

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мережних технологій факультету інформатики

Розробка застосунку з побудови розкладів з використанням бібліотеки Yii.

Текстова частина до курсової роботи за спеціальністю «Інженерія
програмного забезпечення» 121

Керівник курсової роботи

Кобзар О.О.

(підпис)

“ ____ ” _____ 2021 р.

Виконав студент

Гордін М.Е.

“ ____ ” _____ 2021 р.

Київ 2021

Тема: Розробка застосунку з побудови розкладів з використанням бібліотеки Yii.

№	Назва етапу	Термін виконання	Примітка
1.	Отримання теми курсової роботи	10.09.2020	
2.	Вивчення предметної області	10.10.2020	
3.	Огляд існуючих рішень	10.11.2020	
4.	Огляд інструментів розробки	01.01.2021	
5.	Написання технічного завдання	01.02.2021	
6.	Перша консультація з керівником	02.02.2021	
7.	Написання курсової роботи	01.04.2021	
8.	Друга консультація з керівником	01.05.2021	
9.	Створення презентації курсової роботи	17.05.2021	

Студент Гордін М.Е.

Керівник Кобзар О.О.

ЗМІСТ

Вступ	5
Розділ 1. Огляд існуючих рішень	7
Опис функцій існуючих рішень.....	7
1.1.1 Google Календар.....	7
1.1.2 Google Таблиці, Microsoft Excel.....	8
1.1.3 KMAScheduler	9
Розділ 2. Огляд використаних технологій та інструментів	11
2.2 Active Record.....	12
2.3 Генератор базового коду Gii	13
2.4 Yii Migrations	14
Розділ 3. Програмна реалізація застосунку	14
3.1 Технічне завдання	14
3.1.1 Основні вимоги.....	14
3.1.1.1 Опис функціоналу сторінок застосунку.....	14
3.3.1.2 Опис моделей.....	16
3.3.2 Заплановані доопрацювання	17
3.3.3 Можливі покращення.....	18
3.4 Розробка системи	18
3.4.1 Налаштування.....	18
3.4.1.1 Встановлення базового Yii проекту	18
3.4.1.2 Конфігурування проекту	18
3.4.1.3 Підключення додаткових бібліотек.....	18
3.4.3 Робота з Gii	20
3.4.3.1 Генерування моделей ActiveRecord.....	20
3.4.3.1.1 Генерування моделі викладача	20
3.4.3.1.2 Генерування моделі поста викладача.....	21
3.4.3.1.3 Генерування моделі типу групи.....	21

3.4.3.1.4 Генерування моделі курсу.....	22
3.4.3.1.5 Генерування моделі групи.....	22
3.4.3.1.6 Генерування моделі аудиторії.....	23
3.4.3.1.7 Генерування моделі робочого дня.....	23
3.4.3.1.8 Генерування моделі робочого часу	24
3.4.3.1.9 Генерування моделі студента.....	24
3.4.3.1.10 Генерування моделі, що відповідає за зв'язок студента і групи.....	25
3.4.3.1.11 Генерування моделі, що відповідає за запис групи до розкладу	26
3.4.4 Генерування файлів контролерів та представлень для базових операцій для взаємодії з записами.....	27
3.4.5 Налаштування головної сторінки	29
3.4.6 Створення системи конструювання розкладу	30
3.4.6.1 Перегляд розкладу.....	30
3.4.6.2 Створення запису	32
3.4.6.3 Видалення записів.....	33
3.4.6.4 Перегляд індивідуального розкладу студента.....	34
3.4.6.5 Перегляд розкладу викладача	35
3.4.6.6 Інструмент запису студентів у групи	35
3.5 Загальна інформація про використані функції.....	35
Висновок	37

Вступ

Початок нового навчального триместру завжди супроводжується проблемами з генеруванням та переглядом розкладу навчальних дисциплін. Учні доводиться відфільтровувати власні групи від загальних з розкладу. Окрім цього, студентам, які записані на предмети, що належать іншим факультетам доводиться збирати інформацію про записи з різних файлів розширення Excel. Але такий спосіб збору даних не дає можливості переглянути розклад на певний тиждень. Тому студенту доводиться переносити усі записи до власного файлу, або зберігати усе у пам'яті. У свою чергу асистент або методист повинен власноруч заповнювати загальний розклад переписуючи назви предметів, номери груп та імена викладачів. Але такий спосіб заповнення даних не дає можливості продивитися список записаних студентів.

Система побудови та перегляду розкладу допоможе як студентам, так і методистам зберігати час при побудові та перегляді розкладу відповідно. Користувач повинен мати можливість записати групу студентів з певної дисципліни на час, передивитися загальний розклад на предмети, у якому буде вказано ім'я викладача, назва предмета, день, час, номер та тип групи, тижні, на яких цей предмет буде проходити, аудиторію та список студентів. Окрім цього, додаток має реалізовувати можливість додавання, перегляду, редагування учня, вчителя, предмета, часу, дня, групи, аудиторію та запису студента на дисципліну. Також важливо мати можливість переглянути розклад на певний тиждень.

Yii – високопродуктивна Model-View-Controller веб-бібліотека написана на мові PHP з відкритим кодом, яка надає можливість будувати складні застосунки, а саме забезпечує можливість використання наступних можливостей: відладка, міграції, кешування окремих фрагментів, генерування базового PHP коду, візуалізація на стороні серверу.

Об'єкт дослідження: бібліотека Yii. Мета курсової роботи – засвоєння нових навичок та можливостей під час роботи з Yii та інструментами: Active Record, Widgets, PHP в якості шаблонізатору, Bootstrap шляхом створення додатку з розбудови розкладу.

Розділ 1. Огляд існуючих рішень

Опис функцій існуючих рішень

1.1.1 Google Календар

Студенти використовують Google Календар для фіксування подій, які відбуваються регулярно.

Це зручний додаток, який дозволяє регулювати та керувати планами на майбутнє. Система безкоштовна та зрозуміла, надає можливість записати подію, зробити її регулярною та повідомити користувача про його плани заздалегідь. Також певна організація, якщо користувач до неї належить може власноруч додати подію до індивідуального розкладу. Інтерфейс Google Календаря інтуїтивно зрозумілий.

