**Створення застосування**

ПРЕДМЕТНА ОБЛАСТЬ:

ТЕМА: 1. Середовище дистанційного навчання

Викладачі використовують систему дистанційного навчання (далі – СДН) для надання доступу через Інтернет до навчальних матеріалів, як-от: конспекти лекцій, інструкції по лабораторних роботах, перелік тем курсових проектів і т.ін., а також для оцінювання знань студентів за допомогою тестів та контрольних робіт.

Особа, що користується СДН, повинна бути зареєстрована як користувач, вона крім особистих даних має свій логін і пароль. Адміністратор СДН реєструє користувачів, створює навчальні дисципліни і призначає для дисциплін викладачів з множини користувачів. У дисципліни може бути кілька викладачів.

Викладач для своїх дисциплін створює групи і залучає у групи студентів з числа користувачів. Група існує лише для дисципліни.

Оцінки студентам виставляє викладач або СДН автоматично. Викладач оцінює виконання завдань (зокрема, лабораторних робіт) і контрольних робіт. СДН автоматично оцінює виконання студентами тестів.

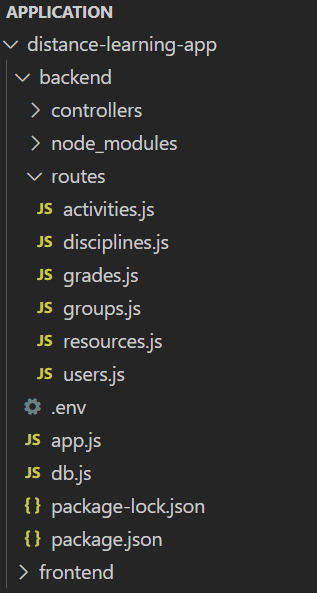
Елементи дисципліни, які надаються студентам для вивчення як готовий матеріал, називаються ресурси. Елементи дисципліни, які потребують дій студента по демонстрації своїх знань, називаються *діяльності*.

Звітні форми, які цікавлять користувачів:

* Звіт щодо діяльності за останній час (реєстрація студентів, зміна ресурсів та діяльностей викладачем, виконання діяльностей студентами).
* Поточна успішність студентів: викладачу надаються всі оцінки, а студенту – його власні.

**Структура проекту:**

Back-end реалізований на основі Node.js з моделлю MVC (не повна, бо контролери на задіювали) фреймворком Express:



Вміст файлу app.js, що відповідає за запуск back-end сервера:

const express = require('express');

const cors = require('cors');

const disciplinesRouter = require('./routes/disciplines');

const usersRouter = require('./routes/users');

const groupsRouter = require('./routes/groups');

const activitiesRouter = require('./routes/activities');

const resourcesRouter = require('./routes/resources');

const gradesRouter = require('./routes/grades');

const app = express();

app.use(cors());

app.use(express.json());

app.use('/api/disciplines', disciplinesRouter);

app.use('/api/users', usersRouter);

app.use('/api/groups', groupsRouter);

app.use('/api/activities', activitiesRouter);

app.use('/api/resources', resourcesRouter);

app.use('/api/grades', gradesRouter);

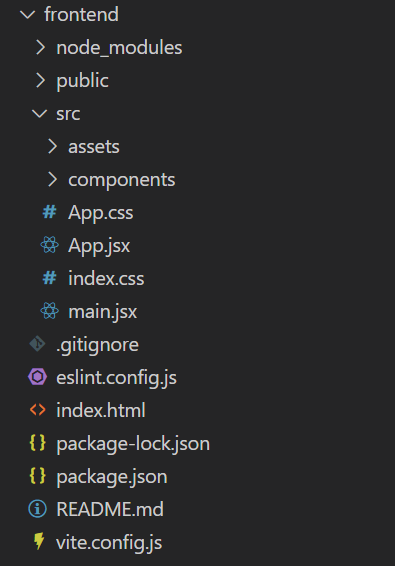
const PORT = 3001;

