



CS 353 Term Project

Final Report

Section 1
Group 35

Group Members:

- | | |
|--------------------|----------|
| ● Mert Şen | 21802602 |
| ● Arda Atahan İbiş | 21901941 |
| ● Şebnem Türkoğlu | 21901819 |

Topic: Online Language Learning Platform

Link: <https://online-language-learning-platform.vercel.app>

Link to ER Diagram:

https://github.com/ardaatahan/Online-Language-Learning-Platform/blob/main/diagrams/final_er_diagram.md

1. Description

The application is an online language learning platform for a course center. The functionalities of the center are implemented in the system to provide a database system for a center. The system allows different user types (admin, language native, teacher, and student) and has functionalities specific for each user type. Admins view reports and general statistics about the grades of the students and enrollment data for a specific language. Language natives have the ability to hold speaking classes with request sending students. They also have the right to decline the sent requests. On the other hand, teachers have the right to hold classes with specific books, enrollment quotas and homework assignments. Additionally, students may also request classes. Students are the main users of the system where they can enroll into classes, have speaking lessons with native speakers, are assigned homework, receive grades for each homework or speaking lessons. To support each of the users types, there are different signup pages.

2. Contribution of Each Teammate

- **Şebnem Türkoğlu**

Project Proposal

- ER diagram

Design Report

- Revised ER diagram
- GUI design

Implementation

- Frontend of the system
- SQL queries for the database

- **Arda Atahan İbiş**

Project Proposal

- ER diagram
- Project description

Design Report

- Revised ER diagram
- SQL queries

Implementation

- Backend of the system

- SQL queries for the database
- Model creation

- **Mert Sen**

Project Proposal

- ER diagram
- Requirements

Design Report

- Revised ER diagram
- Relational models

Implementation

- Backend of the system
- SQL queries for the database
- Model creation

3. Implementation Details

For our project, we have chosen React framework for the frontend, Django for the backend, and PostgreSQL for the database. We also utilized Redux and Axios packages to connect our frontend application to our backend application. We connected our database to the backend application in `settings.py` document where we specified the engine, account, and other necessary information. Each type of model's operations are done by their own views which implement Rest Framework's views. To execute SQL queries we get a connection object from '`django.db.connection`' and by using '`.cursor()`' method we create a cursor object. Using this object we can write raw SQL queries and call '`.execute()`' method to execute a query. All of the functionality is executed by raw SQL queries we wrote and does not depend on any of Django's automatic functions. The GUI of the website is built with React components and MaterialUI library. To send requests from the frontend to backend we have used Axios packages, and to make api calls we have also used the Redux framework. In order to send and receive requests, we have added relevant routes to the backend and connected them to the database via Django views. In order to send and receive JSON response objects, custom serializers were also implemented. Authentication has been managed by tokens that are handled by Django. Authorized requests are to be sent with a token header value in order to be processed.

4. Advanced DB Components

a) Reports:

Report For Finding Number of Students Learning a Specific Language:

```
SELECT COUNT(DISTINCT S.user_id) FROM levelup_api_student S WHERE
    S.user_id IN (SELECT S2.user_id FROM levelup_api_student S2, levelup_api_class C,
    levelup_api_takes T WHERE T.student_id = S2.user_id AND C.language_id = %s) OR
    S.user_id IN
        (SELECT S3.user_id FROM levelup_api_student S3, levelup_apiSpeaking_exercise Sp
    WHERE S3.user_id = Sp.student_id AND
    Sp.language_id = %s)
```

Report For Finding The Average Grade of Students taken Speaking Exercises in a Specific Language:

```
SELECT AVG(Sp.grade) FROM levelup_apiSpeaking_exercise Sp
    WHERE Sp.language_id = %s AND Sp.grade IS NOT NULL
```

Report For Finding The Average Grade of Students taken Homework From a Class With a Specific Language and Level:

```
SELECT AVG(H.grade) FROM levelup_api_student S, levelup_api_homework H
    WHERE H.given_class_id IN (SELECT C.id FROM levelup_api_class C WHERE
    C.level_id = %s AND C.language_id = %s)
    AND H.grade IS NOT NULL
```

b) Query Including Like:

```
SELECT FT.id, FT.topic_title, U.name, FT.datetime FROM
    levelup_api_forum_topic FT, levelup_api_user U WHERE U.id = FT.topic_owner_id
    AND FT.topic_title LIKE '\% %s \%'
```

c) Queries Including Group By, Avg, Count:

```
SELECT AVG(Sp.grade) FROM levelup_apiSpeaking_exercise Sp
    WHERE Sp.language_id = %s AND Sp.grade IS NOT NULL
```

```
WITH native_rating(language_native_id, avg_rating) AS
```

```
(
```

```

        SELECT Sp.language_native_id, AVG(R.rate)
        FROM levelup_api_rate_exercise_details R, levelup_api_speaking_exercise Sp
        WHERE R.speaking_exercise_id = Sp.id
        GROUP BY Sp.language_native_id
    )

    SELECT U.name, N.avg_rating, L.description FROM levelup_api_user U,
native_rating N, levelup_api_language_native L
        WHERE L.user_id = %s AND U.id = L.user_id AND
native_rating.language_native_id = U.id

WITH native_rating(language_native_id, avg_rating) AS
(
    SELECT Sp.language_native_id, AVG(R.rate)
    FROM levelup_api_rate_exercise_details R, levelup_api_speaking_exercise Sp
    WHERE R.speaking_exercise_id = Sp.id
    GROUP BY Sp.language_native_id
)

SELECT U.id, U.name, N.avg_rating FROM levelup_api_user U, levelup_api_speaks S,
levelup_api_language La
    native_rating N WHERE N.language_native_id = S.language_native_id AND La.id =
S.language_id AND U.id = S.language_native_id AND La.lang_name = %s

WITH native_rating(language_native_id, avg_rating) AS
(
    SELECT Sp.language_native_id, AVG(R.rate)
    FROM levelup_api_rate_exercise_details R, levelup_api_speaking_exercise Sp
    WHERE R.speaking_exercise_id = Sp.id
    GROUP BY Sp.language_native_id
)

SELECT U.name, N.avg_rating FROM levelup_api_user U, native_rating N WHERE
N.language_native_id = U.id

WITH native_rating(language_native_id, avg_rating) AS
(

```

```

SELECT Sp.language_native_id, AVG(R.rate)
FROM levelup_api_rate_exercise_details R, levelup_apiSpeaking_exercise Sp
WHERE R.speaking_exercise_id = Sp.id
GROUP BY Sp.language_native_id
)

SELECT U.name, N.avg_rating FROM levelup_api_user U, native_rating N WHERE
N.language_native_id = U.id

```

d) View:

```

CREATE VIEW student_classes as SELECT * FROM levelup_api_student S,
levelup_api_takes T, levelup_api_class C WHERE S.user_id = @student_id AND
T.student_id = @student_id AND C.id = T.class_id

```

This view combines all of the classes a student takes.

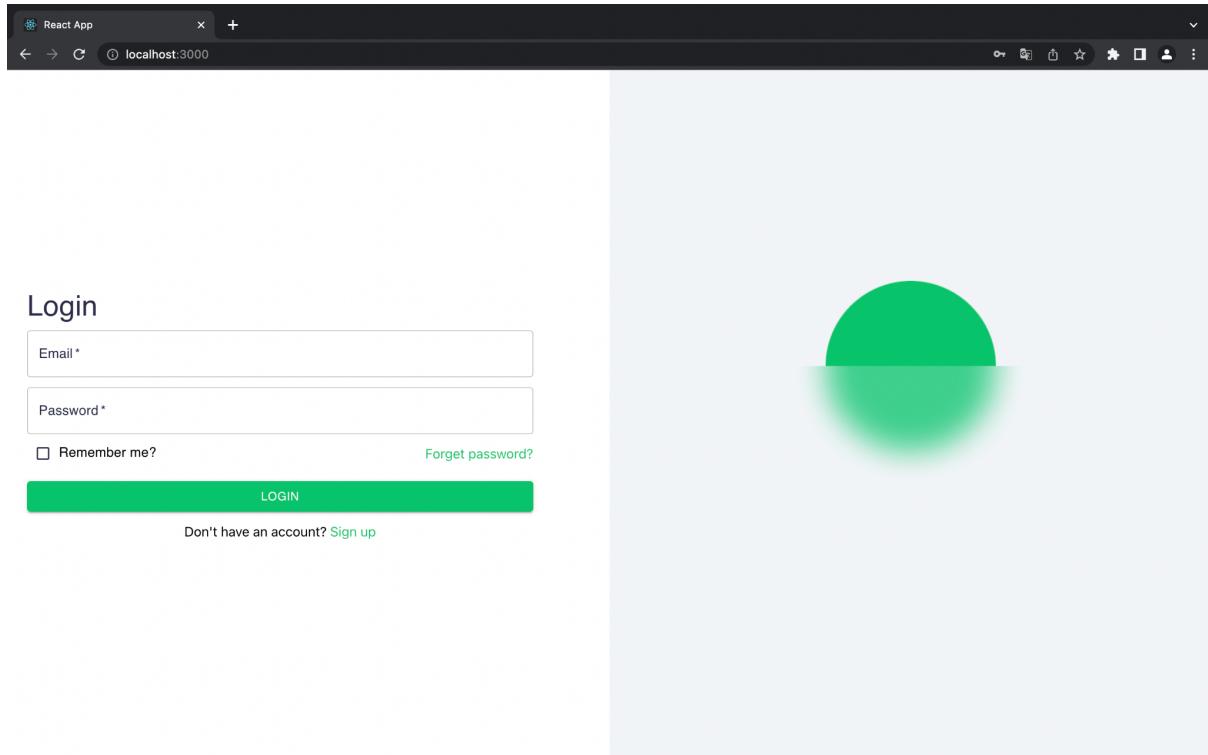
e) Triggers, Constraints, Secondary Indices:

We have a trigger that functions when a class or exercise is rated by a student. We update corresponding average rating whenever a new rate is entered.