Додаток надає наступні можливості:

- перегляд індивідуального розкладу
- запис разової події
- запис регулярної події
- фільтрування календарів
- додавання інших користувачів до події
- інтеграція з Google Meet

Всі можливі поля, що доступні для події (рисунок 1.1.1.1):

- назва
- тип (подія або задача)
- дата та час (часовий пояс, регулярність події)
- користувачі, що належать до цієї події
- наявність Google Meet для цієї події
- місце знаходження
- опис та файли
- календар, до якого належить подія
- статус користувача для цієї події (зайнятий або вільний)

- дозволені дії для гостей

Рисунок 0-1.1.1.1 Поля, що доступні для події

1.1.2 Google Таблиці, Microsoft Excel

Google Таблиці та Microsoft Excel дозволяють користувачу створювати таблиці та заповнювати їх даними. Це не додатки для побудови розкладів, але студенти досі використовують ці системи для зберігання інформації про записи. Асистенти та методисти власноруч заповнюють інформацію про кожну дисципліну, викладача, групу, аудиторію та час, що припускає наявність помилок у розкладі. У свою чергу, студентам доводиться обирати свої групи серед усіх наявних у файлі, на що витрачається певна кількість часу і вважається незручним рішенням (рисунок 1.1.2.1).

Google Таблиці – безкоштовний додаток для власників акаунту Google. Файли зроблені у системі зберігаються на Google Drive, але можуть бути завантажені на диск. Платформа представляє собою інтернет сторінку.

Microsoft Excel – платне програмне забезпечення. Файли зберігаються локально, але Microsoft Excel надає можливість працювати online.

Google Табиці та Microsoft Excel використовують як програми, що надають робочий простір для заповнення та зберігання даних про зв'язок часу та групи, що належать йому.

Перегляд

Факультет інформатики						
Спеціальність "Інженерія програмного забезпечення", 3 р.н.						
Розклад занять на додатковий період весняного семестру 2020-2021 н.р.						
6						
7						
8						
9						
10	День	Час	Дисципліна, викладач	Група	Тижні	Аудиторія
11	Понеділок	8:30-9:50				
12		10:00-11:30				
13		11:40-13:00				
14		13:30-14:50				
15		15:00-16:20				
16		16:30-17:50				
17		18:00-19:20				
18	Вівторок	8:30-9:50	Основи Unix систем, ас. А.І. Чередарчук	лекція	1 - 6	Дистанційно
19		10:00-11:20	Основи операційних систем, доц. О.В. Франчук	лекція	1 - 6	Дистанційно
20		11:40-13:00	Основи операційних систем, доц. О.В. Франчук	1	1 - 6	Дистанційно
21		13:30-14:50	Основи операційних систем, доц. О.В. Франчук	2	1 - 6	Дистанційно
22		15:00-16:20				
23		16:30-17:50	Основи Unix систем, ас. А.І. Чередарчук	1	1 - 6	Дистанційно
24		18:00-19:20				

Рисунок 1.1.2.1 Перегляд розкладу у застосунку Microsoft Excel

1.1.3 KMAScheduler

KMAScheduler – бот у соціальній мережі Telegram, який розробили та підтримують учні Національного Університету «Києво-Могилянська академія». Студенти використовують його для перегляду розкладу. Дізнавшись у розробників я вияснив, що додаток перебирає Excel файли, що роблять асистенти та після цього надає можливість студентам переглянути розклад на дисципліни до яких вони належать.

Запис у групі відбувається шляхом введення команди «/manage» у поле для повідомлення та вибором факультету, курсу, спеціальності, викладача та номеру групи (рисунок 1.1.3.1).

Переглянути власний розклад можна командами «/now», «/next», «today», «/tomorrow», «/week [n]», «/date [Число].[Місяць]» у поле для повідомлень, після чого система відповість розкладом на поточну пару, наступну пару, сьогодні, завтра, певний тиждень та певну дату відповідно.

Також додаток надає можливість підтримати розробників, змінити вигляд відображення розкладу, переглянути загальний розклад та записи на певний день тижня.

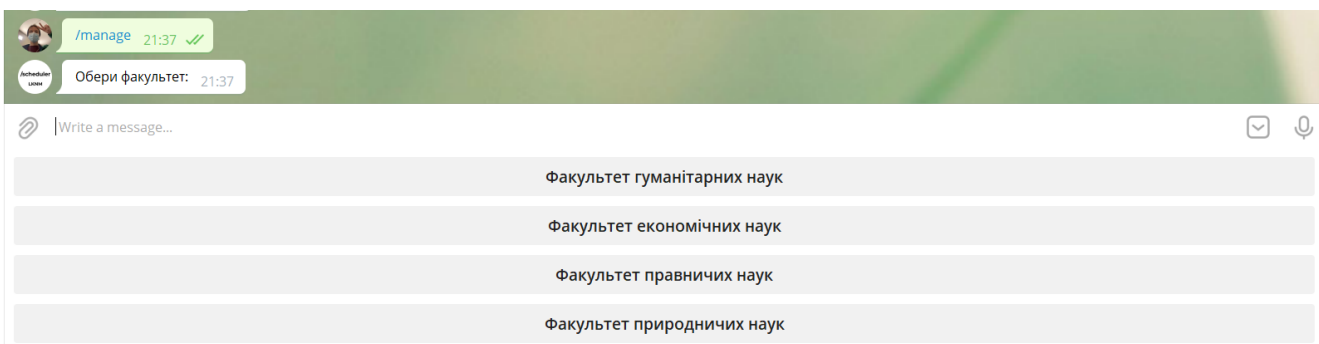


Рисунок 1.1.3.1 Запис у KMAScheduler

1.1 Аналіз функцій існуючих рішень та визначення доцільності розробки системи

Розглянуті рішення потребують певного часу та даних, які слід ввести користувачу. Асистентам та методистам доводиться уважно заповнювати Excel файли інформацією про наявні записи, а саме: назву дня тижня, час, дисципліну, викладача, номер групи, аудиторію, тижні. Важливим кроком при заповненні розкладу є перевірка зайнятості аудиторії на певному тижні у певний час, для запобігання конфлікту накладання дисциплін. Також важливо надати можливість вибору існуючої групи, для того, щоб методистам та асистентам не треба було записувати назву дисципліни, номер групи та прізвище і ім'я викладача з клавіатури.

В результаті аналізу описаних систем було виявлено, що жодна з них не дає можливості автоматизувати процес створення та перегляду розкладу. Кожна з платформ потребує зусиль та часу для коректного відображення інформації для студента.