app.listen(PORT, () => {

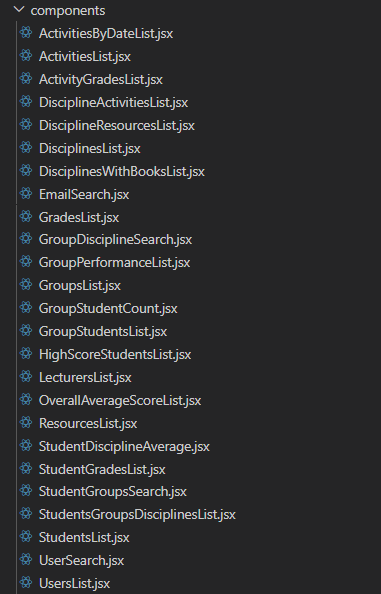
  console.log(`Server running on port ${PORT}`);

});

Front-end реалізований на основі React.js, різні компоненти відповідали за певні запити.



Компоненти:



Вміст файлу App.jsx, що відповідає за запуск React сервера:

import React from 'react';

import './App.css';

import DisciplinesList from './components/DisciplinesList';

import UsersList from './components/UsersList';

import GroupsList from './components/GroupsList';

import ActivitiesList from './components/ActivitiesList';

import ResourcesList from './components/ResourcesList';

import GradesList from './components/GradesList';

import StudentsList from './components/StudentsList';

import LecturersList from './components/LecturersList';

import UserSearch from './components/UserSearch';

import EmailSearch from './components/EmailSearch';

import GroupStudentsList from './components/GroupStudentsList';

import StudentGroupsSearch from './components/StudentGroupsSearch';

import GroupStudentCount from './components/GroupStudentCount';

import DisciplineActivitiesList from './components/DisciplineActivitiesList';

import ActivitiesByDateList from './components/ActivitiesByDateList';

import GroupDisciplineSearch from './components/GroupDisciplineSearch';

import StudentGradesList from './components/StudentGradesList';

import ActivityGradesList from './components/ActivityGradesList';

import StudentDisciplineAverage from './components/StudentDisciplineAverage';

import HighScoreStudentsList from './components/HighScoreStudentsList';

import GroupPerformanceList from './components/GroupPerformanceList';

import DisciplineResourcesList from './components/DisciplineResourcesList';

import DisciplinesWithBooksList from './components/DisciplinesWithBooksList';

import StudentsGroupsDisciplinesList from './components/StudentsGroupsDisciplinesList';

import OverallAverageScoreList from './components/OverallAverageScoreList';

