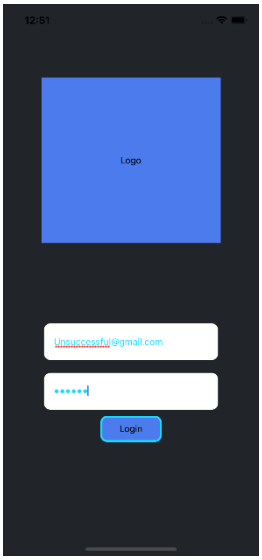
In this page users will be able to navigate easily between courses,quizzes,homepage and user list. When you click the courses it will show all available courses. Same goes for the quiz too.



4.2. User Manual For Mobile Application

In this section, the main flow of mobile application and its visual expression will be demonstrated.

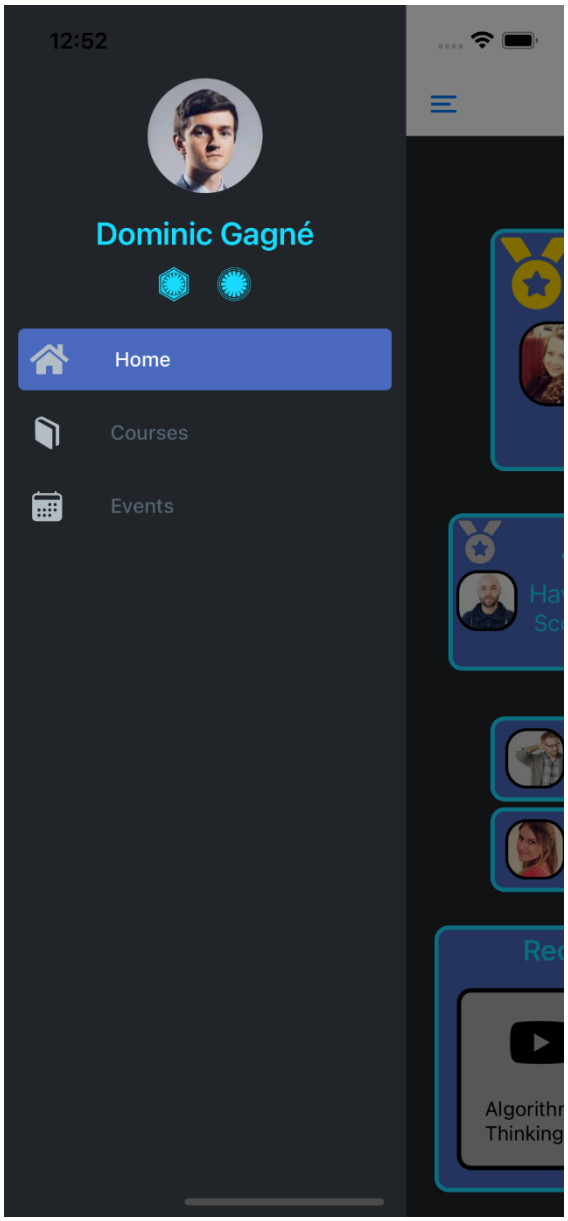
Login Screen



Drawer Navigation

All the navigation through the mobile application is performed by Drawer Navigation. Participants would select the desired screen when pressing the menu button from the header or just swipe from left frame to right on the screen a little bit.

In addition, Profile Photo, Name, Surname and the badges that were acquired before positioned at the top of the drawer.