Завданням розробленої системи є забезпечення можливості конструювання розкладу з існуючих даних з перевіркою звільненості певної аудиторії у певний час, додавання, редагування, читання нових та існуючих дисциплін, днів тижня, часу пари, груп, викладачів, аудиторій, студентів для методистів та асистентів. В свою чергу студент повинен мати можливість передивитися свій загальний розклад та записи на певний тиждень. Окрім цього, має бути реалізована система запису студента до групи з функцією збереження цієї інформації. Важливо додати можливість перегляду записів для певного викладача.

Розділ 2. Огляд використаних технологій та інструментів

2.1 Yii

Yii – відома бібліотека, яка є у списку інструментів більшості PHP розробників, побудована на основі шаблону Model-View-Controller. Проект розроблюється 13 років, було випущено 2 покоління. Бібліотека – ідеальний вибір для розробки складних проектів. Універсальність бібліотеки дозволяє будувати веб-застосунки різних типів. Yii спонукає підтримувати структуру Model-View-Controller та писати простий та елегантний код. Слід зазначити, що це full-stack бібліотека і дозволяє розроблювати як серверну частину, так і клієнтську. Yii використовує мову PHP версії 5.4.0 і вище.

Бібліотека має зручний вбудований інструмент відладки. Нажаль, мова PHP такого інструмента не має. Під час відладки користувач Yii може передивитися конфігурацію проекту, встановлені додатки, налаштування PHP на машині. Також у інструменті відображується інформація про перенаправлення сторінок, запити, записи у сесії, вивід, запити до бази даних, події, підключені сторонні бібліотеки, пошту та користувача.

Модель у Yii – внутрішня структура даних застосунку. Є певні правила, які повинні виконуватись моделлю.

- модель не повинна використовувати будь-які дані, отримані з запиту напряду
- модель не повинна генерувати код для представлення (View)
- модель повинна містити у собі бізнес-логіку (правила перевірки)

Моделі можуть використовувати для генерування та виконання запитів до бази даних, перевірки коректності введених даних.

Представлення у Yii відповідає за відображення даних з моделі у правильному форматі. Є певні правила, які повинні виконуватись представленням:

- представлення не повинно напряду звертатися до бази даних
- представлення не повинно використовувати будь-які дані, отримані з запиту напряду
- представлення повинно відображенням дані за допомогою HTML та PHP скриптів

До представлень можна під'єднати клієнтські скрити та файли каскаду стилів. Також, файл представлення може містити частковий шаблон.

Контролер у Yii зв'язує моделі, представлення та інші допоміжні компоненти у додаток. Саме у цій частині застосунку оброблюються дані з запиту. У контролері створюються екземпляри моделей та керується логіка перенаправлень. Зазвичай, у добре спроектованому проєкті контролер не буде містити багато коду і те, що можливо перенести буде реалізовано у моделі, але для Yii написання «товстих» контролерів – нормально.

2.2 Active Record

Active Record – потужний інтерфейс для перегляду, редагування, додавання та видалення інформації з бази даних. Для використання цього

класу треба, щоб модель унаслідувала його. Active Record дозволяє не писати SQL запити власноруч, шляхом виклику функцій класу.

Підключення до бази даних відбувається шляхом налаштування компоненту db у папці config проекту.

Налаштування назви таблиці встановлюється шляхом вказання назви таблиці у функції tableName(), яка повертає назву таблиці, або, якщо цього не було зроблено – назвою класу у множині.

Отримання даних відбувається шляхом виклику функції find. Слід зазначити, що об'єкт можна отримати у вигляді масиву. Такий крок збереже пам'ять застосунку, але може заборонити використання певних функцій складних викликів. Запит може бути уточнений за допомогою функцій where, select, having і тд. Типовий запит на отримання даних виглядає так:

```
Students::find()->where(['id' => 1])->limit(1)->one();
```

Цей запит поверне одного студента з id – 1.

Зберігання даних відбувається шляхом створення екземпляру класу, що наслідує Active Record, заповненням атрибутів і виклику функції save. Перед збереженням Active Record перевірить усі поля об'єкта валідатором, правила для якого оголошуються у функції rules моделі.

Active Record дозволяє прописувати реляції для зв'язаних таблиць і отримувати дані з них.

2.3 Генератор базового коду Gii

Gii вміє генерувати базовий PHP код для різних задач на основі створених до цього таблиць у базі даних. Генератор на відмінно справляється з наступними задачами – генерація моделей ActiveRecord, файлів перегляду, створення та редагування екземплярів класу моделей, контролерів, форм, модулів та додатків. Функціонал Gii може бути розширений за допомогою безкоштовного функціоналу, який був написаний іншими користувачами Yii.

Для генерування моделей треба вказати назву таблиці у базі даних та назву моделі, яка буде створена. Після цього буде запропоновано передивитися створений файл, або зберегти у проект.

Для генерування частини додатку, що відповідає за створення, оновлення та перегляд екземплярів класу що наслідує ActiveRecord треба вказати назву моделі, з на основі якої будуть побудовані вищезазначений функціонал, назву контролера та шлях до представлень, що будуть створені.

Окрім цього Gii залишить коментарі у кожному з згенерованих файлів.

2.4 Yii Migrations

Для коректної роботи з базами даних слід використовувати міграції. Цей інструмент дозволить ввести зміни до поточних таблиць. У разі допущення помилки завжди можна повернутися до певної міграції у минулому. Ця система дозволяє створити та змінити таблицю, розташувати індекси, налаштувати реляції та ввести інформацію про нові рядки.

Розділ 3. Програмна реалізація застосунку

3.1 Технічне завдання

3.1.1 Основні вимоги

3.1.1.1 Опис функціоналу сторінок застосунку

Функціонал на сторінці студентів:

- створення нового студента
- редагування існуючого студента
- перегляд детальної інформації про студента
- перегляд списку усіх студентів
- перегляд загального розкладу студента
- перегляд розкладу студента на певний тиждень
- запис студента до груп

Функціонал на сторінці аудиторій:

- створення нової аудиторії
- редагування існуючої аудиторії
- перегляд детальної інформації про аудиторію
- перегляд списку усіх аудиторій

Функціонал на сторінці постів викладача:

- створення нового поста викладача
- редагування існуючого поста викладача
- перегляд детальної інформації про пост викладача
- перегляд списку усіх постів викладачів

Функціонал на сторінці викладачів:

- створення нового викладача
- редагування існуючого викладача
- перегляд детальної інформації про викладача
- перегляд списку усіх викладачів
- перегляд загального розкладу викладача
- перегляд розкладу викладача на певний тиждень

Функціонал на сторінці курсів:

- створення нового курсу
- редагування існуючого курсу
- перегляд детальної інформації про курс
- перегляд списку усіх курсів

Функціонал на сторінці типів груп:

- створення нового типу групи
- редагування існуючого типу групи
- перегляд детальної інформації про тип групи
- перегляд списку усіх типів груп

Функціонал на сторінці груп:

- створення нової групи з зазначенням викладача, курсу та типу групи
- редагування існуючої групи
- перегляд детальної інформації про групу
- перегляд списку усіх груп

Функціонал на сторінці робочих днів:

- створення нового робочого дня
- редагування існуючого робочого дня
- перегляд детальної інформації про робочий день
- перегляд списку усіх робочих днів

Функціонал на сторінці робочих часів:

- створення нового робочого часу
- редагування існуючого робочого часу
- перегляд детальної інформації про робочий час
- перегляд списку усіх робочих часів

Функціонал на сторінці загального розкладу:

- перегляд загального розкладу
- видалення інформації про існуючий запис
- додавання інформації про новий запис
- перевірка зайнятості аудиторії у цей час
- перегляд усіх студентів, що записані

3.3.1.2 Опис моделей

Студент має ім'я, прізвище та унікальний ідентифікатор. Також студент можливо записаний у певні групи.

Аудиторія має корпус, номер та унікальний ідентифікатор. Використовується під час створення нового запису у розкладі.

Викладач має ім'я, прізвище, пост та унікальний ідентифікатор.
Використовується під час створення нової групи.

Пост викладача має назву та унікальний ідентифікатор.
Використовується під час створення викладача.

Курс має назву та унікальний ідентифікатор. Використовується під час створення групи. Також у групу може бути записаний студент.

Група має номер, тип, курс, викладача та унікальний ідентифікатор.
Використовується під час створення нового запису у розкладі.

Тип групи має назву та унікальний ідентифікатор. Використовується під час створення групи.

Робочий день має назву та унікальний ідентифікатор.
Використовується під час створення робочого часу.

Робочий час має назву, день та унікальний ідентифікатор.
Використовується під час створення нового запису.

Запис має час, групу, аудиторію та тиждень. При чому декілька записів у один час у одній аудиторії бути не може.

Запис до групи має студента і групу. При чому студент не може бути двічі записаним до однієї групи.

3.3.2 Заплановані доопрацювання

- розподілення ролей на студента та методиста або асистента
- експорт даних про записи у Excel
- конструювання розкладів для різних факультетів
- робота з реальними даними студентів Національного Університету «Києво-Могилянська академія»
- використання авторизації Microsoft Office 365

3.3.3 Можливі покращення

- Зміна архітектури на Rest API.
- Використання системи студентами Національного Університету «Києво-Могилянська академія»

3.4 Розробка системи

3.4.1 Налаштування

3.4.1.1 Встановлення базового Yii проекту

Для створення базового проекту бібліотеки Yii був використаний менеджер залежності composer та командний рядок. Команда, яка була викликана: `composer create-project --prefer-dist yiisoft/yii2-app-basic coursework`, після чого у директорію `coursework` було встановлено останню стабільну версію бібліотеки.

3.4.1.2 Конфігурування проекту

Після встановлення проекту важливо налаштувати загальну конфігурацію проекту та підключення до бази даних.

У файлі `config/web.php` встановлюємо значення для масиву налаштування `$config['components']['request']['cookieValidationKey']` значення, яке повинно бути збережено у секреті. Без цього проект не буде запущено.

У файлі `config/db.php` встановлюємо потрібні дані для підключення до бази даних. Серед них треба знати реляційну систему, назву бази даних, ім'я користувача та пароль.

Після чого, використавши запускаємо роботу бібліотеки шляхом виконання наступної команди у командний рядок: `php yii serve`. Ця команда забезпечить безперервну роботу Yii без потреби перезапустити додаток після зміни коду.

3.4.1.3 Підключення додаткових бібліотек

Підключення сторонніх бібліотек можливо тільки після встановлення їх за допомогою менеджера залежності composer. Після чого їх можна

підключити у файлі `assets/AppAsset.php`, додавши простір імен у масив `$depends`. У нашому випадку встановлення додаткових бібліотек не потребувалось.

3.4.2 Робота з базою даних

3.4.2.1 Створення міграцій

Шляхом створення та застосування міграцій було створено наступні таблиці:

- `audiences`
- `courses`
- `days`
- `groups`
- `groups_students`
- `group_types`
- `posts`
- `students`
- `teachers`
- `times`
- `times_groups_audiences`,

Створення таблиці відбувається шляхом виклику функції `createTable` в функції `safeUp` файла міграції. Після чого були виставлені потрібні зв'язки, обмеження та індекси шляхом викликів функцій `createIndex` та `addForeignKey`. Таким чином відбувається забезпечення коректного використання таблиць у базі даних, накладання обмежень для певних значень. Також цей крок дозволяє правильно використовувати `Gii` для генерування моделей `ActiveRecord`.

Шляхом виконання команди 'yii migrate' приводимо у дію написані раніше міграції. Після цього слід перевірити наявність усіх таблиць та даних у них шляхом виконання Select запитів.

3.4.3 Робота з Gii

3.4.3.1 Генерування моделей ActiveRecord

Для використання генератора базового PHP коду моделей слід перейти на посилання '/gii', після чого обрати вкладку Model Generator та натиснути кнопку «Start». Після чого додаток відобразить форму з певними полями. Серед них користувачу треба буде заповнити поле «Table name», що відповідає за назву таблиці у базі даних, «Model Class Name», що відповідає за назву моделі, що буде створена, та налаштувати деякі опції. Серед них: стандартизація великих букв, перенесення назви моделі з множини у однину, зміна базового класу, вибір підключення до бази даних, використання префіксів у таблицях, зміна простору імен, генерування зв'язків і шаблон коду, який буде використаний для генерації.

Також додаток пропонує функцію попереднього перегляду та збереження файлів у директорії проекту.

Розглянемо кожну модель детальніше.

3.4.3.1.1 Генерування моделі викладача

Модель викладача повинна мати наступні поля: ім'я, ідентифікатор поста, дата створення запису, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'teachers', після чого система автоматично заповнить поле 'Model Class Name' значенням Teachers. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Teachers, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі

- attributeLabels, що повертає назви полів для користувача
- getGroups, що повертає зв'язок усіх групи, що належать до цього викладача
- getPost, що повертає зв'язок одного поста, що належить до цього викладача

3.4.3.1.2 Генерування моделі поста викладача

Модель поста викладача повинна мати наступні поля: назва, дата створення запису, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'posts', після чого система автоматично заповнить поле 'Model Class Name' значенням Posts. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Posts, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getTeachers, що повертає зв'язок усіх викладачів, що належать до цього поста

3.4.3.1.3 Генерування моделі типу групи

Модель поста викладача повинна мати наступні поля: назва, дата створення запису, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'group_types', після чого система автоматично заповнить поле 'Model Class Name' значенням 'GroupTypes'. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі GroupTypes, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getGroups, що повертає зв'язок усіх груп, що належать до цього типу групи

3.4.3.1.4 Генерування моделі курсу

Модель курсу повинна мати наступні поля: назва, дата створення запису, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'courses', після чого система автоматично заповнить поле 'Model Class Name' значенням 'Courses'. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Courses, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getGroups, що повертає зв'язок усіх груп, що належать до цього курсу

3.4.3.1.5 Генерування моделі групи

Модель курсу повинна мати наступні поля: номер, назва, ідентифікатор типу групи, ідентифікатор курсу, ідентифікатор викладача, дата створення запису, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'groups', після чого система автоматично заповнить поле 'Model Class Name' значенням 'Groups'. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Groups, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці

- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getCourse, що повертає зв'язок курсу, що належить до цієї групи
- getTeacher, що повертає зв'язок викладача, що належить до цієї групи
- getType, що повертає зв'язок типу групи, що належить до цієї групи
- getStudents, що повертає зв'язок усіх студентів, що належить до цієї групи
- getTimesGroupsAudiences, що повертає зв'язок записів цієї групи у розкладі

3.4.3.1.6 Генерування моделі аудиторії

Модель аудиторії повинна мати наступні поля: номер будівлі, номер аудиторії, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'audiences', після чого система автоматично заповнить поле 'Model Class Name' значенням 'Audiences'. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Audiences, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getTimesGroupsAudiences, що повертає зв'язок записів цієї аудиторії у розкладі

3.4.3.1.7 Генерування моделі робочого дня

Модель дня повинна мати наступні поля: назва, дата створення запису, дата останньої зміни запису.

У полі генератору ‘Table name’ обираємо таблицю ‘days’, після чого система автоматично заповнить поле ‘Model Class Name’ значенням ‘Days’. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Days, що наслідуює клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getTimes, що повертає зв’язок усіх робочих часів, що належать до цього дня

3.4.3.1.8 Генерування моделі робочого часу

Модель робочого часу повинна мати наступні поля: назва, ідентифікатор робочого дня, дата останньої зміни запису.

У полі генератору ‘Table name’ обираємо таблицю ‘times’, після чого система автоматично заповнить поле ‘Model Class Name’ значенням ‘Times’. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Times, що наслідуює клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getDay, що повертає зв’язок робочого дня, що належить до цього часу
- getTimesGroupsAudiences, що повертає зв’язок записів цього часу розкладі

3.4.3.1.9 Генерування моделі студента

Модель студента повинна мати наступні поля: імя, прізвище, дата створення запису, дата останньої зміни запису.

У полі генератору 'Table name' обираємо таблицю 'students, після чого система автоматично заповнить поле 'Model Class Name' значенням 'Students'. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Times, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getGroups, що повертає зв'язок груп, що належать до цього студента
- getGroupsStudents, що повертає зв'язок записів до груп цього студента

3.4.3.1.10 Генерування моделі, що відповідає за зв'язок студента і групи

Модель зв'язку студента і групи повинна мати наступні поля: ідентифікатор студента, ідентифікатор групи.

У полі генератору 'Table name' обираємо таблицю 'groups_students, після чого система автоматично заповнить поле 'Model Class Name' значенням 'GroupsStudents. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Times, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getGroup, що повертає зв'язок групи, що належить до цього запису
- getStudent, що повертає зв'язок студента, що належить до цього запису

3.4.3.1.11 Генерування моделі, що відповідає за запис групи до розкладу

Модель зв'язку групи, часу, аудиторії і тижня повинна мати наступні поля: ідентифікатор часу, ідентифікатор групи, ідентифікатор аудиторії, номер тижня, дату створення запису.

У полі генератору 'Table name' обираємо таблицю 'times_groups_audiences', після чого система автоматично заповнить поле 'Model Class Name' значенням 'TimesGroupsAudiences'. Переглядаємо створену модель та зберігаємо у директорії проекту.

У моделі Times, що наслідує клас ActiveRecord було згенеровано наступні функції:

- tableName, що повертає назву таблиці
- rules, що повертає правила для кожного поля моделі
- attributeLabels, що повертає назви полів для користувача
- getGroup, що повертає зв'язок групи, що належить до цього запису
- getTime, що повертає зв'язок часу, що належить до цього запису
- getAudience, що повертає зв'язок аудиторії, що належить до цього запису
- getGroupsStudents, що повертає зв'язок студентів та груп, що належать до цієї групи

```
/**
 * Gets query for [[Time]].
 *
 * @return \yii\db\ActiveQuery
 */
public function getTime()
{
    return $this->hasOne(Times::className(), ['id' => 'time_id']);
}
```

Рисунок 3.2.1.5.11 Приклад згенерованого зв'язку для моделі зв'язку груп, аудиторій, часу та тижня

3.4.4 Генерування файлів контролерів та представлень для базових операцій для взаємодії з записами

Для генерування файлів контролерів та представлень, що відповідають за базові операції для взаємодії з записами у додатку Gii треба перейти до пункту 'CRUD Generator' та натиснути на кнопку з написом 'Start'. Слід зазначити, що до базових операцій відноситься наступний функціонал: можливість створення нового запису, читання існуючого запису, оновлення існуючого запису, видалення існуючого запису, перегляд усіх записів. Після чого система пропонує обрати модель, на основі якої будуть створені файли контролеру та представлення. Також обов'язково слід зазначити простір імен для контролера та простір імен для представлення. З додаткових можливостей – зміна двигуна додатку відображення, уточнення класу пошуку записів та шаблон файлу на основі якого будуть згенеровані контролер та представлення. Після заповнення необхідних полів надається можливість попереднього перегляду та збереження файлів у відповідні місця проекту.

Після перегляду та збереження контролерів та представлень для реалізації базових операцій до наступних моделей: Students, Audiences, Courses, Groups, GroupTypes, Posts, Teachers, Times, Days у папці проекту '/controllers' з'являються наступні файли: StudentsController.php, AudiencesController.php, CoursesController.php, GroupsController.php, PostsController.php, TeachersController.php, TimesController.php, DaysController.php.

Оглянувши структуру згенерованих файлів робимо висновок, що усі вони містять у собі 7 наступних функцій:

- behaviors, що повертає поведінку для функцій
- actionIndex, що знаходить усі записи моделі, що відноситься до контролеру та повертає відображення файлу 'index' з переданими до нього даними

- `actionView`, що приймає ідентифікатор запису, знаходить запис та повертає відображення файлу `'view'` з переданими до нього даними
- `actionCreate`, що створює новий запис на основі даних взятих з запиту та повертає відображення файлу представлення `'create'`
- `actionUpdate`, що приймає ідентифікатор запису, знаходить запис, змінює його на ті, що приходять з запитом та повертає відображення файлу представлення `'update'` з переданими до нього даними
- `actionDelete`, що приймає ідентифікатор запису та видаляє запис, що має цей ідентифікатор
- `findModel`, що приймає ідентифікатор, та повертає запис, що йому належить, або повідомляє користувача про відсутність такого запису

В середині були використані наступні функції, що належать до класу `ActiveRecord`:

- `load` – заповнює поля екземпляру класу даними з переданого масиву
- `save` – створення запису на основі екземпляру класу
- `findOne` – пошук одного запису з переданим ідентифікатором
- `delete` – видаляє запис, що відповідає екземпляру класу

```
/**
 * Creates a new Days model.
 * If creation is successful, the browser will be redirected to the 'view' page.
 * @return mixed
 */
public function actionCreate()
{
    $model = new Days();

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    }

    return $this->render( view: 'create', [
        'model' => $model,
    ]);
}
```

Рисунок 3.4.4.1 Приклад функції `actionCreate` у `DaysController.php`

Окрім файлів контролерів були згенеровані файли представлень, відображення яких повертається у контролерах. У нашому випадку файли представлень знаходяться у папці ‘/views’ проекту. Для кожної моделі було згенеровано по папці з наступними файлами: _form.php, create.php, index.php, view.php, update.php. Ці файли виконують роль відображення даних користувачу.

3.4.5 Налаштування головної сторінки

Для того, щоб спростити використання додатку створюється навігаційна панель сайту у яку, шляхом створення списку, додаються усі потрібні посилання. Посилання у Yii генеруються наступним шляхом: ‘/<Назва контролеру>/<Назва action>’. У файл /views/site/index.php’ додаємо HTML код, що відповідає спискам.

Сторінка робочих часів
Сторінка робочих днів
Сторінка курсів
Сторінка вчителів
Сторінка груп
Сторінка типів груп
Сторінка аудиторій
Сторінка студентів
Конструктор розкладу

Рисунок 3.2.4.1 Список посилань

```

<div>
  <ul class="list-group">
    <li class="list-group-item"><a href="/times">Сторінка робочих часів</a></li>
    <li class="list-group-item"><a href="/days">Сторінка робочих днів</a></li>
  </ul>

  <ul class="list-group">
    <li class="list-group-item"><a href="/courses">Сторінка курсів</a></li>
    <li class="list-group-item"><a href="/teachers">Сторінка вчителів</a></li>
    <li class="list-group-item"><a href="/groups">Сторінка груп</a></li>
    <li class="list-group-item"><a href="/group-types">Сторінка типів груп</a></li>
  </ul>

  <ul class="list-group">
    <li class="list-group-item"><a href="/audiences">Сторінка аудиторій</a></li>
    <li class="list-group-item"><a href="/students">Сторінка студентів</a></li>
  </ul>

  <ul class="list-group">
    <li class="list-group-item"><a href="/schedules">Конструктор розкладу</a></li>
  </ul>
</div>

```

Рисунок 3.2.4.2 HTML код для списку посилань.

3.4.6 Створення системи конструювання розкладу

3.4.6.1 Перегляд розкладу

Для побудови системи конструювання розкладу треба детально продумати модель запису. Як вже було зазначено раніше запис складається з часу, групи, аудиторії та тижня, до якого цей запис відноситься. В свою чергу з часу ми можемо дізнатися день цей запис чинний. З групи ми можемо дізнатися викладача, тип групи, дисципліну. З викладача дізнаємося його посаду. Таким чином можемо побудувати наступний логічний рядок. Для того, щоб побудувати розклад треба знайти усі дні, усі часи, що належать цим дням, усі групи, аудиторії та тижні, що належать цьому зв'язку. Запит слід доповнити викладачем, студентами, дисципліною та типом які відносяться до усіх груп.

Першим кроком слід створити новий контролер з назвою «SchedulesController» у папці '/controllers'. Після цього було зареєстровано

```
$days = Days::find()->toArray()->with([
    'times',
    'times.timesGroupsAudiences' => function (\yii\db\ActiveQuery $query) {
        $query->select(['*', 'group_concat(week order by week)'])->groupBy(['group_id', 'time_id']);
    },
    'times.timesGroupsAudiences.group',
    'times.timesGroupsAudiences.audience',
    'times.timesGroupsAudiences.group.students',
    'times.timesGroupsAudiences.group.course',
    'times.timesGroupsAudiences.group.teacher',
    'times.timesGroupsAudiences.group.teacher.post',
    'times.timesGroupsAudiences.group.type'
])->all();
```

Рисунок 3.4.6.1 Конструювання запиту

функцію `actionIndex`. Головна задача цієї функції – підготовка даних до відображення у представленні.

Для того, щоб правильно відобразити розклад слід використати зв'язки, що були раніше побудовані у моделях. Зв'язок `times.timesGroupsAudiences` групуємо по ідентифікатору групи та ідентифікатору часу та використовуємо функцію `group_concat` отримуємо перелік тижнів, для яких цей запис є дійсним. Функцією `toArray()` відображаємо дані не у вигляді об'єкта, а у вигляді масива. Використовуємо інші допоміжні зв'язки, щоб зрозуміло відобразити інформацію користувачу.

Понеділок	Дисципліна, викладач	Група	Тижні	Аудиторія	Дія	Студенти
	PHP, assistant Doughman Julian	2	1 - 15	1-132	Видалити зв'язок	Debrah Shey Emmanuel Czepiel Hae Maciag Floretta Rumpfelt Cole Pough Sydney Sacks Ma Welman Ligia McClerkin Ma Liske Khalilah Alpperspach Deon Stroder Kylie Pough Kam Buckridge Brigitte Persky Emmanuel Cullins
8:30 Додати зв'язок	PHP, Jr. assistant Gallop Brigitte	3	1 - 15	1-343	Видалити зв'язок	Debrah Shey Valery Haydt Eric Galardo Cole Herrion Jefferey Herrion Ligia McClerkin Elizabeth Doughman Deon Czepiel Jefferey Rinks Cole Gallop Nena Boberg Deon Stroder Harvey Mugleston Gladis Bossardet Lisette Bahoora Jamar Lawter Melodi Persky

Рисунок 3.4.6.2 Вивід таблиці розкладу

3.4.6.2 Створення запису

Для створення запису користувачу важливо розуміти у які тижні певна аудиторія вільна. Для виконання цієї задачі реалізується асинхронна функція `actionAllowed`, яка повертає усі вільні тижні у певний час для певної аудиторії.

Після того, як користувач заповнює поля часу та аудиторії скрипт написаний на мові JavaScript відправляє на сервер запит, та відображає статус вільності кожного тижня. Для реалізації цієї системи було розроблено модальне вікно, у якому обирається аудиторія з доступних, група з доступних та відображаються доступні тижні для цього запису. Вікно відкривається шляхом натискання кнопки «Додати зв'язок» біля часу, на який користувач бажає здійснити запис. Також з контролера передаються усі аудиторії та групи, для їх відображення у модальному вікні.

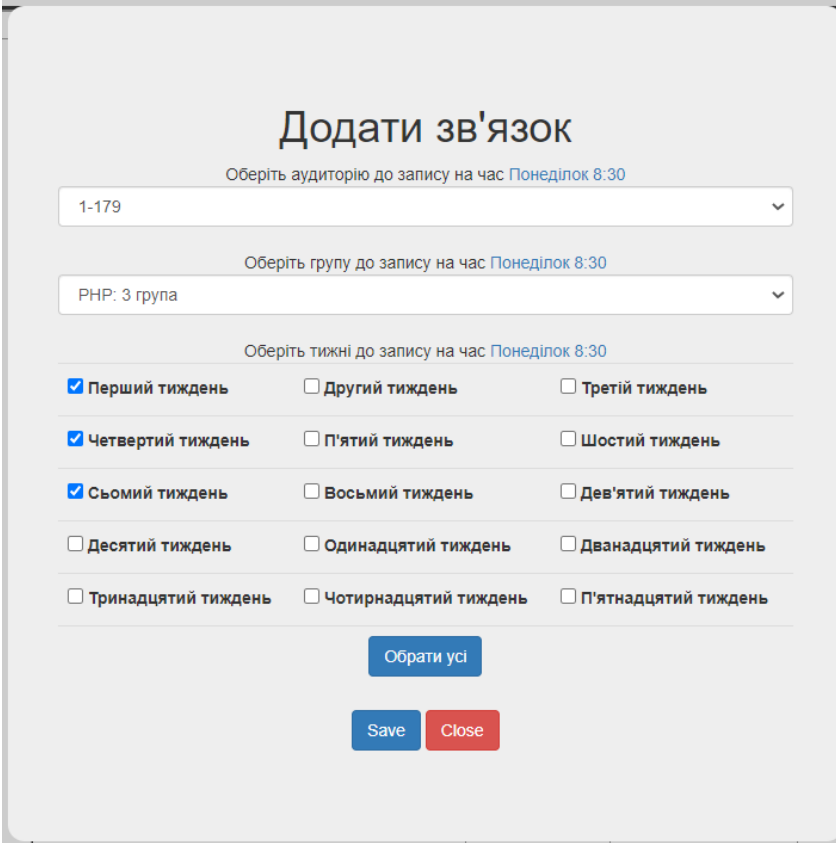
У свою чергу контролер робить ряд перевірок даних, що надходять з запитом. Список перевірок:

- наявність параметра часу
- наявність параметра аудиторії
- наявність параметра групи

- наявність параметра тижня
- наявність тіла запита

3.4.6.3 Видалення записів

Для видалення зв'язку була реалізована функція `actionRemoveAll()`, яка видаляє усі записи певної групи на певний час. Використовується функція `deleteAll`, у котру передається ідентифікатор групи та ідентифікатор часу. Також передбачена перевірка наявності переданих параметрів.



Додати зв'язок

Оберіть аудиторію до запису на час [Понеділок 8:30](#)

1-179

Оберіть групу до запису на час [Понеділок 8:30](#)

РНР: 3 група

Оберіть тижні до запису на час [Понеділок 8:30](#)

<input checked="" type="checkbox"/> Перший тиждень	<input type="checkbox"/> Другий тиждень	<input type="checkbox"/> Третій тиждень
<input checked="" type="checkbox"/> Четвертий тиждень	<input type="checkbox"/> П'ятий тиждень	<input type="checkbox"/> Шостий тиждень
<input checked="" type="checkbox"/> Сьомий тиждень	<input type="checkbox"/> Восьмий тиждень	<input type="checkbox"/> Дев'ятий тиждень
<input type="checkbox"/> Десятий тиждень	<input type="checkbox"/> Одинадцятий тиждень	<input type="checkbox"/> Дванадцятий тиждень
<input type="checkbox"/> Тринадцятий тиждень	<input type="checkbox"/> Чотирнадцятий тиждень	<input type="checkbox"/> П'ятнадцятий тиждень

Обрати усі

Save Close

Рисунок 3.4.6.3 Модальне вікно для додавання запису

3.4.6.4 Перегляд індивідуального розкладу студента

Для перегляду індивідуального розкладу студента треба реалізувати функцію `actionSchedule` у контролері `StudentsController.php`. Треба знайти усі робочі часи, на котрі записана група до якої належить певний студент. Для отримання даних шляхом конструювання був створений наступний запит:

```
$days = Days::find()->asArray()->joinWith([
    'times' => function (\yii\db\ActiveQuery $query) {
        $query->having(['>', 'count(tga.time_id)', 0])->joinWith([
            'timesGroupsAudiences tga' => function (\yii\db\ActiveQuery $query) {
                $query->joinWith([
                    'groupsStudents gs' => function (\yii\db\ActiveQuery $query) {
                        $query->where(['students_id' => $_GET['id']]);
                    },
                    'group',
                    'group.type',
                    'group.course',
                    'group.teacher',
                    'audience'
                ]->select(['tga.*', 'group_concat(week order by week)']->groupBy(['group_id', 'time_id']));
            },
        ],
    );
},
]);
->all();
```

Рисунок 3.4.6.4.1 Побудова запиту для отримання розкладу певного студента

Цей запит повертає усі записи, що відносяться групам, у які записаний певний студент.

Також важливо додати можливість перегляду потижневого розкладу. Для цього реєструємо нову функцію, що називається `actionScheduleByWeek`, яка отримує у якості параметрів ідентифікатор студента та номер тижня, розклад якого переглядається. Важливо зазначити, що відбувається перевірка

```
$days = Days::find()->asArray()->joinWith([
    'times' => function (\yii\db\ActiveQuery $query) {
        $query->having(['>', 'count(tga.time_id)', 0])->joinWith([
            'timesGroupsAudiences tga' => function (\yii\db\ActiveQuery $query) {
                $query->joinWith([
                    'groupsStudents gs' => function (\yii\db\ActiveQuery $query) {
                        $query->where(['students_id' => $_GET['student_id']]);
                    },
                    'group',
                    'group.type',
                    'group.course',
                    'group.teacher',
                    'audience'
                ]->select(['tga.*'])->groupBy(['group_id', 'time_id']->where(['week' => $_GET['week']]));
            },
        ],
    );
},
]);
```

Рисунок 3.4.6.4.2 Побудова запиту для отримання розкладу певного студента на певному тижні.

на існування студента та валідності номеру тижня. Для цієї задачі був створений наступний запит

3.4.6.5 Перегляд розкладу викладача

Важливою частиною додатку є перегляд розкладу викладача. Ця можливість надається за допомогою побудови наступного запиту:

```
$days = Days::find()->toArray()->joinWith([
    'times' => function (\yii\db\ActiveQuery $query) {
        $query->having(['>', 'count(tga.time_id)', 0])->joinWith([
            'timesGroupsAudiences tga' => function (\yii\db\ActiveQuery $query) {
                $query->joinWith([
                    'group',
                    'group.type',
                    'group.course',
                    'group.teacher tr' => function(\yii\db\ActiveQuery $query) {
                        $query->where(['tr.id' => $_GET['id']])->with('post');
                    },
                    'audience'
                ]->select(['tga.*', 'group_concat(week order by week)']->groupBy(['group_id', 'time_id']));
            },
        ],
    );
    },
]);
->all();
```

Рисунок 3.4.6.5.1 Побудова запиту для отримання розкладу певного викладача

3.4.6.6 Інструмент запису студентів у групи

Для реалізації інструменту запису студентів у групи реєструємо `actionEnrollment`, яка повертає усі групи, та ті групи, до яких вже записаний студент. Після заповнення та відправлення форми студента спочатку виписують з усіх груп, а потім записують на відмічені.

3.5 Загальна інформація про використані функції

Під час розробки додатку були використані певні функції, які слід описати детальніше.

- `AppAsset::register` - функція, що належить до класу `AppAsset`. Реєструє усі підключені до проекту додаткові бібліотеки у представленні.
- `json_decode` – функція, що вбудована у мову PHP, що створює об'єкт на основі переданої стрічки, що містить json-об'єкт.

- `Yii::setFlash` – функція, що встановлює повідомлення, яке буде збережено у сесії, а після відображення видалене
- `array_diff` – функція, що вбудована у мову PHP, повертає різницю у даних між двома масивами
- `asJson` – функція, що переносить об’єкт або масив у формат JSON.
- `ActiveRecord::save` – зберігає новий запис, або оновлює існуючий
- `ActiveRecord::unlinkAll` – видаляє, або витирає усі записи, що зв’язані з певною моделю
- `ActiveRecord::select` – запит, поверне тільки вказані колонки
- `ActiveRecord::with` – додає до запиту існуючий зв’язок
- `ActiveRecord::joinWith` – додає до запиту існуючий зв’язок шляхом використання ‘JOIN’
- `ActiveRecord::having` – додає до запиту умову ‘HAVING’
- `ActiveRecord::where` – додає до запиту умову ‘WHERE’
- `ActiveRecord::find` – запускає конструювання запиту
- `ActiveRecord::groupBy` – додає до запиту ‘GROUP BY’

Висновок

Під час виконання курсової роботи були розглянуті проблеми існуючих застосунків для побудови розкладу. Для розробки додатку була обрана бібліотека Yii. Особливості проекту сприяли розвитку навичок проектування реляційних баз даних та розуміння структури Модель-Представлення-Контролер. Була розглянута система генерації PHP коду Gii, що значно зекономила час та сили, що могли бути затрачені на написання моделей, контролерів та представлень. Система міграцій дозволила безпечно легко та безпечно проектувати таблиць. Будування складних запитів безперечно розвинуло розуміння баз даних.

Головна ціль цього проекту – зручний додаток для студентів та працівників Національного Університету «Києво-Могилянська академія», тому в якості руху розвитку додатку була обрана інтеграція з сервісами Microsoft Office 365 та робота з реальними даними, а не прикладними.

Список використаної літератури та електронних ресурсів

1. Українська спільнота Yii Framework [Електронний ресурс] // Повний посібник (v2) – Режим доступу до ресурсу:
<https://yiiframework.com.ua/uk/doc/guide/2/>.
2. Повний посібник з Yii 2.0 [Електронний ресурс] – Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/uk>.
3. API Documentation for Yii 2.0 [Електронний ресурс] // Class yii\db\ActiveRecord – Режим доступу до ресурсу:
<https://www.yiiframework.com/doc/api/2.0/yii-db-activerecord>.
4. Повний посібник з Yii 2.0 [Електронний ресурс] // Генерування коду за допомогою Gii – Режим доступу до ресурсу:
<https://www.yiiframework.com/doc/guide/2.0/uk/start-gii>.
5. Сторінка "Google Calendar" [Електронний ресурс] – Режим доступу до ресурсу: <https://calendar.google.com/calendar/u/0/r>.
6. Сторінка "Google Sheets" [Електронний ресурс] – Режим доступу до ресурсу: https://www.google.com/intl/ru_ua/sheets/about/.