function App() {

  return (

    <div className="container">

      <h1>Система дистанційного навчання</h1>

      <p>Вітаємо у вашій системі дистанційного навчання!</p>

      <UserSearch />

      <EmailSearch />

      <GroupStudentsList />

      <StudentGroupsSearch />

      <GroupStudentCount />

      <DisciplineActivitiesList />

      <ActivitiesByDateList />

      <GroupDisciplineSearch />

      <StudentGradesList />

      <ActivityGradesList />

      <StudentDisciplineAverage />

      <HighScoreStudentsList />

      <GroupPerformanceList />

      <DisciplineResourcesList />

      <DisciplinesWithBooksList />

      <StudentsGroupsDisciplinesList />

      <OverallAverageScoreList />

      <DisciplinesList />

      <UsersList />

      <StudentsList />

      <LecturersList />

      <GroupsList />

      <ActivitiesList />

      <ResourcesList />

      <GradesList />

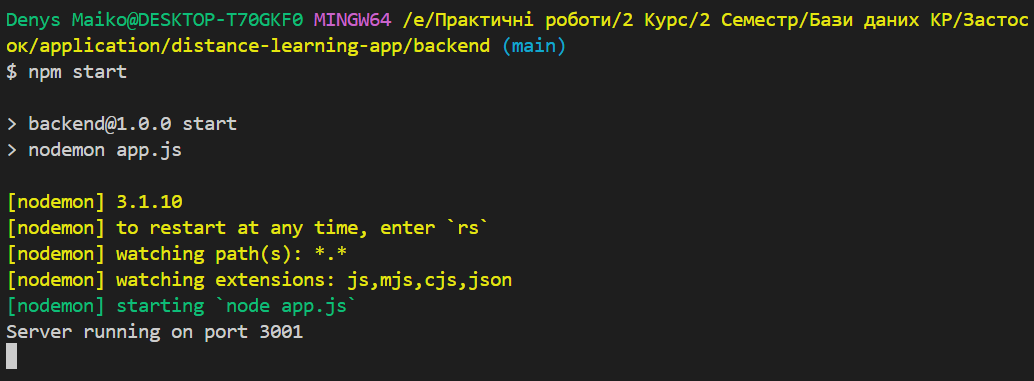
    </div>

  );

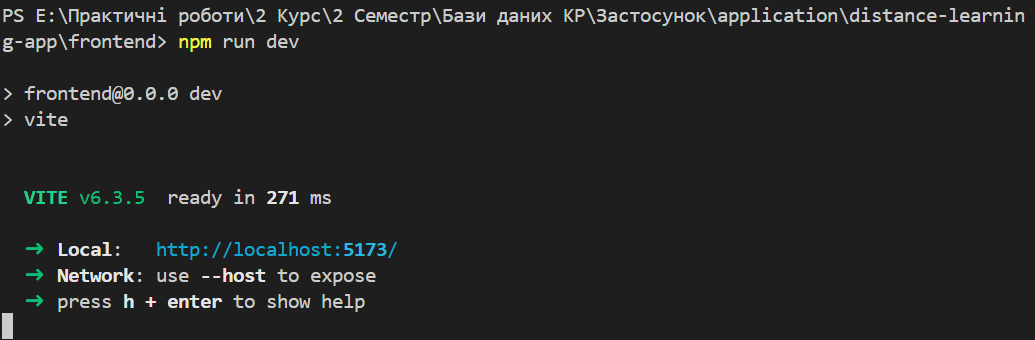
}

export default App;

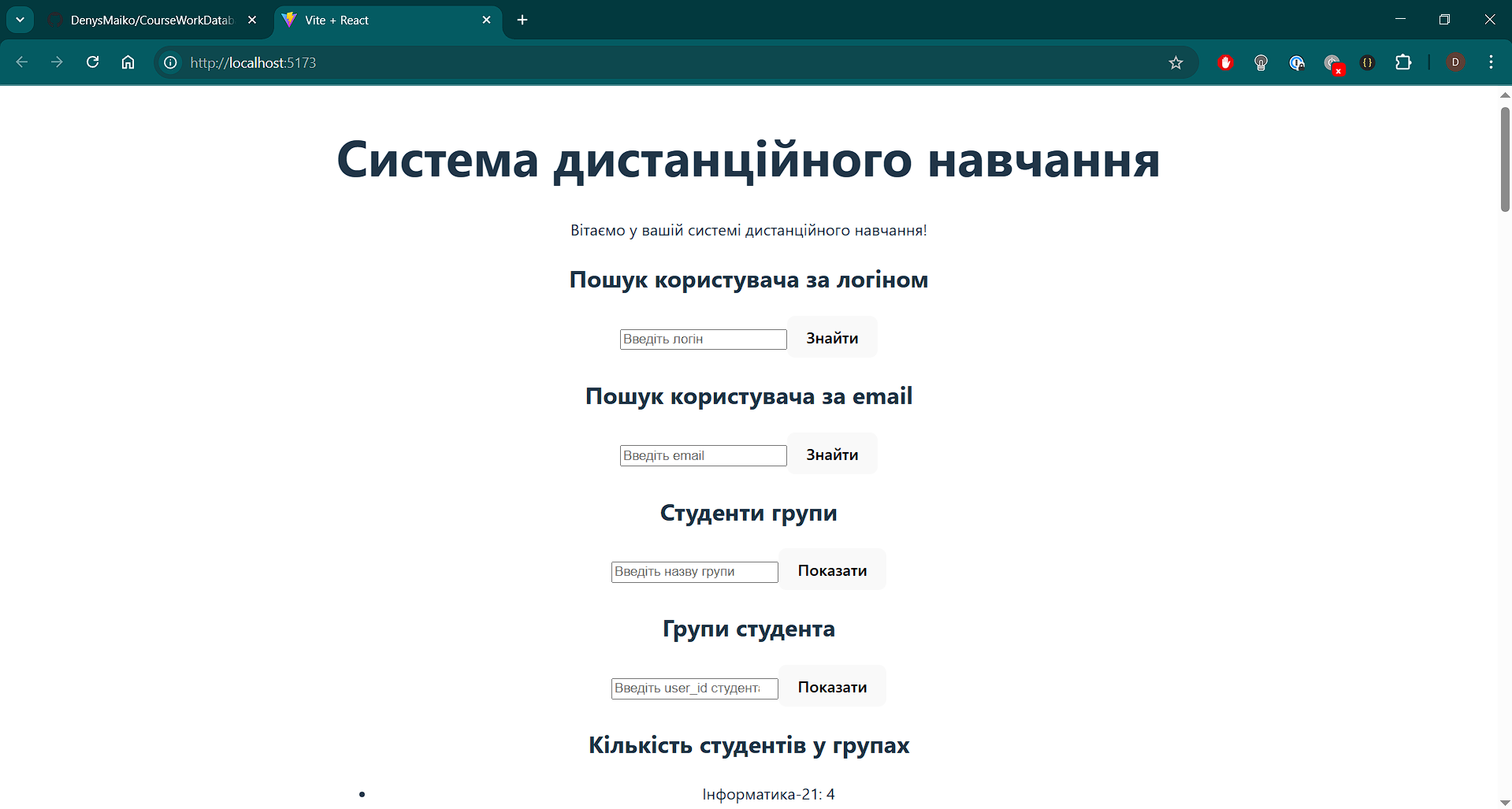
Запустимо сервер back-end за допомогою команди `npm start`:



Запустимо сервер front-end за допомогою команди `npm run dev`:



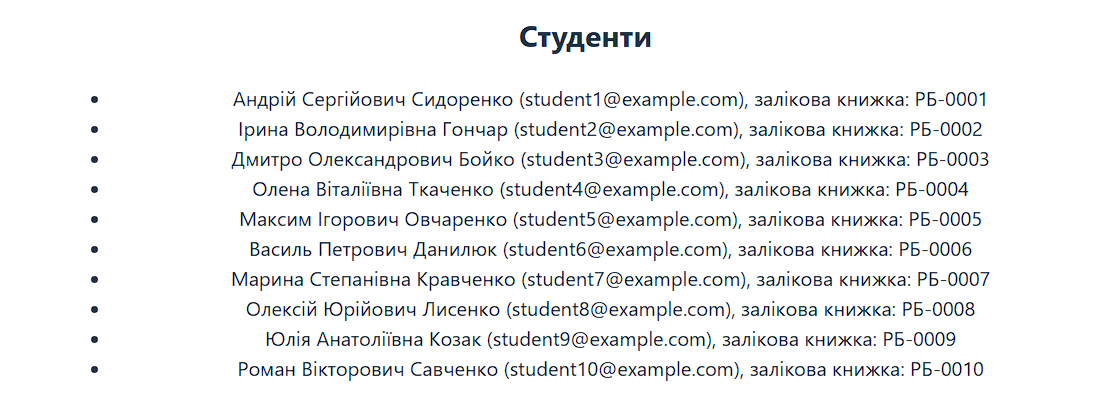
За URL: http://localhost:5173/ потрапляємо на сайт, де реалізовано користувацьку взаємодію із системою дистанційного навчання:



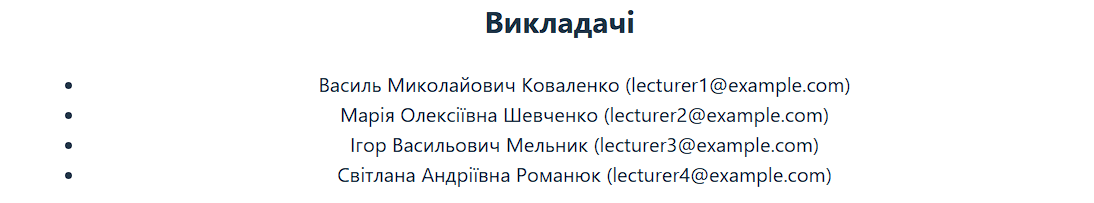
**API:**

Запити для отримання інформації про користувачів:

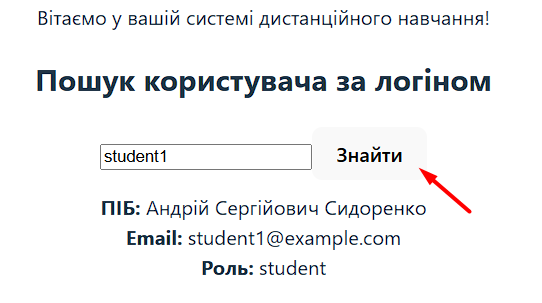
Отримати всіх студентів:



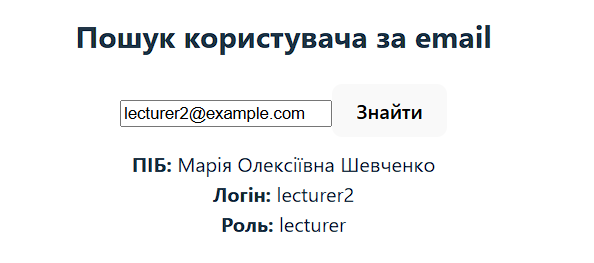
Отримати всіх викладачів:



Знайти користувача за логіном: (у нашому випадку з логіном student1)

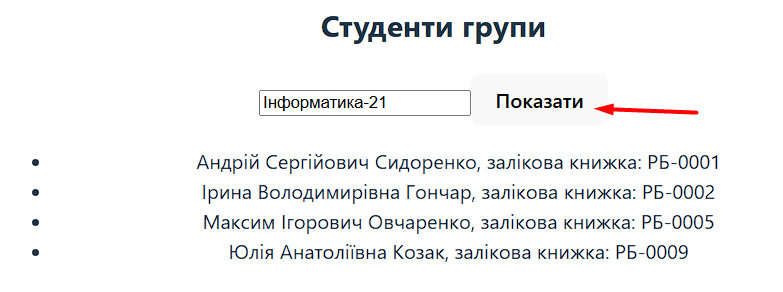


Знайти користувача за email:

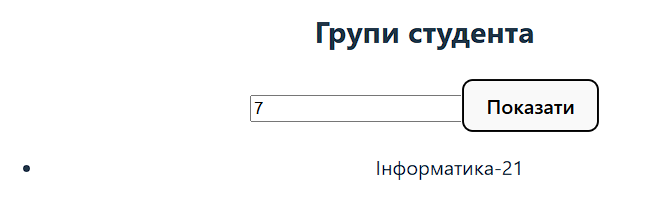


Запити для роботи з групами та студентами:

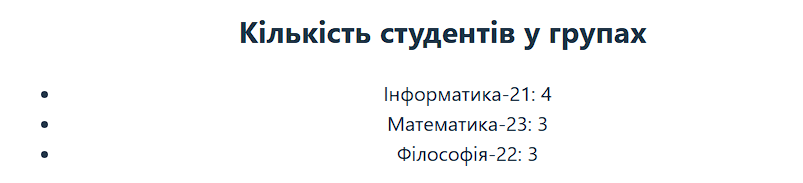
Отримати список студентів у певній групі (наприклад, група 'Інформатика-21'):



Знайти, до якої групи належить студент (наприклад, студент з user\_id = 7):