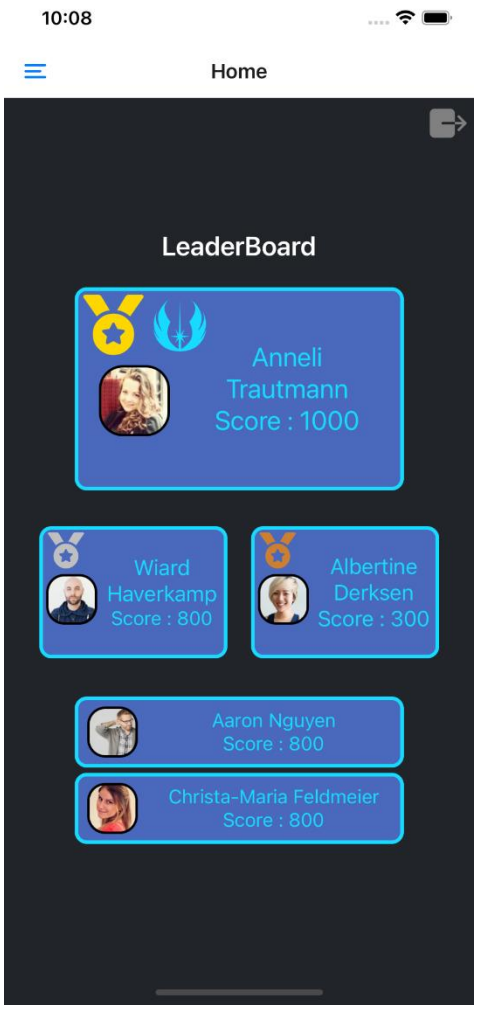
Home Screen

There are two pages for participants with respect to their success levels which have been determined via the Machine Learning process stated already. The main difference between two is If the participant specified as Unsuccessful then Recommended Materials emerges on the Home Screen in order to provide more embracive education.

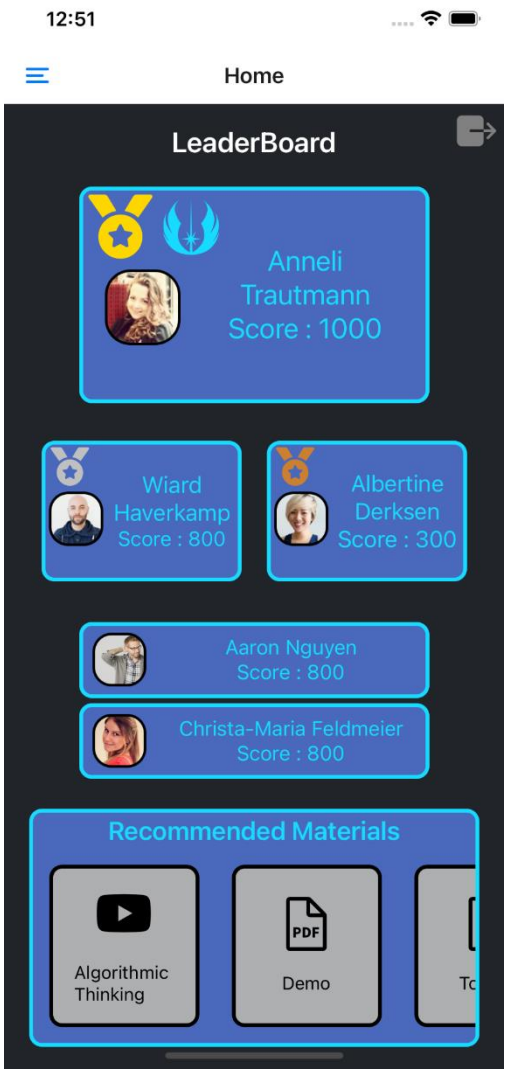
Moreover, Under the gamification intention, Home Screen welcomes participants with LeaderBoard which contains top five competitors. Also, Top three participants will get medals in the area reserved for them. However, the only victor can have the badge for their success.

On top of it, participants would logout from the session by pressing the button located just below the right side of the header.

Home Screen for Successful / Average Participants

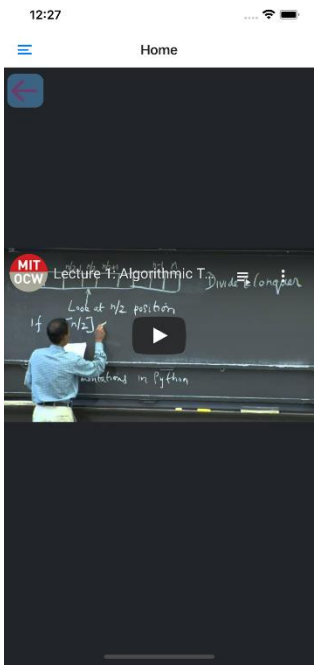


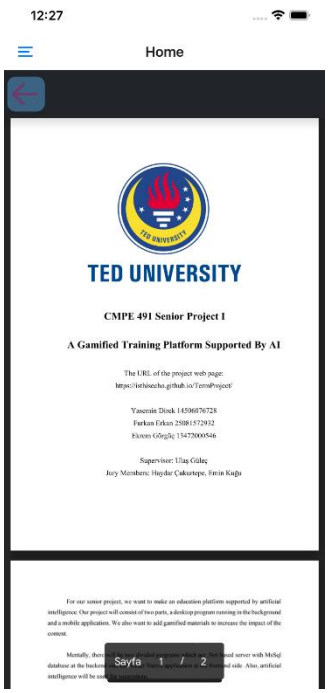
Home Screen for Unsuccessful Participants

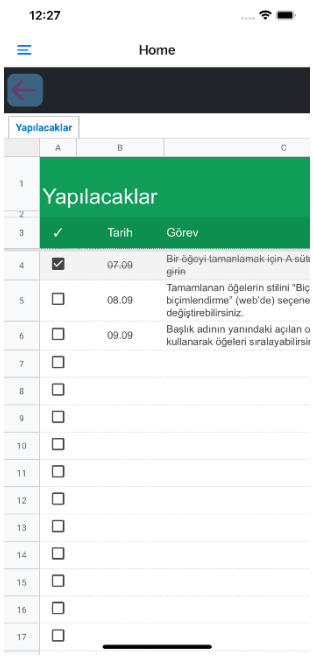


Recommended Materials

Participants would select a material from the Recommended Materials area. Material can be either a Youtube video with playlist, Pdf file, Word file, Excel file or Powerpoint slide.







Courses Screen

Participants would select Courses Screen through drawer navigation.

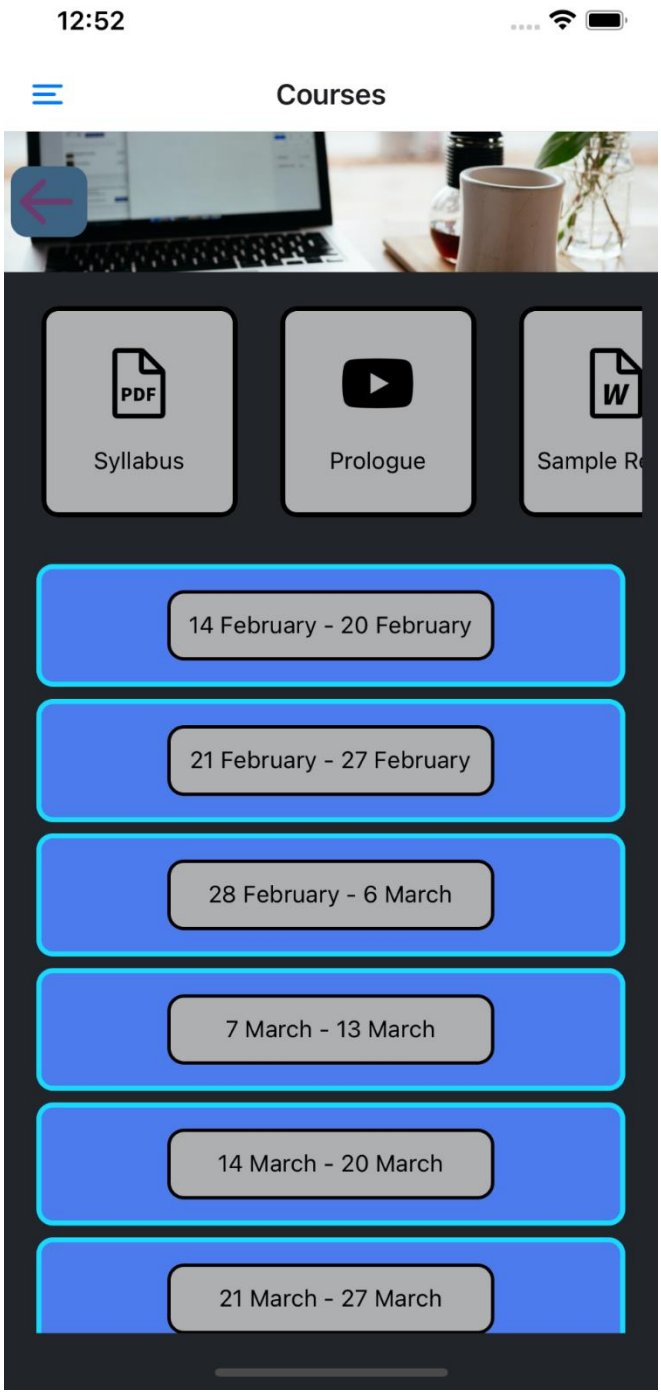
Course Selection Screen

Participants would select the desired Course that offered for them through animated flatlist.

Specific Course Content Screen

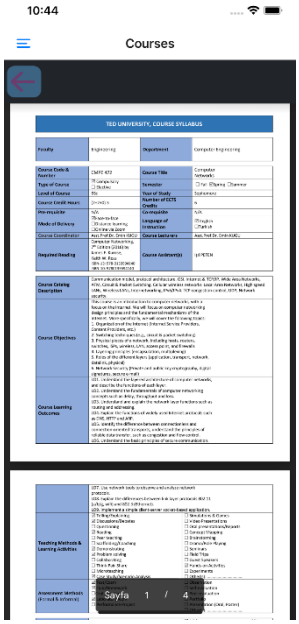
There are two choices the participant would make;

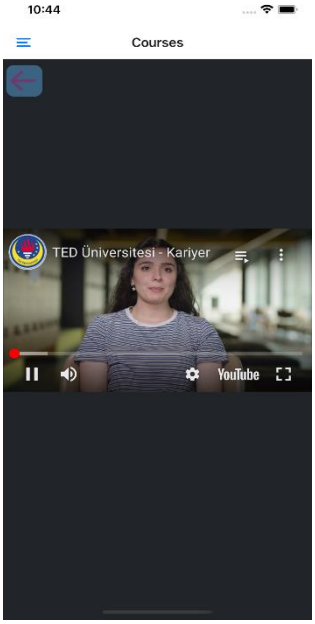
- Participants would access the materials which are not included in the curriculum.
- Participants would access the materials which are included in the curriculum separated by weeks.

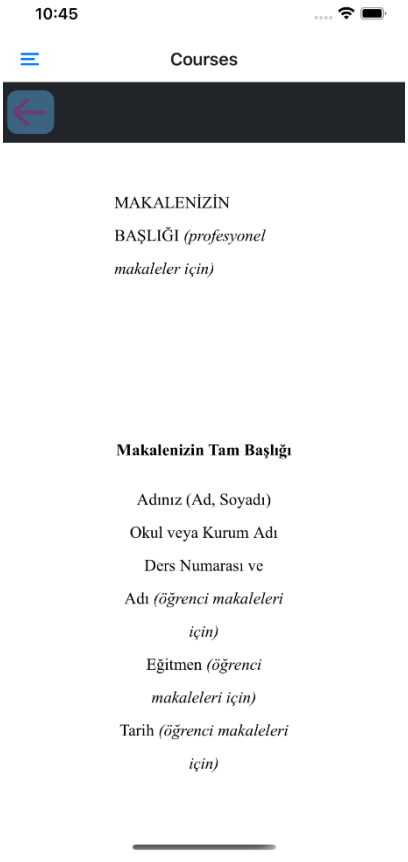


Specific Course Content Screen without Week Selection not included in the curriculum

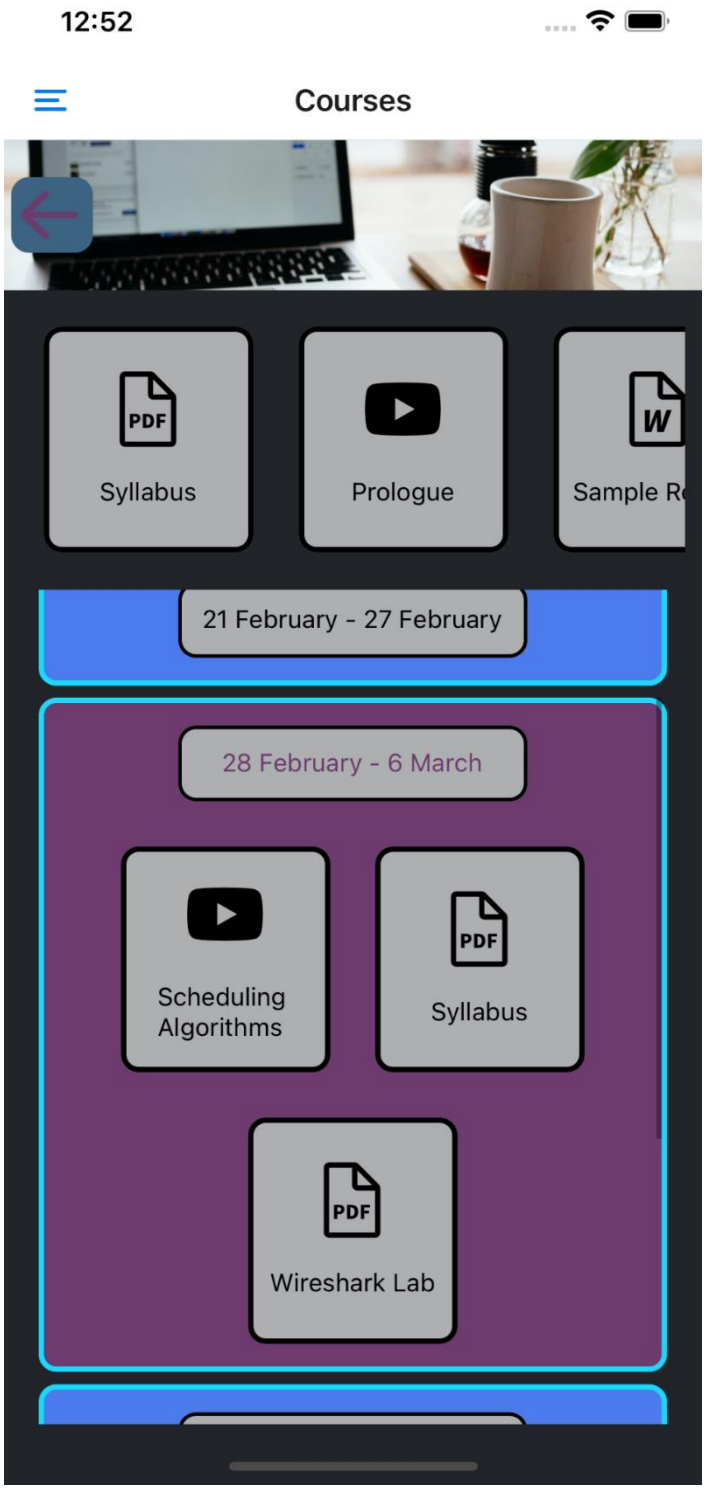
Material can be either a Youtube video with playlist, Pdf file, Word file, Excel file or Powerpoint slide.







Specific Course Content Screen with Week Selection

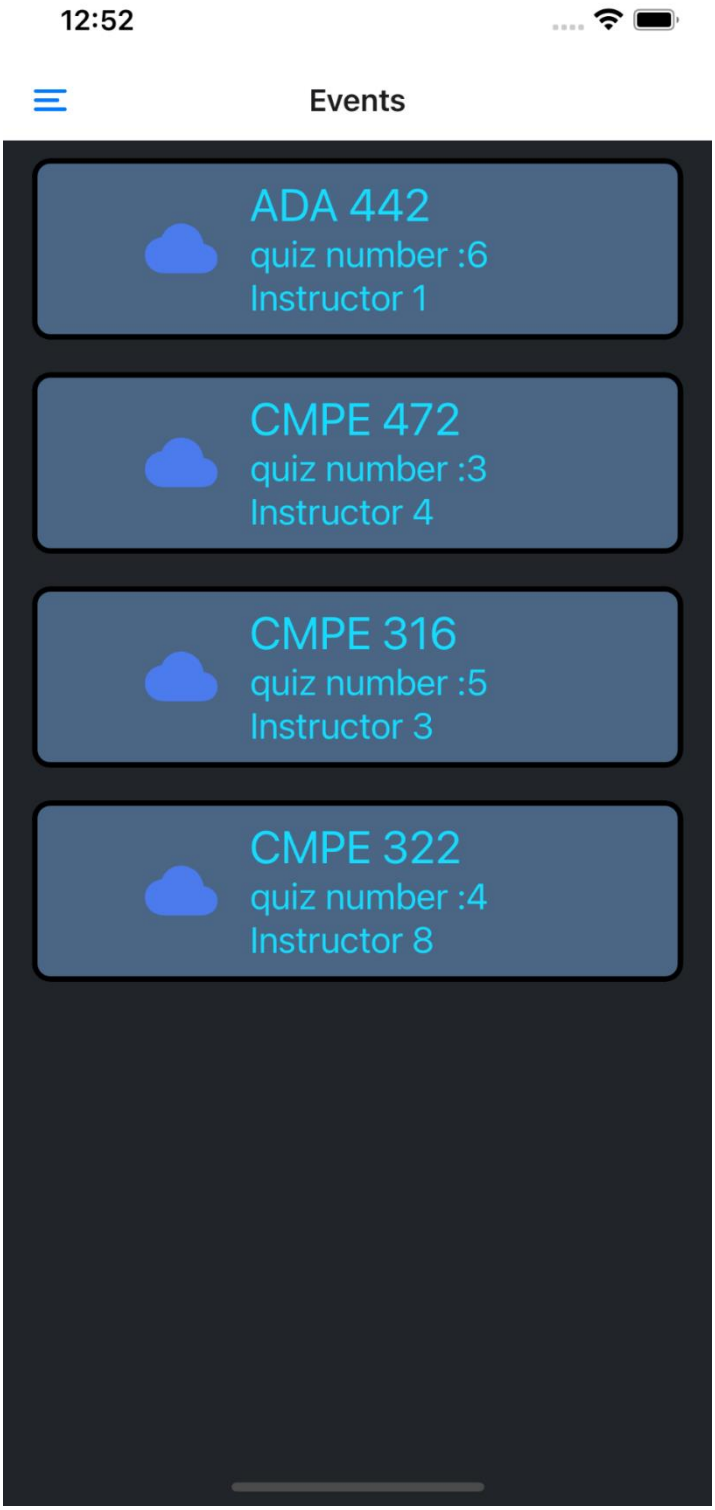


Events Screen

Participants would select Event Screen through drawer navigation.

Events Selection Screen

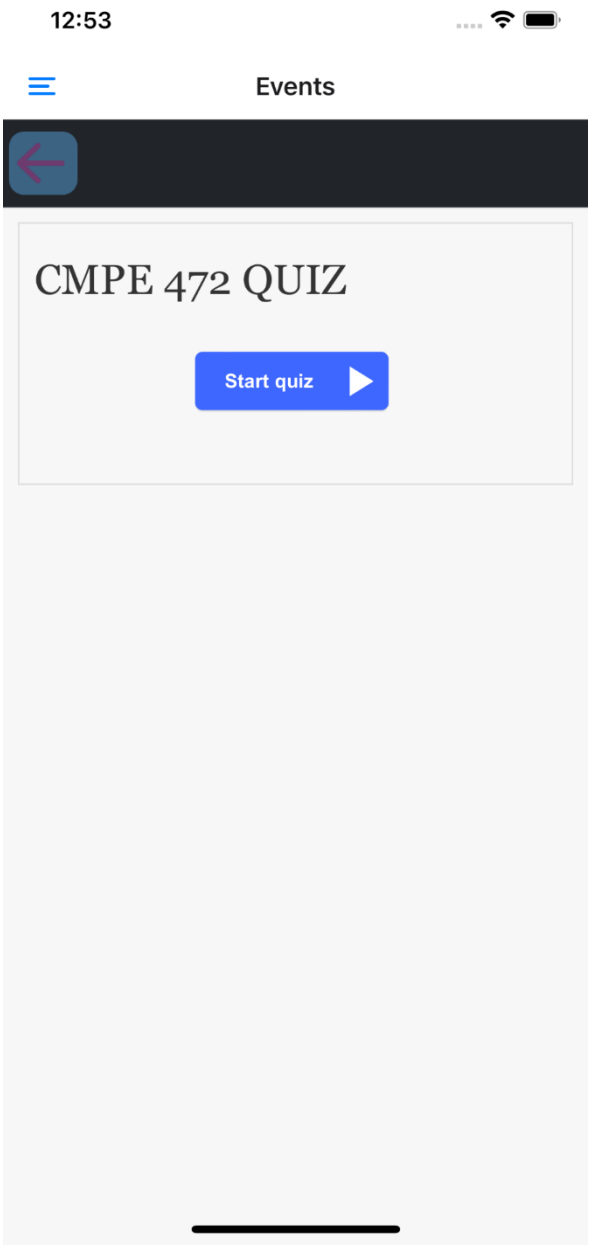
Participants would select the desired event through Flatlist.



Specific Event Content Screen

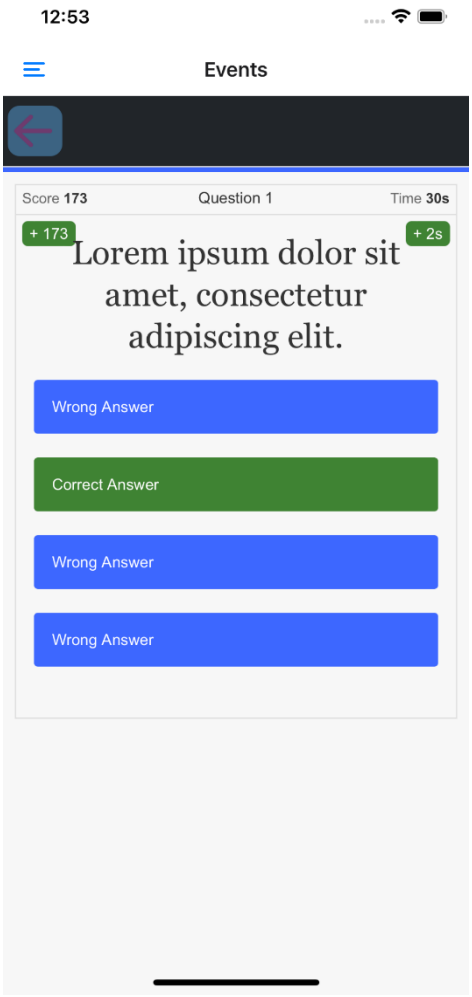
Start Quiz

Participants would start the desired event by pressing a button.

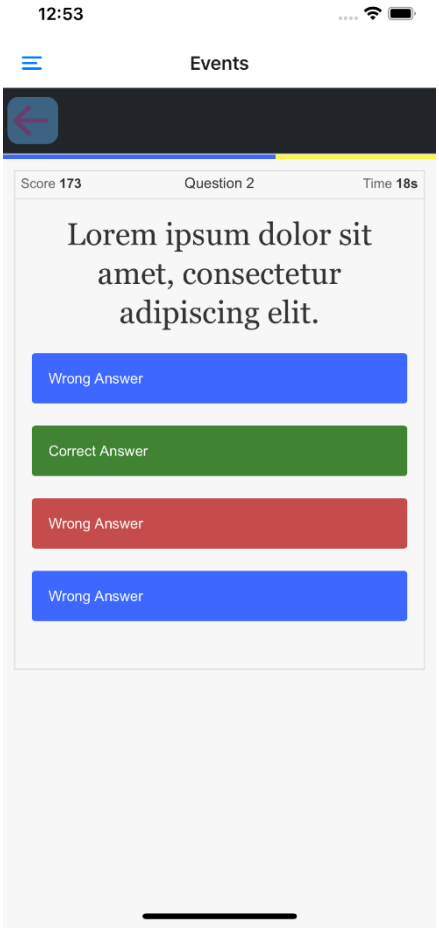


Perform Quiz

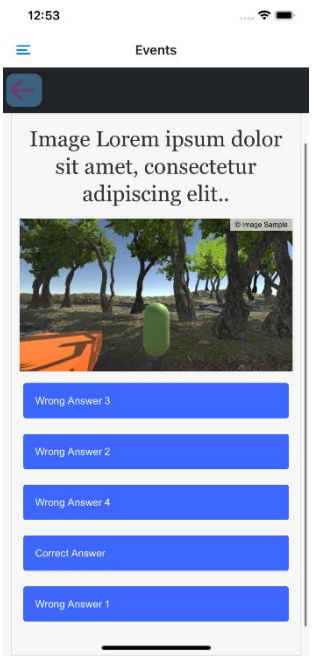
If the answer of participants is correct, participants will have the questions' points and bonus time.



If the answer of participants is wrong, participants will not have the questions' points and be punished with negative time.

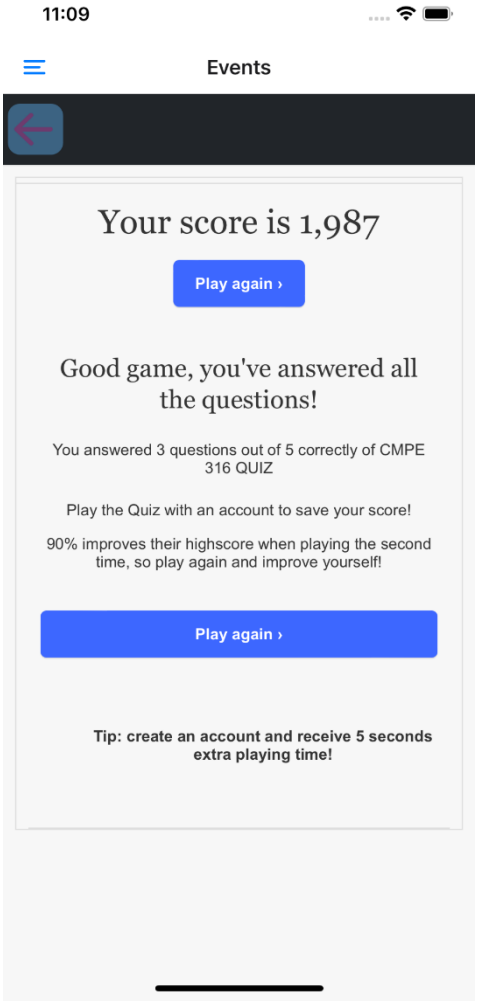


Instructors would add image and multiple selection speciality



Quiz Score Announcement

Participants would see their score at the end of the quiz.



5. Impact of Engineering Solutions

5.1. Economic Impacts

Our application allows to teach students without requiring any physical school environment. In addition, by presenting course materials to students online, it prevented students from paying fees to access these materials. By using the application, students can access the course material they want at any time, and they can realize their deficiencies according to their performance in the exams and study these subjects.

5.2. Environmental Impacts

The fact that our application removes the need for physical course materials has positive effects on the environment. It avoids the waste of paper both by taking the exams online and presenting the course materials online. It contributes to preventing paper waste by providing an online exam system instead of papers that will have no meaning after the exam is over.

5.3. Social Impacts

One of the most important effects of our project is its social effects. There is a bias towards online education on the grounds that it reduces students' interaction with the course. This bias prevents the widespread use of online education platforms. Our project, on the other hand, aims to make learning fun and more interactive by creating a competitive environment. Thanks to the leaderboard and badge system it contains, it aims to attract the attention of students and encourages the use of the application by providing personal recommendations according to their performance in the exam.

6. References

[1] https://www.scikit-yb.org/en/latest/

[2] https://www.scikit-yb.org/en/latest/

[3] https://seaborn.pydata.org/

[4] https://matplotlib.org/

[5] https://joblib.readthedocs.io/en/latest/

[6] https://numpy.org/

[7] https://pandas.pydata.org/

[8] https://www.statsmodels.org/stable/index.html

[9] https://docs.microsoft.com/en-us/dotnet/machine-learning/

[10] https://docs.microsoft.com/en-us/dotnet/machine-learning/automate-training-with-model-builder

[11] https://docs.expo.dev/

[12] https://fontawesome.com/

[13] https://www.npmjs.com/package/firebase-tools

[14] https://reactnavigation.org/

[15] https://getbootstrap.com

[16] https://tr.reactjs.org

[17] https://www.geeksforgeeks.org

[18] https://stackoverflow.com