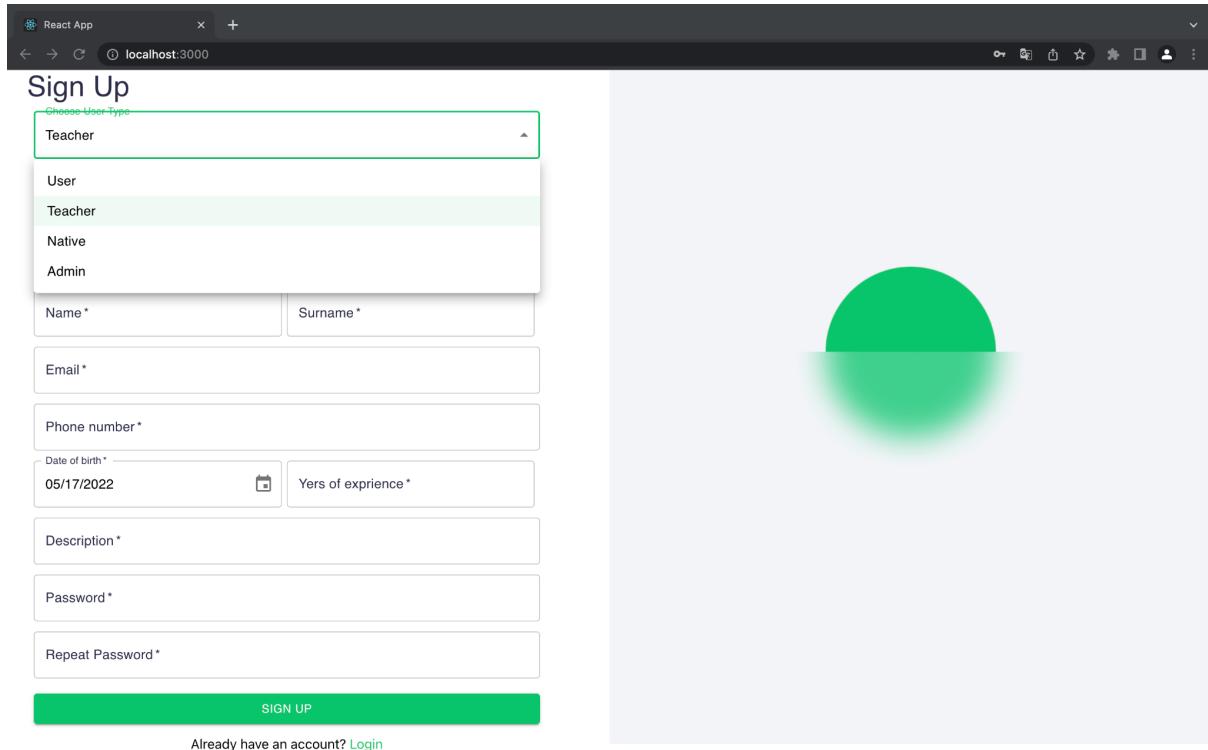
We have constraints such as ‘Not Null’ and ‘UNIQUE’ to check the value to be added to the database.

We used PostgreSQL, and in PostgreSQL, every indices other than the primary ones are secondary indices. Therefore, this was done internally. We did not need to implement any external index feature.

User Manual

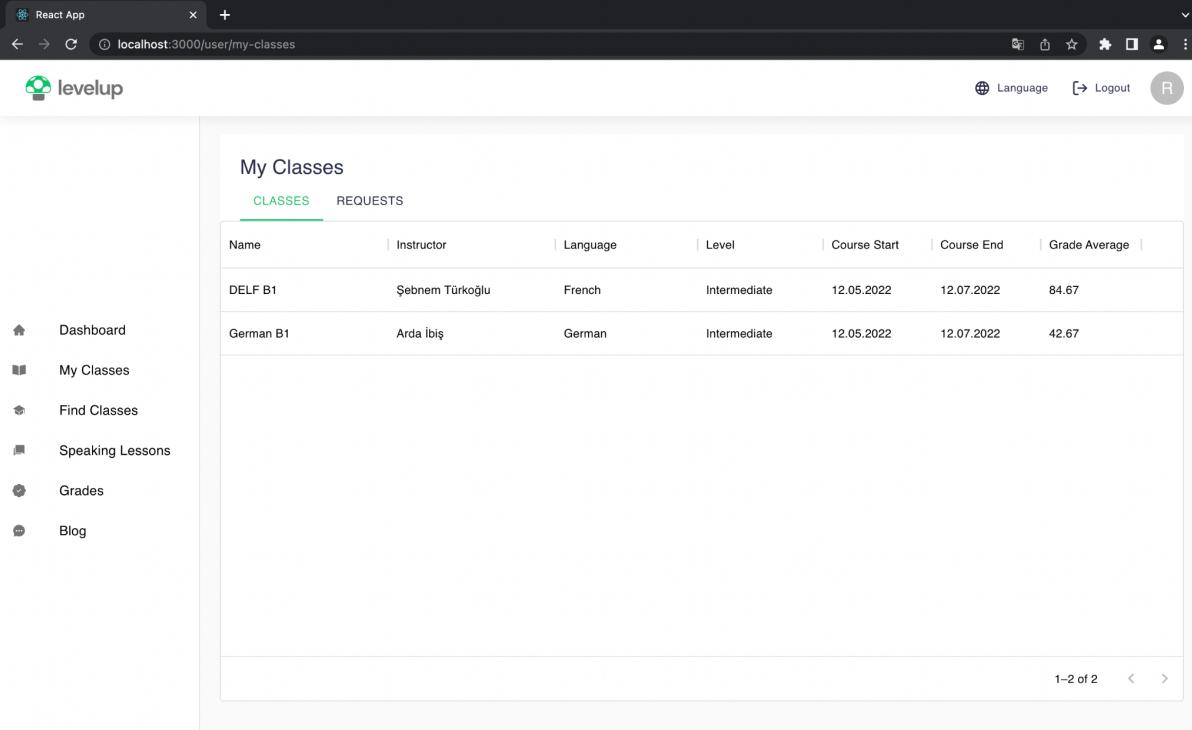


Users can login from the login page, or click the 'Sign up' button below the page and create a new account.



In the Sign Up page the user chooses the account type and fills the necessary information of the system.

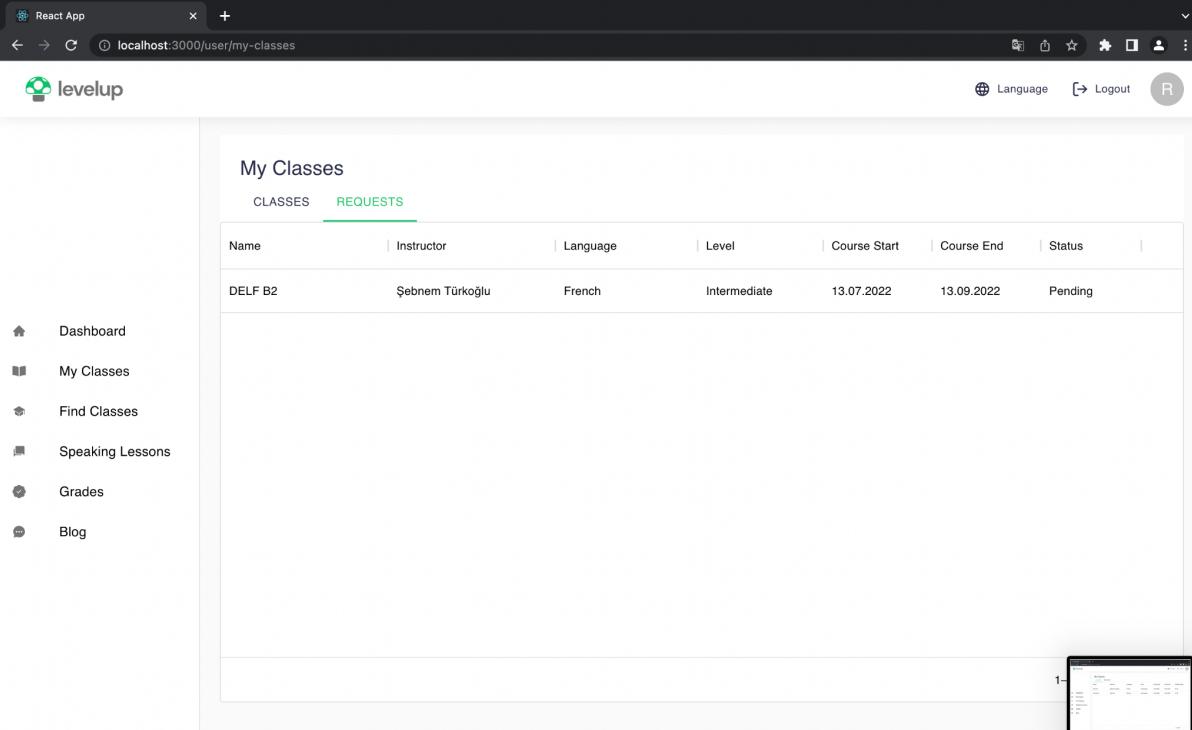
● User View



The screenshot shows a user interface for managing classes. On the left, a sidebar lists navigation options: Dashboard, My Classes (which is selected), Find Classes, Speaking Lessons, Grades, and Blog. The main content area is titled 'My Classes' and contains a table with two rows of data. The table has columns for Name, Instructor, Language, Level, Course Start, Course End, and Grade Average. The first row represents a class named 'DELF B1' taught by 'Şebnem Türkoğlu' in French at an intermediate level, starting on 12.05.2022 and ending on 12.07.2022, with a grade average of 84.67. The second row represents a class named 'German B1' taught by 'Arda İbiş' in German at an intermediate level, starting on 12.05.2022 and ending on 12.07.2022, with a grade average of 42.67. At the bottom right of the table, there is a pagination indicator '1-2 of 2' with arrows.

Name	Instructor	Language	Level	Course Start	Course End	Grade Average
DELF B1	Şebnem Türkoğlu	French	Intermediate	12.05.2022	12.07.2022	84.67
German B1	Arda İbiş	German	Intermediate	12.05.2022	12.07.2022	42.67

From My Classes page in the sidebar, users can view the current classes they take and their information including their grade average.



The screenshot shows the same user interface as the previous one, but the 'REQUESTS' tab is now selected in the top navigation bar. The main content area is titled 'My Classes' and contains a table with one row of data. The table has columns for Name, Instructor, Language, Level, Course Start, Course End, and Status. The single row represents a class named 'DELF B2' taught by 'Şebnem Türkoğlu' in French at an intermediate level, starting on 13.07.2022 and ending on 13.09.2022, with the status listed as 'Pending'. A small inset window in the bottom right corner shows a preview of another part of the application, possibly a dashboard or report.

Name	Instructor	Language	Level	Course Start	Course End	Status
DELF B2	Şebnem Türkoğlu	French	Intermediate	13.07.2022	13.09.2022	Pending

From the requests tab user can view the class requests they made previously and their status.

The screenshot shows a user interface for finding classes. On the left, a sidebar menu includes options like Dashboard, My Classes, Find Classes, Speaking Lessons, Grades, and Blog. The main area is titled 'Find Classes' and has dropdown filters for 'Language' (French) and 'Level' (Intermediate). A course card for 'Writing for DELF B1' by Sırma Kuzukulak is displayed, showing a duration of 6 weeks. To the right, detailed information about the course is shown, including its description ('This course will make you a god in French'), duration (6 weeks), required books ('Français Pour Chats - Un Chat'), weekly plan (Week 1: Chapter 1 & 2, Week 2: Chapter 3, etc.), and start and end dates (03.04.2022 and 15.05.2022). A green 'REQUEST CLASS' button is at the bottom.

From the Find Classes page, users can view offered classes and can filter by language and level. From the Request Class button the user can create a request.

The screenshot shows a user interface for speaking lessons. On the left, a sidebar menu includes options like Dashboard, My Classes, Find Classes, Speaking Lessons, Grades, and Blog. The main area is titled 'Speaking Exercises' and features a table with columns for Instructor Name, Exercise Date, Exercise Link, Grade, and Rate. The table currently displays 'No rows'. At the top right, there is a 'REQUEST NEW EXERCISE' button. The bottom right corner of the table shows pagination information: '0-0 of 0' with arrows for navigation.

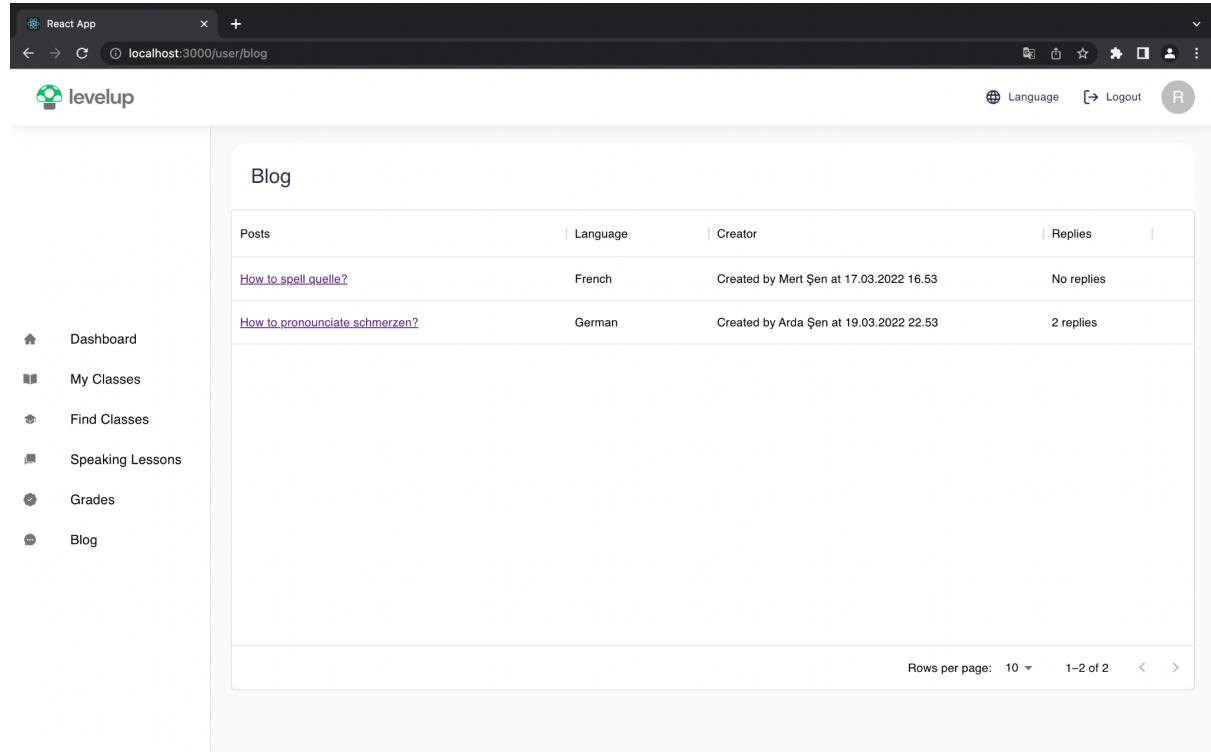
From the Speaking Lessons page user can view their lesson appointments and filter by ‘upcoming’ or ‘all’ tabs. Request tabs show the requests pending for approval.

The screenshot shows a web browser window for a React App at localhost:3000/user/speaking-lessions/new-exercise. The main content area displays a 'Find Classes' search interface with filters for 'Language: French' and 'Level: Intermediate'. Below the filters is a list of five native speakers with their names, average ratings, and small profile icons. To the right of the list is a detailed view for 'Celine Dion', featuring her profile picture, name, a 5-star rating icon, and an average rating of 4.6. A 'Description' section contains a short text about Celine Dion's personality and interests. At the bottom right is a 'REQUEST EXERCISE' button, with a date and time input field set to '13.05.2022 22:14' and a notes field containing the text 'I want to study pronunciations.'.

If the user clicks a request for a new exercise button, they will be directed to a list of language natives where they can filter by language and read reviews of the natives. Users can specify a date and add notes to create a request.

The screenshot shows a web browser window for a React App at localhost:3000/user/grades. The main content area displays a 'Grades' table with three tabs above it: 'EXAMS' (which is active), 'HOMEWORKS', and 'SPEAKING EXERCISES'. The table has columns for Name, Class, Date, Grade, and View Assignment. It lists two assignments: 'Assignment 1' in 'DELF B1' on '12.05.2022' with a grade of '84.67' and 'Assignment 1' in 'German B1' on '12.05.2022' with a grade of '42.67'. The table includes a footer with page numbers '1–2 of 2' and navigation arrows. The left sidebar contains links to 'Dashboard', 'My Classes', 'Find Classes', 'Speaking Lessons', 'Grades', and 'Blog'.

From the Grades page students can see their grades for assignments, exams, and speaking exercises. They can also view assignment documents if the teacher uploaded any documents.

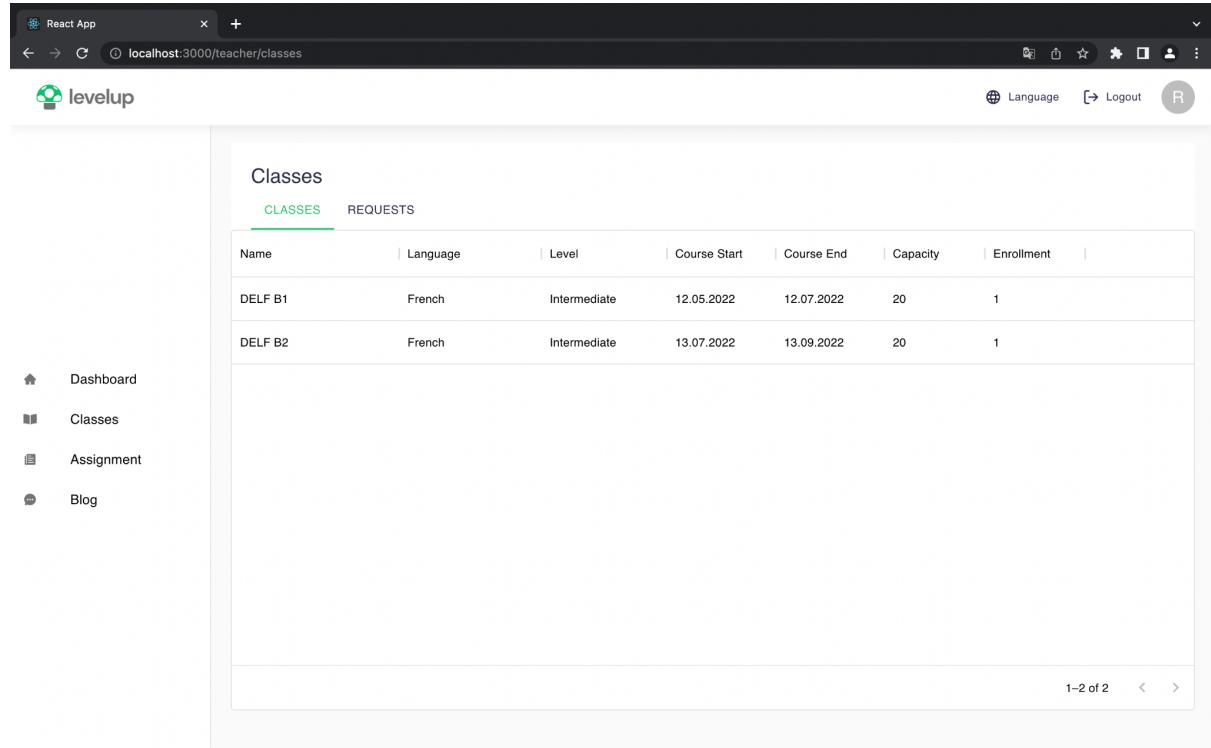


The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/user/blog". The page has a header with a "levelup" logo, language selection, and logout options. On the left is a sidebar with links: Dashboard, My Classes, Find Classes, Speaking Lessons, Grades, and Blog. The main content area is titled "Blog" and contains a table of posts. The first post is "How to spell quelle?", created by Mert Şen at 17.03.2022 16.53, in French, with no replies. The second post is "How to pronounce schmerzen?", created by Arda Şen at 19.03.2022 22.53, in German, with 2 replies. At the bottom right of the table are buttons for "Rows per page: 10" and "1–2 of 2".

Posts	Language	Creator	Replies
How to spell quelle?	French	Created by Mert Şen at 17.03.2022 16.53	No replies
How to pronounce schmerzen?	German	Created by Arda Şen at 19.03.2022 22.53	2 replies

From the Blog page, users can see or write blog posts to ask their questions. This page is the same for all user types.

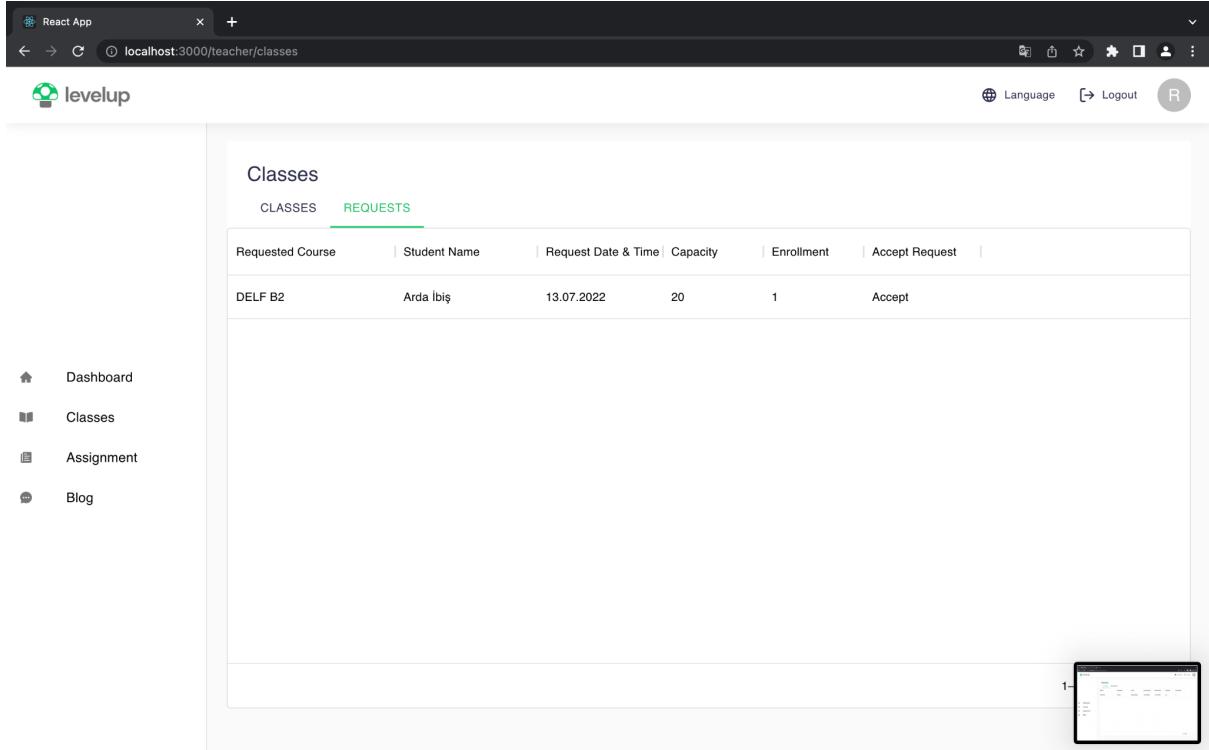
● Teacher View



The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/teacher/classes". The page has a header with a "levelup" logo, language selection, and logout options. On the left is a sidebar with links: Dashboard, Classes, Assignment, and Blog. The main content area is titled "Classes" and contains a table of classes. There are two rows: "DELF B1" (French, Intermediate, 12.05.2022 - 12.07.2022, capacity 20, enrollment 1) and "DELF B2" (French, Intermediate, 13.07.2022 - 13.09.2022, capacity 20, enrollment 1). At the bottom right of the table are buttons for "1–2 of 2".

Name	Language	Level	Course Start	Course End	Capacity	Enrollment
DELF B1	French	Intermediate	12.05.2022	12.07.2022	20	1
DELF B2	French	Intermediate	13.07.2022	13.09.2022	20	1

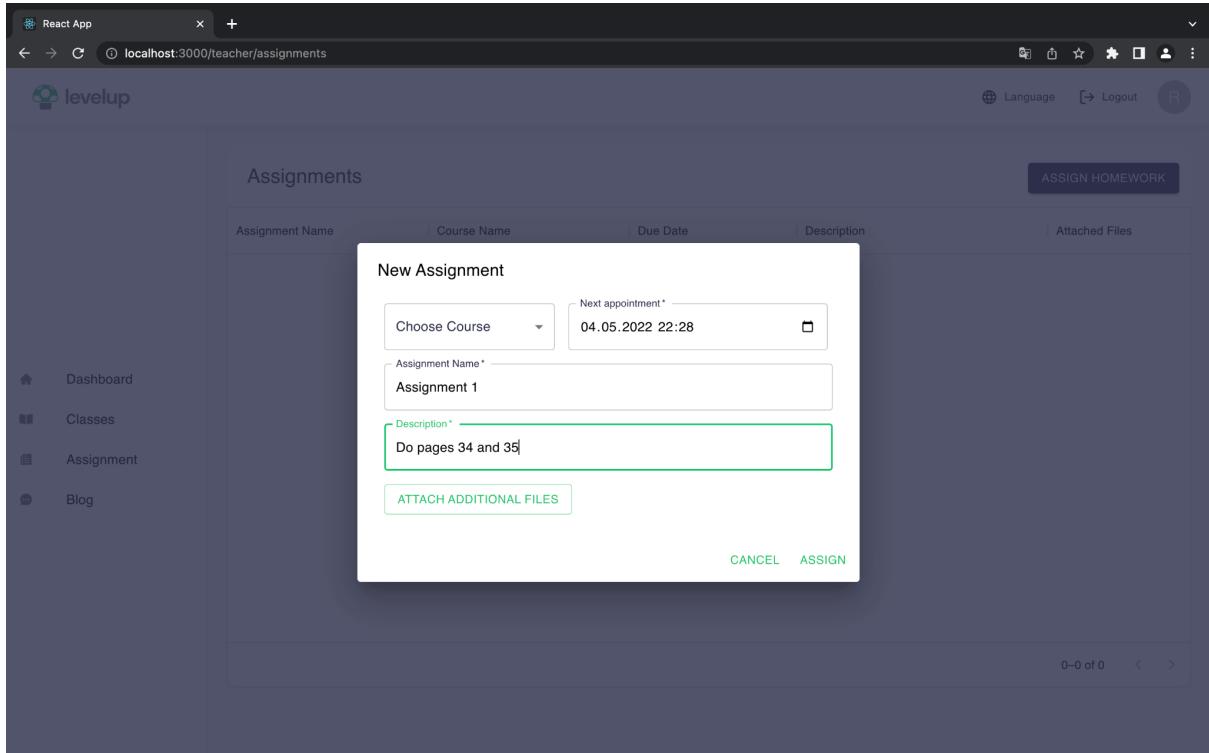
From the classes page, the teacher can see the courses currently given by them and planned for them. They can view details about the course.



The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/teacher/classes". The page has a header with a "levelup" logo, language selection, and logout options. On the left is a sidebar with links to "Dashboard", "Classes" (which is the active tab), "Assignment", and "Blog". The main content area is titled "Classes" and contains two tabs: "CLASSES" and "REQUESTS" (which is highlighted). A table lists a single request:

Requested Course	Student Name	Request Date & Time	Capacity	Enrollment	Accept Request
DELF B2	Arda İbiş	13.07.2022	20	1	Accept

By switching to the requests tab, teachers can see the requests made for their classes.



The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/teacher/assignments". The page has a header with a "levelup" logo, language selection, and logout options. On the left is a sidebar with links to "Dashboard", "Classes", "Assignment" (which is the active tab), and "Blog". The main content area is titled "Assignments" and includes an "ASSIGN HOMEWORK" button. A modal dialog is open, titled "New Assignment", containing fields for "Choose Course" (set to "04.05.2022 22:28"), "Assignment Name" ("Assignment 1"), and "Description" ("Do pages 34 and 35"). At the bottom of the modal are "CANCEL" and "ASSIGN" buttons.

From the assign homework button, teachers can create new assignments for a specific course. They can also add files to assignments.

● Language Native View

The screenshot shows a web browser window with the URL `localhost:3000/native/exercises`. The page has a header with a React App logo, a search bar, and a user menu with Language, Logout, and a profile icon. On the left is a sidebar with links for Dashboard, Exercises, Requests, and Blog. The main content area is titled "Exercises" and contains two dropdown filters: "Graded" and "Upcoming". Below the filters is a table with columns: Student Name, Exercise Date, Exercise Link, Grade, and Grade Student. Two rows are listed: Mert Şen (12.05.2022) and Yağmur Türkoğlu (12.05.2022). Each row has a "Go to exercise" link and a "Grade" button. At the bottom right of the table is a pagination indicator "1-2 of 2" with navigation arrows.

Student Name	Exercise Date	Exercise Link	Grade	Grade Student
Mert Şen	12.05.2022	Go to exercise	-	Grade
Yağmur Türkoğlu	12.05.2022	Go to exercise	-	Grade

From the Exercises page, the native can see their scheduled exercises and they filter by dropdown menus above the page.

The screenshot shows a web browser window with the URL `localhost:3000/native/requests`. The layout is identical to the Exercises page, with a header, sidebar, and main content area. The main content is titled "Requests" and contains a table with columns: Student Name, Exercise Date & Time, Notes, Accept, and Reject. Two rows are listed: Mert Şen (12.05.2022, notes "I want to exercise on my pronunciations") and Yağmur Türkoğlu (16.05.2022, notes "I want to practice for my exam"). Each row has an "Accept" button and a "Reject" button. A pagination indicator "1-2 of 2" is at the bottom right.

Student Name	Exercise Date & Time	Notes	Accept	Reject
Mert Şen	12.05.2022	I want to exercise on my pronunciations	Accept	Reject
Yağmur Türkoğlu	16.05.2022	I want to practice for my exam	Accept	Reject

From the Requests page the native can see speaking exercise requests made for them and accept or reject.

- Admin View

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/admin". The main content is a "System Report for 07.04.2022" card. At the top of the card are two dropdown menus: "Choose Language" set to "French" and "Choose Level" set to "Intermediate", followed by a green "CREATE REPORT" button. Below these are three data cards:

- Number of students learning French: **143**
- Speaking exercise average of French students: **87.72**
- Homework average of intermediate students: **84.96**

On the left side of the dashboard, there is a sidebar with two items: "Dashboard" and "Classes".

System Admin can create reports by specifying the language and the level.