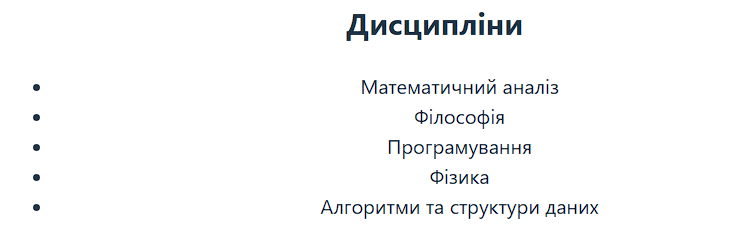


Порахувати кількість студентів у кожній групі:

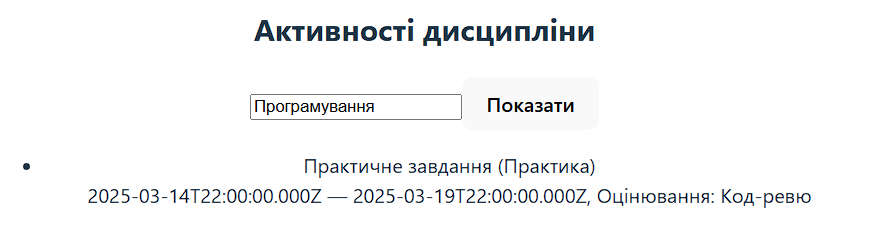


Запити для роботи з дисциплінами та активностями:

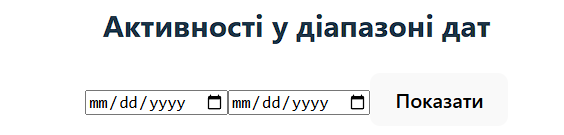
Отримати всі дисципліни:

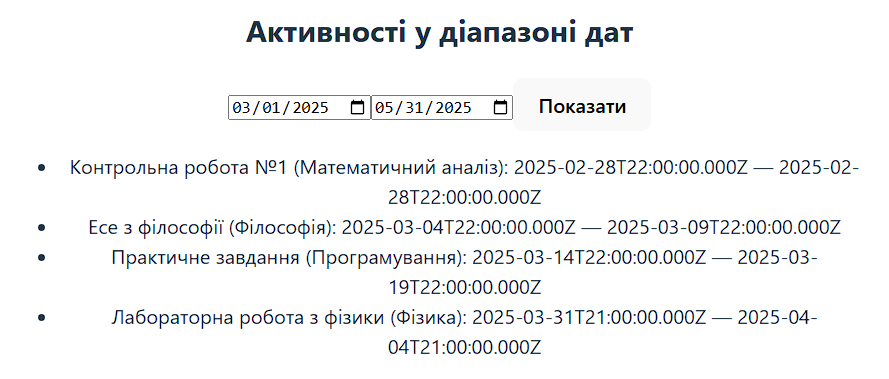


Отримати всі активності для певної дисципліни (наприклад, 'Програмування'):

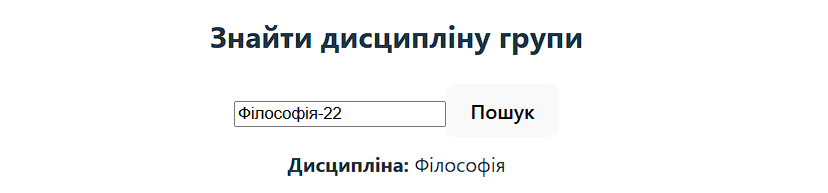


Отримати активності, що відбуваються у певному діапазоні дат (у нашому випадку між 2025-03-01 та 2025-03-31:



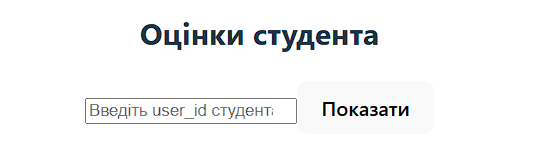


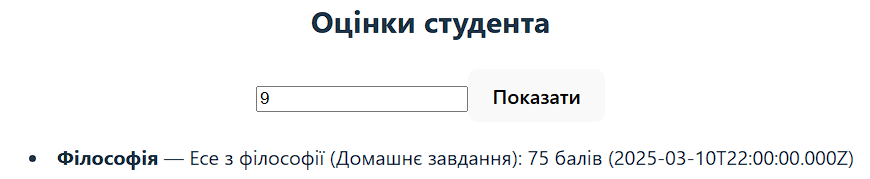
Знайти дисципліну, до якої належить певна група (наприклад, 'Філософія-22'):



Запити для роботи з оцінками:

Отримати всі оцінки конкретного студента (наприклад, студента з user\_id = 9):

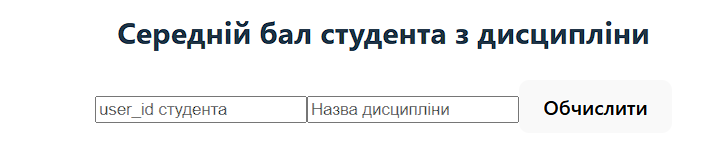


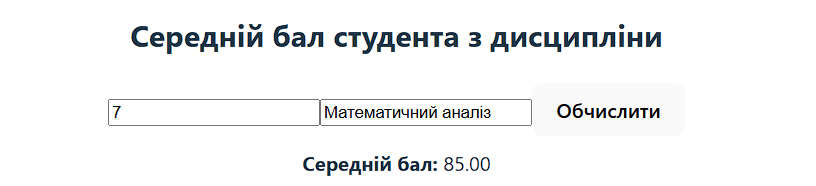


Отримати всі оцінки для конкретної активності (наприклад, 'Контрольна робота №1'):

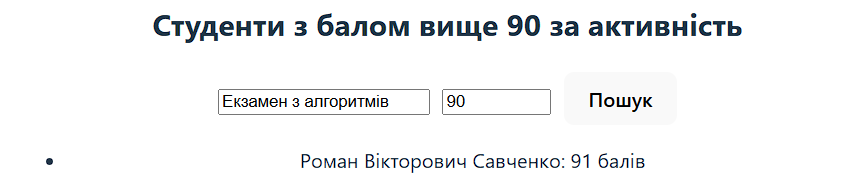


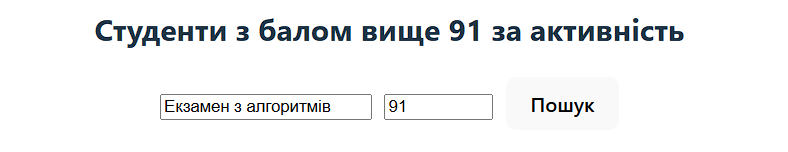
Обчислити середній бал студента з певної дисципліни:





Знайти студентів, які отримали бал вище 90 за певну активність:



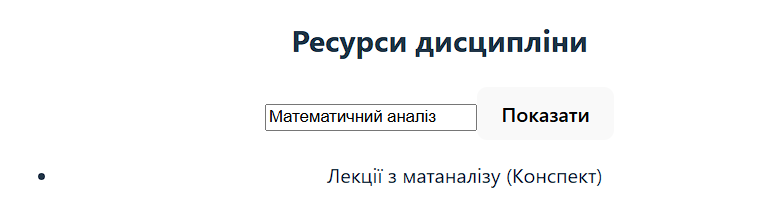


Отримати поточну успішність усіх студентів групи (наприклад, 'Інформатика-21'):

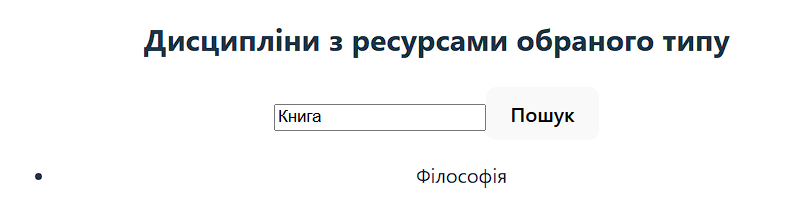


Запити для роботи з ресурсами:

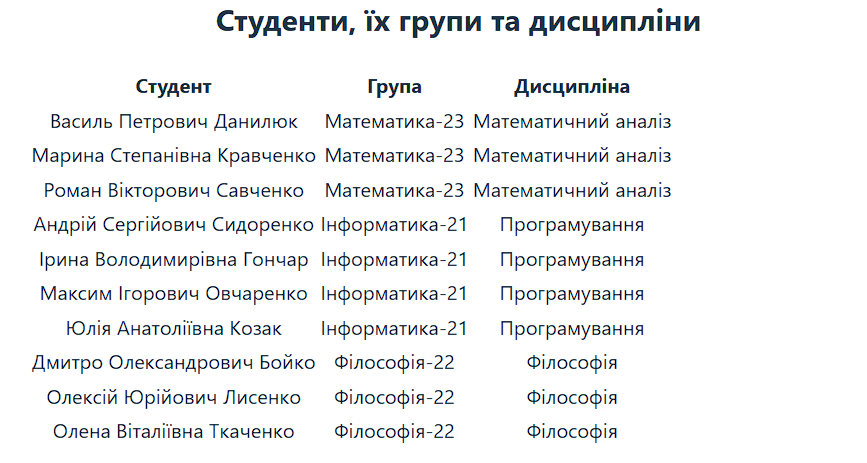
Отримати всі ресурси для певної дисципліни (наприклад, 'Математичний аналіз'):



Знайти всі дисципліни, що мають ресурси вказаного типу (у нашому випадку 'Книга'):



Вивести список студентів, їх групи та дисципліни цих груп:



Вивести середній бал кожного студента по всіх активностях:

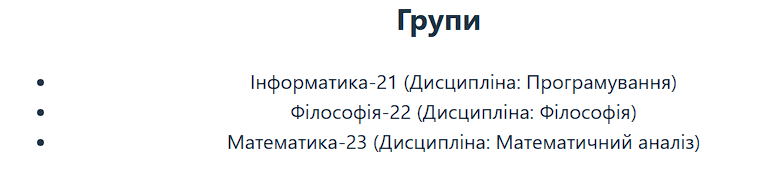


Переліки

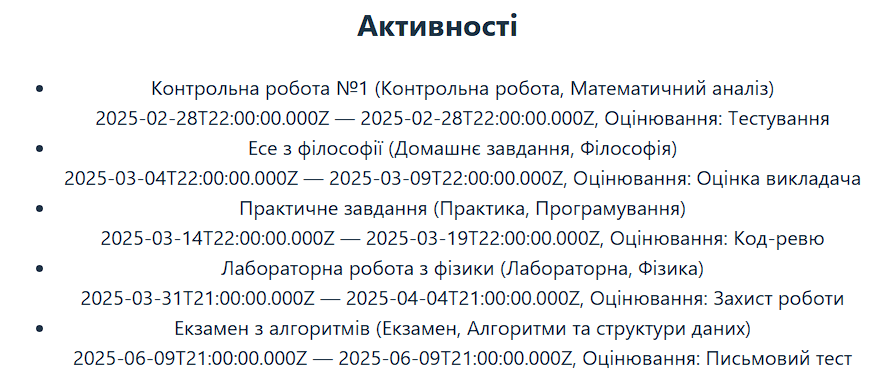
Вивести всіх користувачів:



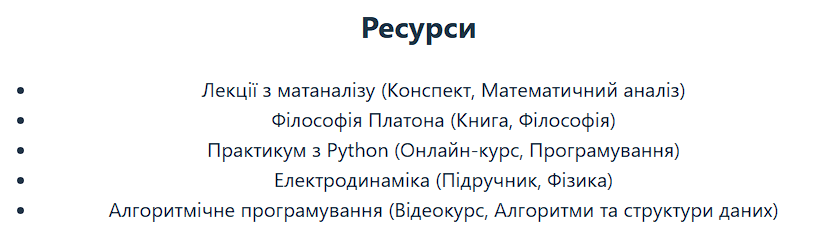
Вивести всі групи:



Вивести всі активності:



Вивести всі ресурси:



Вивести всі оцінки:

