НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

***з дисципліни "Основи програмування"***

Виконала: Левицька Юлія  
 Володимирівна

Група: КП-01

Допущено до захисту

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 семестр 2020/2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено  Керівник роботи    \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ | ЗАХИЩЕНА "\_\_"\_\_\_\_\_\_\_\_\_20ХХр.  з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ |

**Програмний додаток для керування електронною базою навчальних курсів**

Виконавець роботи

Левицька Юлія Володимирівна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021р.

Зміст

[**Вступ**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.cr2eoqpdh4mm)[**4**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.cr2eoqpdh4mm)

[**1. Аналіз предметної галузі**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.b73pd3816f8p)[**5**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.b73pd3816f8p)

[1.1. Аналіз вимог до функціональності програмних засобів](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.qz6vreit6wm5) [5](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.qz6vreit6wm5)

[**2. Аналіз мов програмування та технологій розроблення**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.ewtylcuh8bvc)[**5**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.ewtylcuh8bvc)

[2.1. Мова програмування С](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.fh0j2unhluf2)# [5](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.fh0j2unhluf2)

[2.2. Фреймворк](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.ayd32pcv0jlc) .NET 5

[2.3. База даних SQLite](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.hul0g052l08h) [5](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.hul0g052l08h)

[2.4. Бібліотека](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.6jqqk9eelxe) Terminal.Gui [5](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.6jqqk9eelxe)

2.5. Бібліотека Microsoft.Data.Sqlite

[**3. Опис розроблених програмних засобів**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.kbi4mqs5pjyt)[**6**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.kbi4mqs5pjyt)

[3.1. Загальна структура системи](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.nlu4ckytvy2k) [6](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.nlu4ckytvy2k)

[3.2. Структура бази даних](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.ny4p6qq0v8a9) [7](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.ny4p6qq0v8a9)

[3.3.](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.vkb2811r6l9f) Модуль репозиторіїв даних [8](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.vkb2811r6l9f)

[3.4. Модуль](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.dpq6kpdwm35k) генерації даних 14

3.5. Модуль аутентифікації 14

3.6. Бібліотека ProcessData 14

**4. Аналіз розроблених програмних засобів 10**

4.1. Особливості реалізації 10

4.2. Дизайн та вміст вікон 10

[**Висновки**](https://docs.google.com/document/d/13udNoC-uATtf-lLjJbCEjmSlD841DjKLHfhroFxrJ14/edit#heading=h.6dbqg2jsslr5) **11**

Вступ

2019 рік показав всьому світу, що життя є непередбачуваним і безконтрольним. Мільйони людей опинилися замкненими в чотирьох стінах, без роботи, хобі, можливості подорожувати і,найголовніше, без можливості здобути освіту.   
Саме тому пропонується розробка програмних засобів, які дозволяють людям вчитися та освоювати нові професії не виходячи з дому і не ставлячи своє життя під загрозу.  
Створення системи, що забезпечує швидкий та вільний доступ до структурованої та корисної інформації, як ніколи, є актуальною задачею.

Даний проект присвячено розробленню додатка, який призначено  
для управління електронною базою навчальний курсів, що забезпечує наявність широкого вибору напрямів і спеціальностей.

# 1. Аналіз предметної галузі

## 1.1. Аналіз вимог до функціональності програмних засобів

В процесі аналізу вимог до системи було виділено такі функціональні вимоги:

1. Наявний консольний інтерфейс користувача. Команди використані відповідно до їхнього основного призначення.
   1. Наявно декілька станів інтерфейсу користувача.
   2. Дотримується єдиний стиль для всіх команд
2. Наявні стани для створення і редагування даних.
   1. Можна видаляти дані (з підтвердженням цієї дії).
   2. Наявний стан редагування даних має мати вже заповнені поля із попередніми значеннями. Є можливість відмінити оновлення сутності.
3. Використання списків та таблиць для показу колекцій даних.
   1. Дані в таблиці не є лише відображенням однієї таблиці БД, а містять склеєні чи змінені дані.
   2. Дані у списках та таблицях пагінуються, доступний пошук та фільтрація їх записів.
4. Всі вхідні дані від користувача та з файлів перевіряються, помилки обробляються, користувачу відображається інформація про помилки.
5. Програма має можливість завантаження зображень з файлової системи, їх збереження у сховище даних та їх показ у інтерфейсі користувача.
6. Програма взаємодіє з реляційною базою даних. SQL запити до бази даних не конкатенуються, а їх виконання винесено у спеціальний окремий модуль (вид репозиторія).
   1. База даних містить мінімум 3 таблиці. Таблиці в БД приведені до 1-ї та 2-ї нормальних форм.
   2. Між таблицями бази даних є one-to-many та/або many-to-many зв'язки
7. Наявна реєстрація користувачів у системі. Паролі користувачів не зберігаються у відкритому виді, а хешуються обраним алгоритмом хешування рядків.
   1. Наявний стан аутентифікації. Пароль користувача при введені прихований. Цей стан має сповіщення про некоректність введених користувачем даних.
   2. Програма авторизує доступу до даних по користувачах. У базі є дані, що належать конкретному користувачу.
8. Частина проекту винесена у власні бібліотеки класів. Винесення саме цього коду обгрунтоване.

# 2. Аналіз мов програмування та технологій розроблення

## 2.1. Мова програмування С#

Були використані наступні особливості мови С# як:

* Об’єктно - орієнтованість
* Наслідування
* Стандартні типи даних(int, string, bool..)
* Цикли(for,while),умовні оператори(if-else)
* Розділ програми на модулі
* Оператор try-catch

## 2.2. Фреймворк .NET

## Фреймворк .NET має такі можливості:

## 1. Кожна бібліотека в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок

## 2. .NET — крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows і варіант технології для ОС Linux

## 3. Компілятори для .NET випускаються багатьма фірмами.

## 4. .NET поділяється на дві основні частини — середовище виконання та інструментарій розробки.

## 2.3. База даних SQLite

Були використані наступні SQL скрипти:

● UPDATE .... SET … WHERE …

● INSERT INTO … VALUES ....

● SELECT \* FROM …

● SELECT \* FROM ... WHERE …

● UPDATE... SET ... WHERE …

● DELETE FROM ... WHERE …

● SELECT \* FROM … WHERE ... AND …

● SELECT \* FROM ... WHERE ... LIMIT ... OFFSET …

● DELETE FROM ... WHERE ... AND …

● SELECT \* FROM ... WHERE ... AND …

● DELETE FROM … WHERE ... AND …

● SELECT COUNT(\*) AS count\_ FROM ... WHERE ...

Також були використані такі особливості:

● використання FK для створення звʼязків між таблицями

● автоінкременте значення PK в таблицях

## 2.4. Бібліотека Terminal.Gui Бібліотека була використана для реалізації подійно - орієнтованої програми. Класи, що були використані: Button,Label,TextField,DateField,ListView,Window,Dialog

## 2.5. Бібліотека Microsoft.Data.Sqlite Microsoft.Data.Sqlite - це легкий постачальник ADO.NET для SQLite. Ця бібліотека реалізує загальні абстракції ADO.NET для з'єднань, команд, зчитувачів даних тощо.

# 3. Опис розроблених програмних засобів

## 3.1. Загальна структура системи

Програмні засоби реалізовані у вигляді десктопного додатку з графічним інтерфейсом користувача. Структурна схема системи зображена на рис.

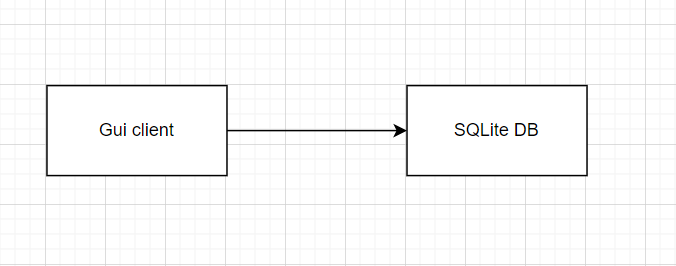


Рис. Структурна схема системи

Проект розбитий на такі модулі:

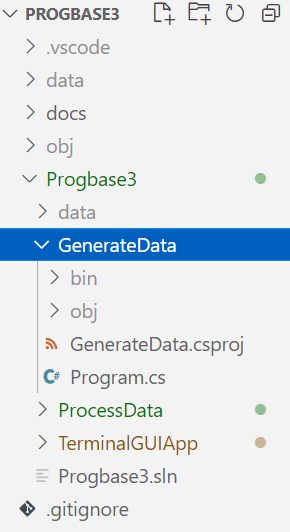
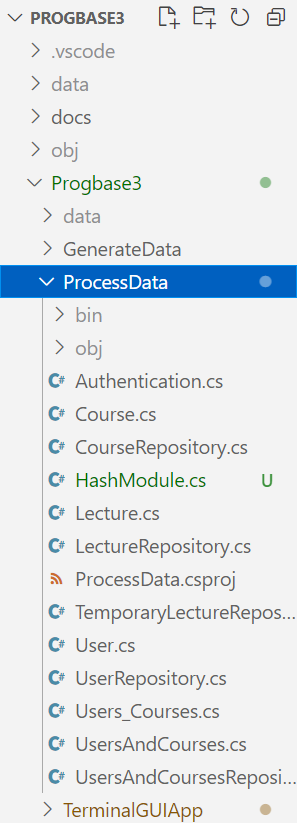
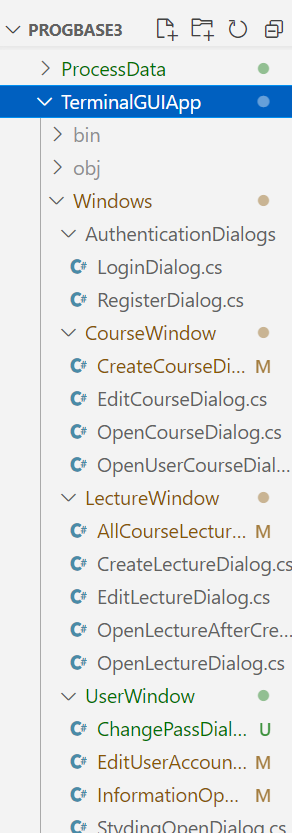
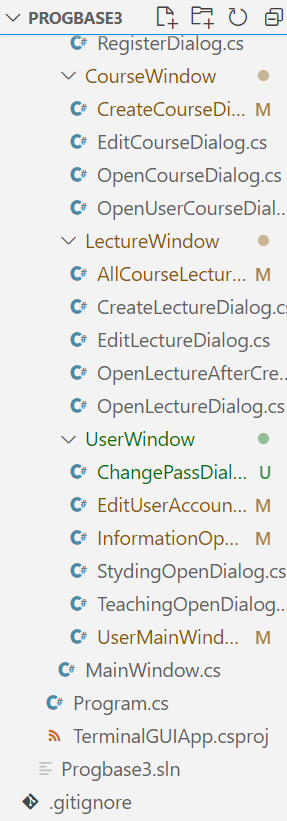
   

Рис. Модулі проекту

Діаграма залежностей модулів IDE:

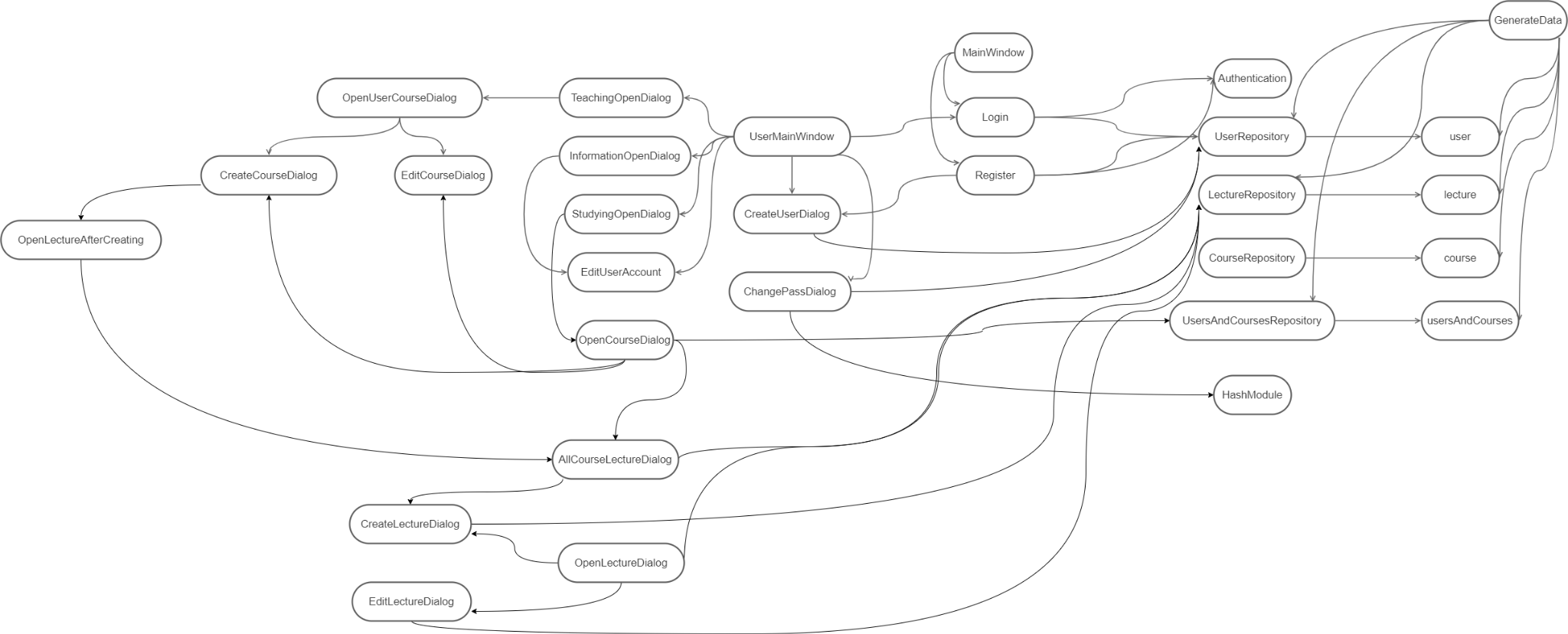


Рис. Залежність модулів

## 3.2. Структура бази даних

На рис. представлена діаграма структури бази даних системи.

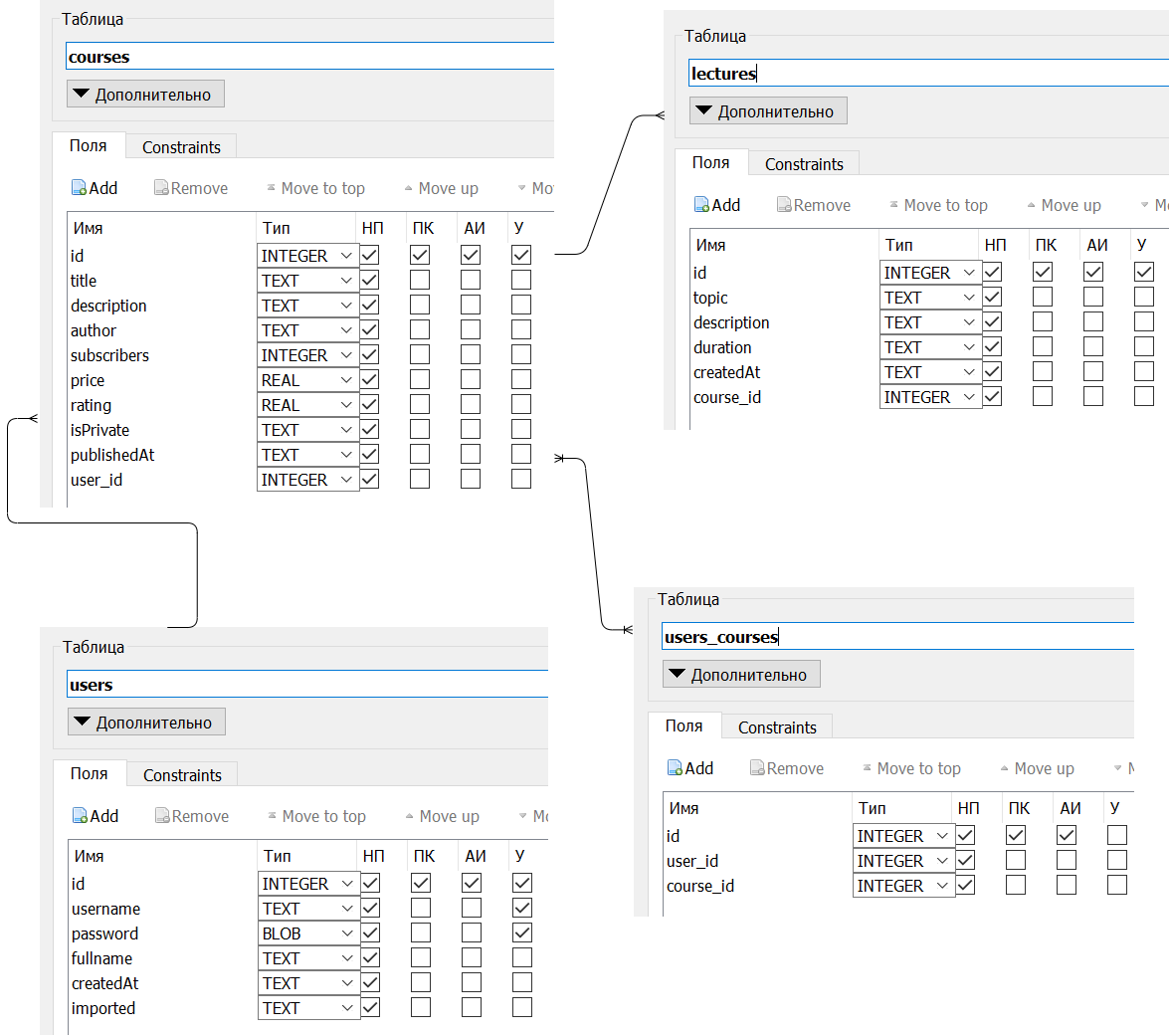


Рис. Структура бази даних

Опис таблиць:

* **courses** - містить інформацію про курси. Поля:

1. id(PK) - автоінкрементне поле РК
2. title(TEXT) - поле для зберігання назви курсу
3. description(TEXT) - поле для зберігання опису курсу
4. author(TEXT) - поле для зберігання ім’я автора курсу
5. subscribers(INTEGER) - поле для зберігання кількості підписників курсу
6. price(REAL) - поле для зберігання ціни курсу
7. rating(REAL) - поле для зберігання рейтингу курсу
8. isPrivate(TEXT) - поле для зберігання статусу курсу
9. publishedAt(TEXT) - поле для зберігання дати публікації курсу
10. user\_id(INTEGER) - поле для зберігання ключа користувача,який є автором курсу

* **lectures** - містить інформацію про лекції курсу. Поля:

1. id(PK) - автоінкрементне поле РК
2. topic(TEXT) - поле для зберігання теми лекції
3. description(TEXT) - поле для зберігання опису лекції
4. duration(TEXT) - поле для зберігання тривалості лекції
5. createdAt(TEXT) - поле для зберігання часу публікації
6. course\_id(INTEGER) -поле для зберігання ключа курсу, якому належить лекція

* **users** - містить інформацію про користувачів. Поля:

1. id(PK) - автоінкрементне поле РК
2. username(TEXT) - поле для зберігання нікнейму
3. password(TEXT) - поле для зберігання паролю
4. fullname(TEXT) - поле для зберігання повного ім’я
5. createdAt(TEXT) - поле для зберігання дати рееєстрації
6. imported(TEXT) - поле для зберігання статусу

* **users\_courses** - містить інформацію про користувачів і курси, на які вони підписались. Поля:  
   1. id(PK) - автоінкрементне поле РК  
   2. user\_id(INTEGER) - поле для зберігання ключа користувача  
   3.course\_id(INTEGER) - поле для зберігання ключа курсу

## 3.3. Модулі репозиторіїв даних

Інтерфейс модуля:

|  |
| --- |
| **CourseRepository.сs** |
| int Insert(Course course) bool Update(int courseId, Course course)  bool DeleteById(int courseId)  Course[] GetAll()  int GetSearchPagesCount(int pageSize, string searchValue)  List<Course> GetSearchPage(string searchValue, int pageNum, int pageSize)  int GetSearchPagesCountByUserId(int pageSize, string searchValue, int userId)  List<Course> GetSearchPageByUserId(string searchValue, int pageNum, int pageSize, int userId)  int GetSearchPagesCountOfSubscibedCourses(List<int> coursesId, int pageSize, string searchValue)  List<Course> GetSearchPageOfSubscribedCourses(List<int> coursesId, string searchValue, int pageNum, int pageSize)  List<Course> ReadCourses(SqliteDataReader reader)  int[] GetAllCoursesIds()  List<int> GetListOfIds(SqliteDataReader reader)  Course ReadCourse(SqliteDataReader reader)  bool CheckIfUserIsCourseAuthor(int userId, int courseId) |

## Модуль дозволяє взаємодіяти з таблицею курсів. Він виконує базові CRUD операції, а також додаткові, які забезпечують пагінацію і пошук.

Інтерфейс модуля:

|  |
| --- |
| **LectureRepository.сs** |
| int Insert(Lecture lecture)  bool Update(int lectureId, Lecture lecture)  bool DeleteById(int id)  int GetSearchPagesCount(int pageSize, string searchValue, int courseId)  List<Lecture> GetSearchPage(string searchValue, int pageNum, int pageSize, int courseId)  List<Lecture> ReadLectures(SqliteDataReader reader)  Lecture ReadLecture(SqliteDataReader reader)  Lecture[] GetAllCourseLectures(int course\_id) |

## Модуль дозволяє взаємодіяти з таблицею лекцій. Він виконує базові CRUD операції, а також додаткові, які забезпечують пагінацію і пошук.

Інтерфейс модуля:

|  |
| --- |
| **UserRepository.сs** |
| bool UserExists(string username)  User[] GetAllUsers()  int[] GetAllUsersIds()  int Insert(User user)  User GetByUsername(string username)  bool Update(int userId, User user)  bool DeleteByUsername(string username)  List<User> ReadUsers(SqliteDataReader reader)  User ReadUser(SqliteDataReader reader)  List<int> GetListOfIds(SqliteDataReader reader) |

## Модуль дозволяє взаємодіяти з таблицею користувачів. Він виконує базові CRUD операції, а також додаткові, які забезпечують пагінацію і пошук.

Інтерфейс модуля:

|  |
| --- |
| **UsersAndCoursesRepository.сs** |
| int Insert(UsersAndCourses usersAndCourses)  bool Delete(int userId, int courseId)  bool isExists(int userId, int courseId)  List<int> GetAllUserCoursesId(int userId)  List<int> ReadAllUserCourses(SqliteDataReader reader) |

## Модуль дозволяє взаємодіяти з таблицею користувачів і курсів. Він виконує базові CRUD операції.

## 

## 3.4. Модуль генерації даних Модуль GenerateData містить функції, які дозволяють генерувати велику кількість даних, в залежності від команди користувача

## 3.5. Модуль аутентифікації Модуль Authentication має два основні методи: реєстрацію та авторизацію. Реєстрація створює нового користувача, попередньо перевіривши доступність логіна, а авторизація перевіряє коректність логіна і пароля та, на основі перевірки, дозволяє або не дозволяє вхід в програму.

## 3.6. Бібліотека ProcessData Бібліотека ProcessData призначена для взаємодії користувача з базою даних. Бібліотека дозволяє виконує базові CRUD операції, пагінацію і пошук всіх сутностей.

# 4. Аналіз розроблених програмних засобів

## 4.1. Особливості реалізації

Кожен користувач має доступ до такого функционалу, як : перегляд каталогу курсів та створення/редагування/видалення власних курсів. В залежності від статусу курсу (приватний/неприватний) користувач може або не може підписатися на курс, також якщо користувач не є підписником курсу,він не може переглядати лекції даного курсу. Автор курсу може відкривати чи закривати доступ до курсу.

Щоб авторизуватися, користувачу додатка потрібно ввести в поля коректний логін і пароль. При вводі пароль не показується.

Всі паролі в базі даних зберігаються в захешованому вигляді. Для хешування використовується алгоритм SHA256. Це забезпечує надійність зберігання паролів у базі даних.

## 4.2. Інтерфейс користувача

Для показу даних система має інтерфейс користувача, що складається з:

1. Вікно входу в програму
2. Вікно логіну
3. Вікно реєстрації
4. Вікно користувача
5. Вікно створення користувача
6. Вікно редагування даних користувача
7. Вікно редагування пароля
8. Вікно інформації про користувача
9. Вікно підписок користувача
10. Вікно власних курсів користувача
11. Вікно створення курсу
12. Вікно редагування курсу
13. Вікно перегляду курсу
14. Вікно всіх лекцій курсу
15. Вікно створення лекції
16. Вікно редагування лекції
17. Вікно перегляду лекції при створенні курсу
18. Вікно перегляду лекції

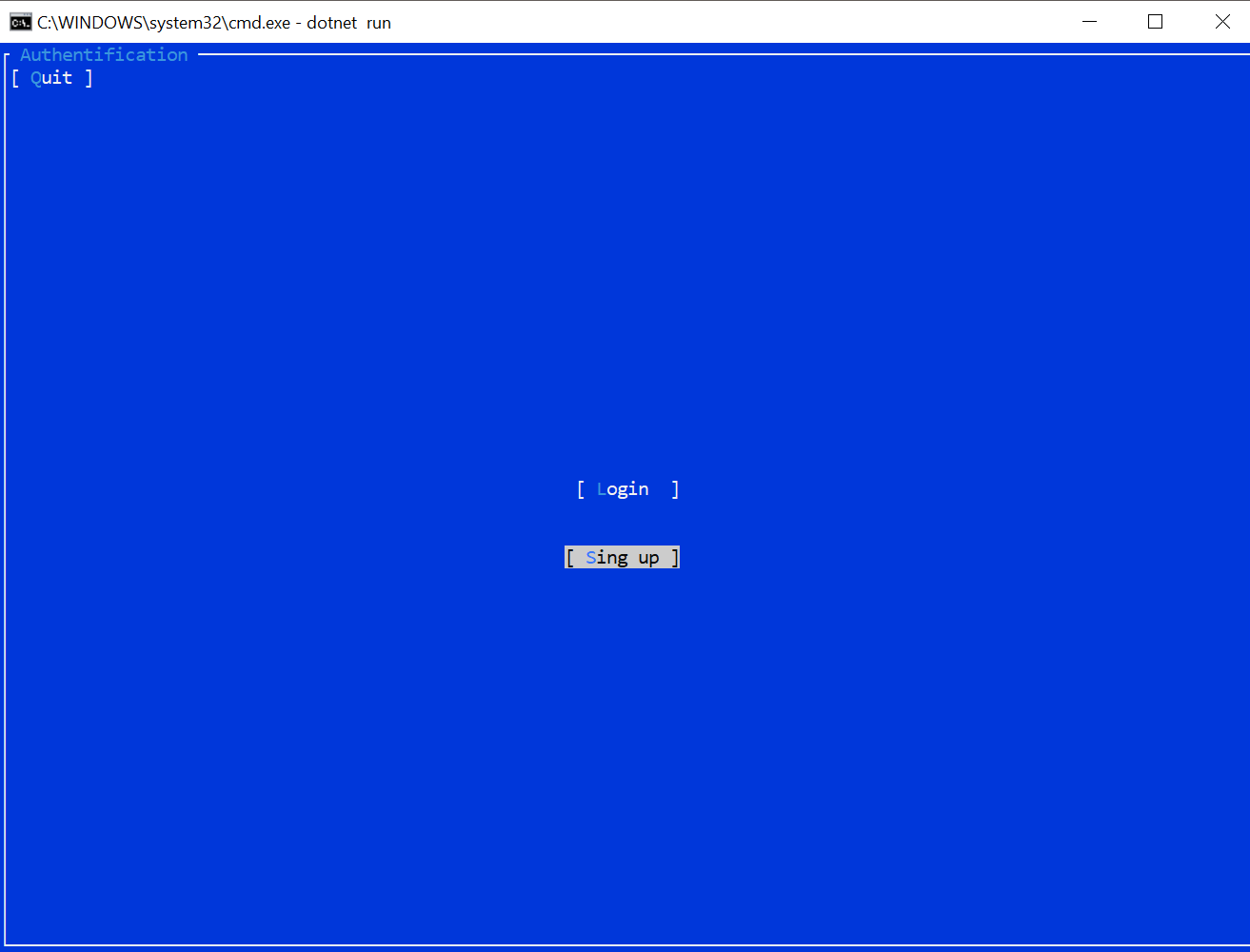


Рис. Вікно входу в програму

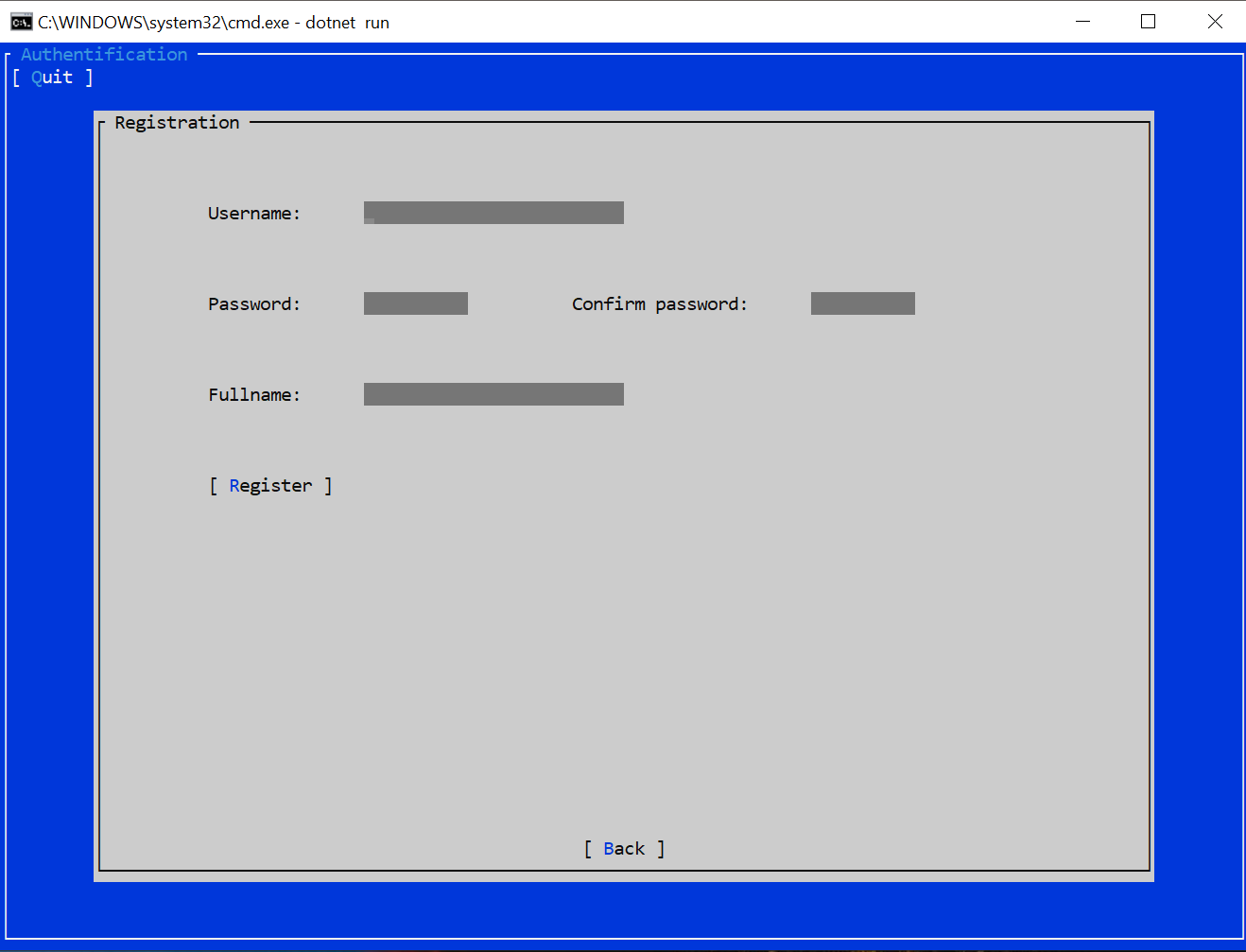


Рис. Вікно реєстрації



Рис. Вікно логіну

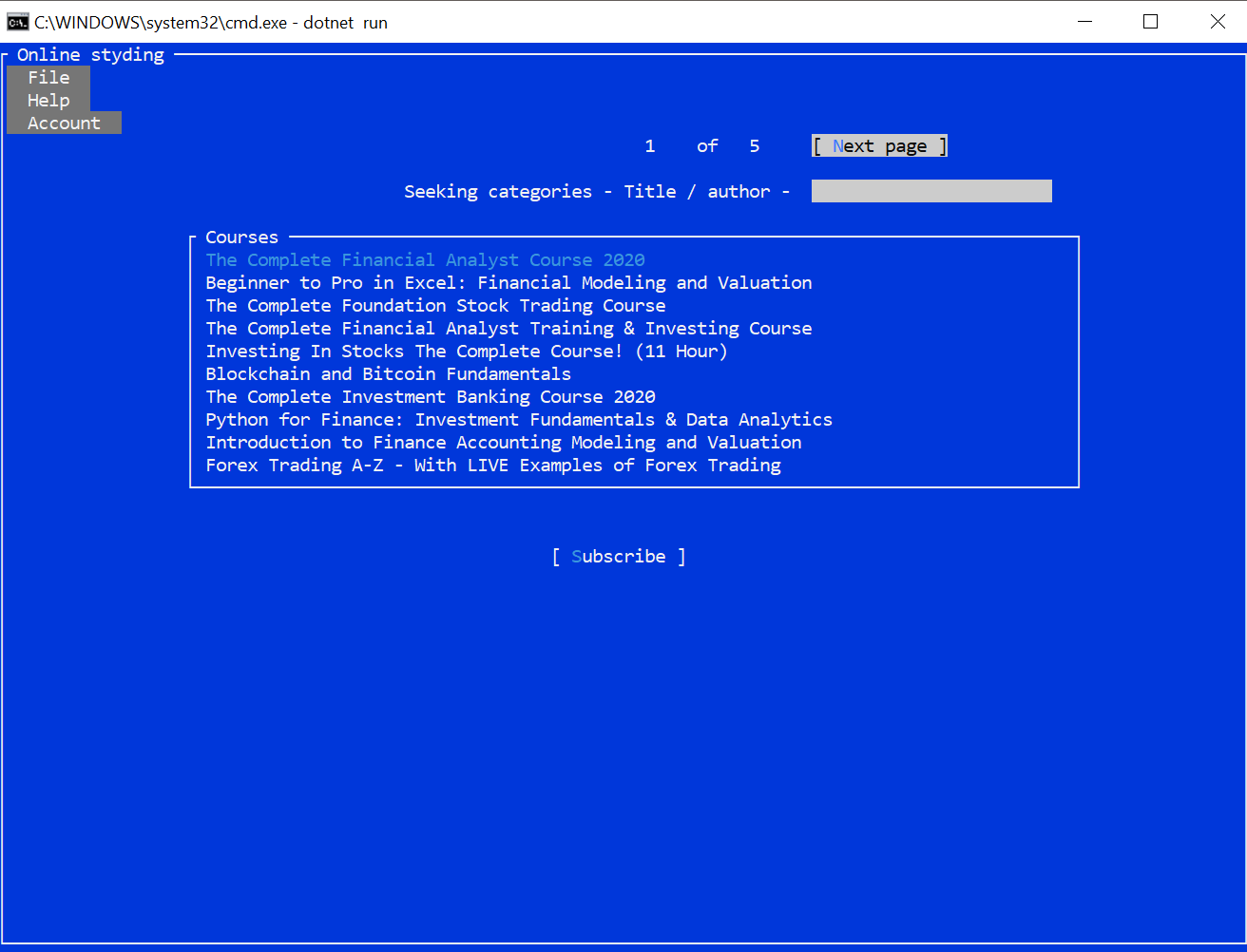


Рис. Вікно користувача

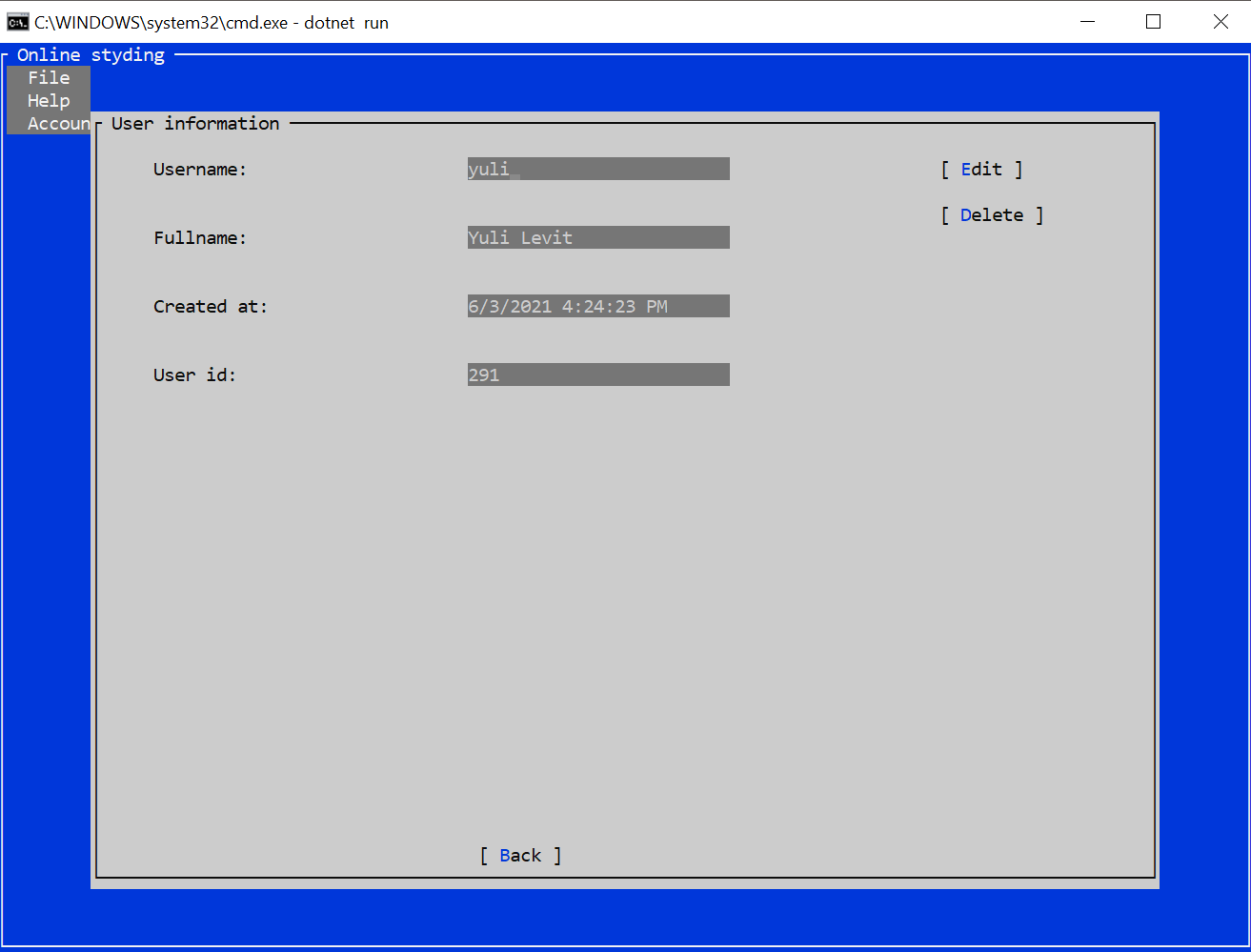


Рис. Вікно перегляду інформації користувача

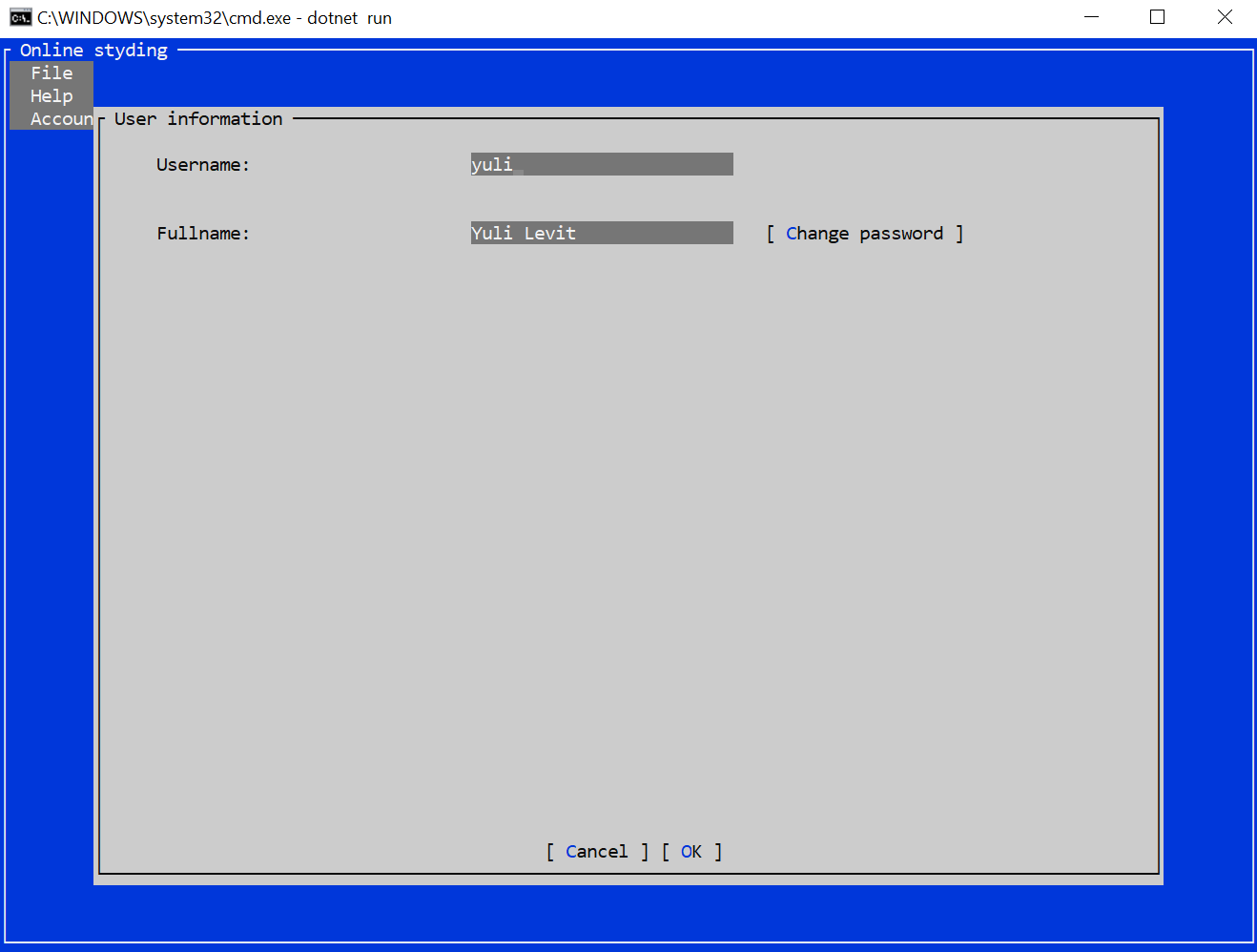


Рис. Вікно редагування користувача

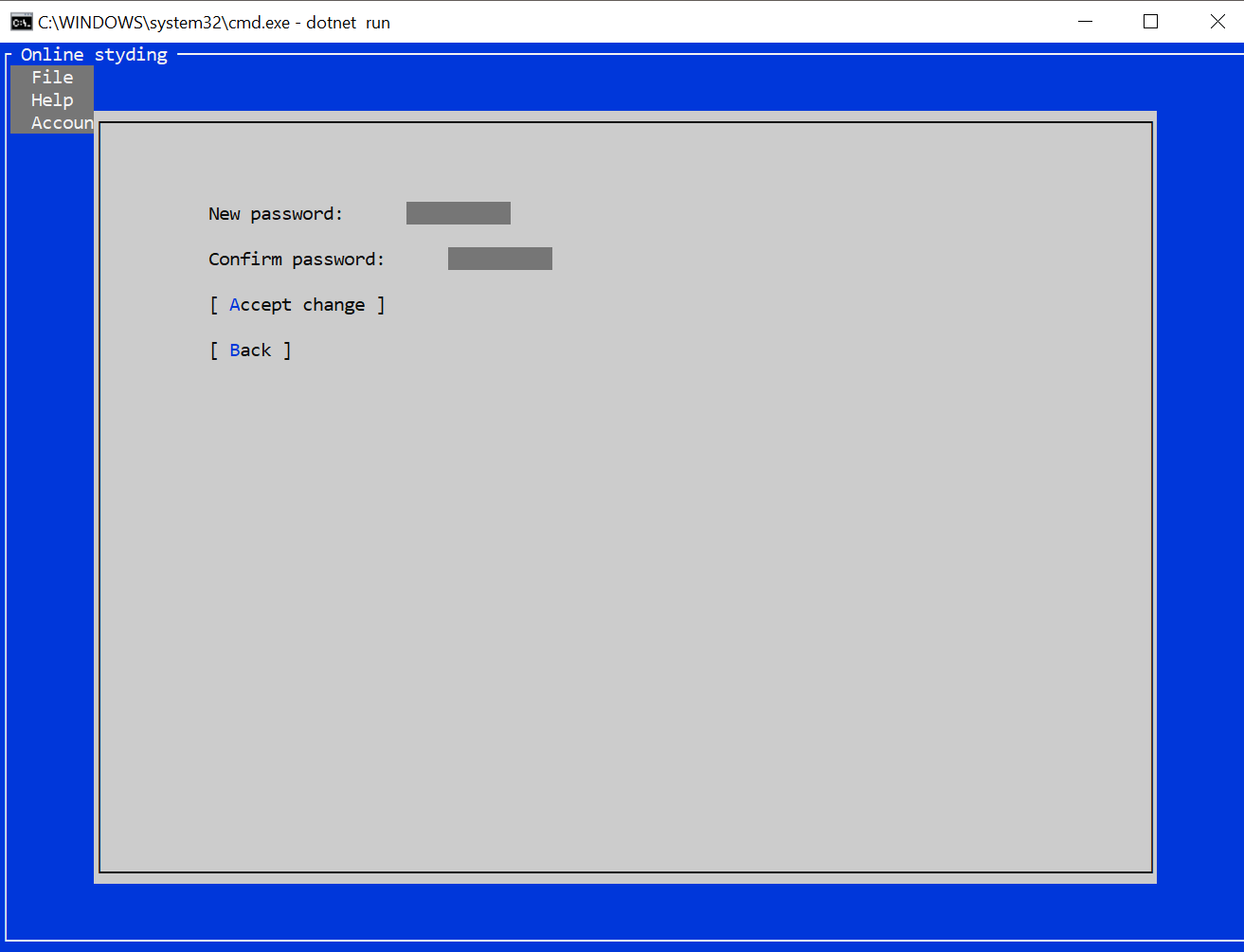


Рис. Вікно редагування паролю

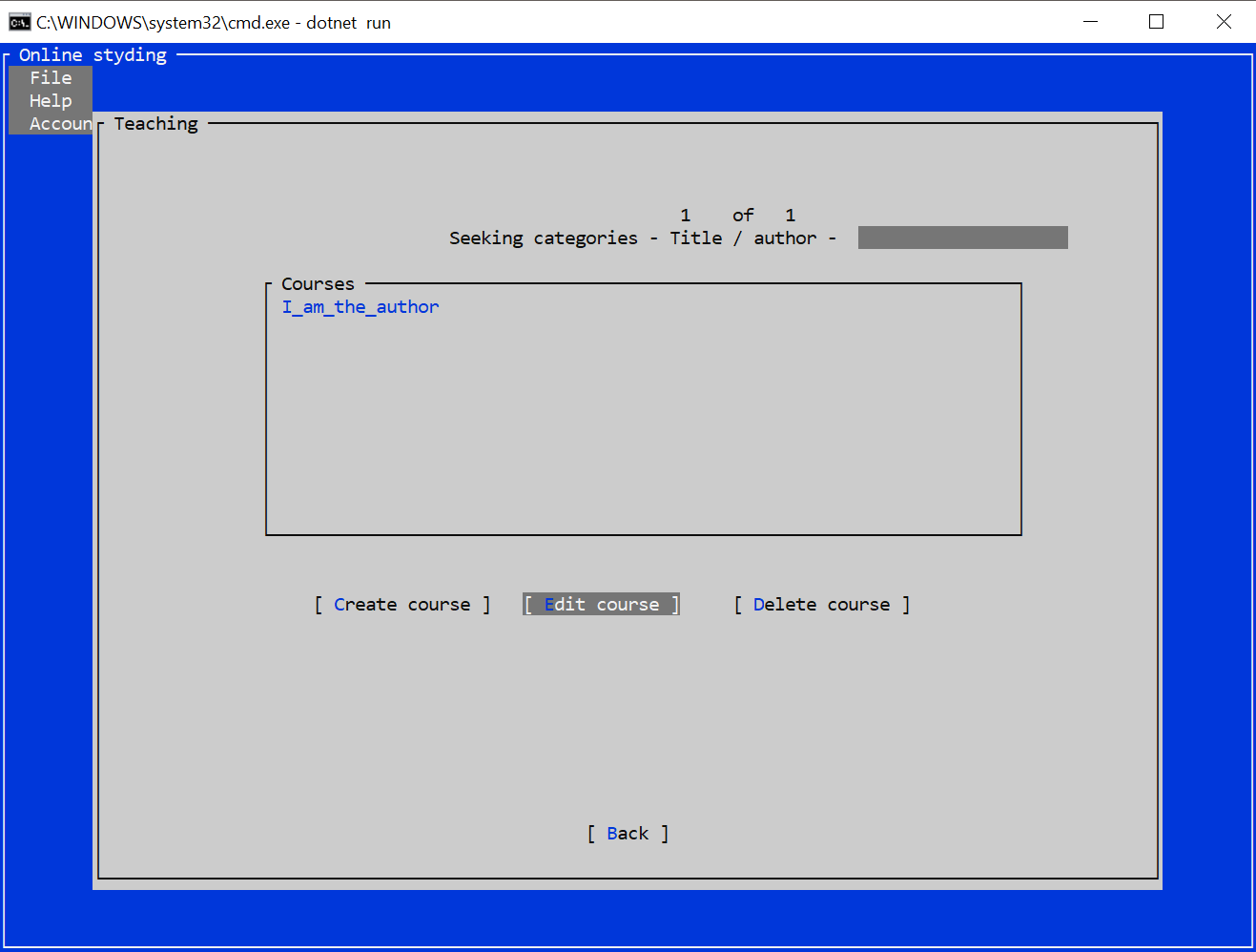


Рис. Вікно власних курсів користувача

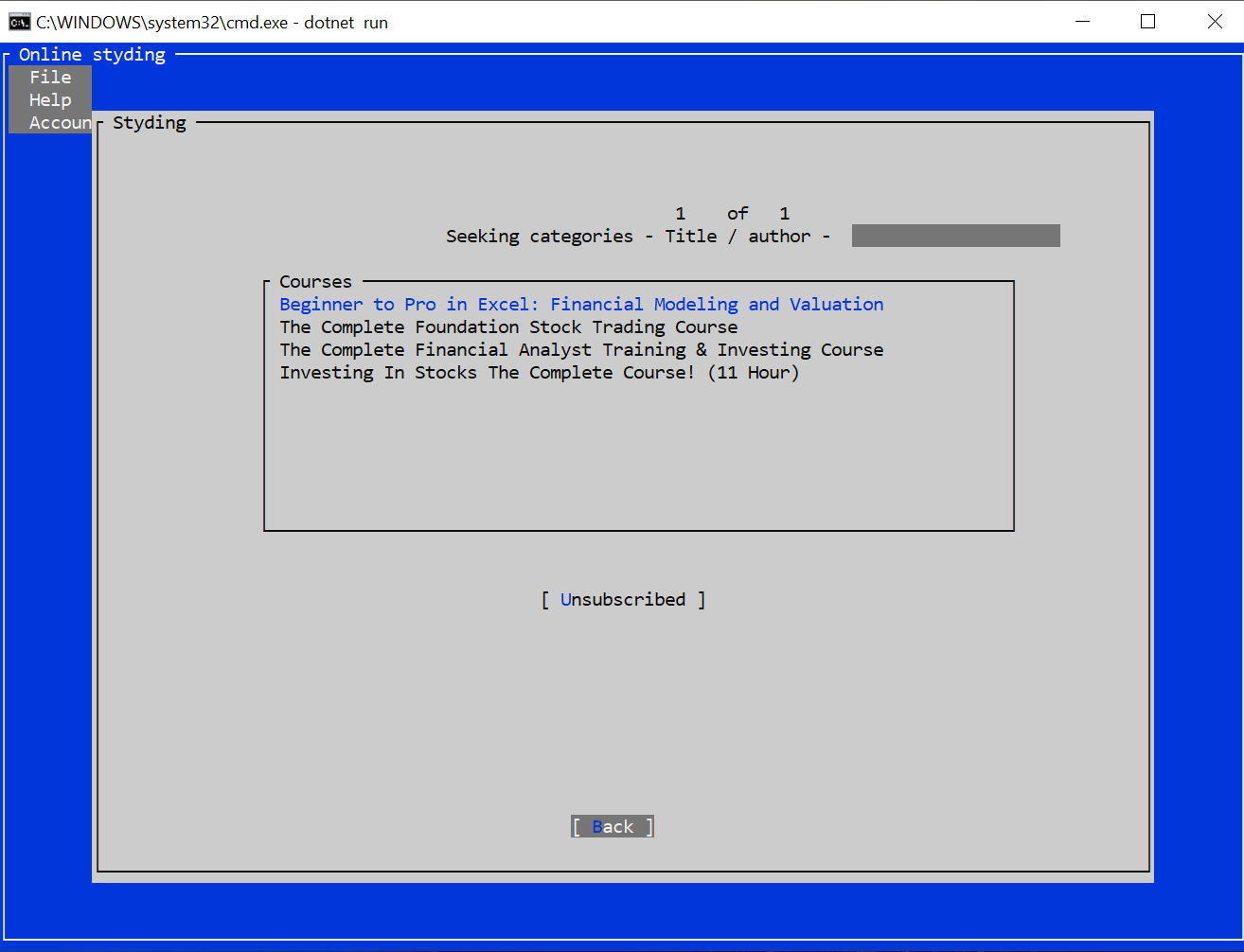


Рис. Вікно підписок користувача

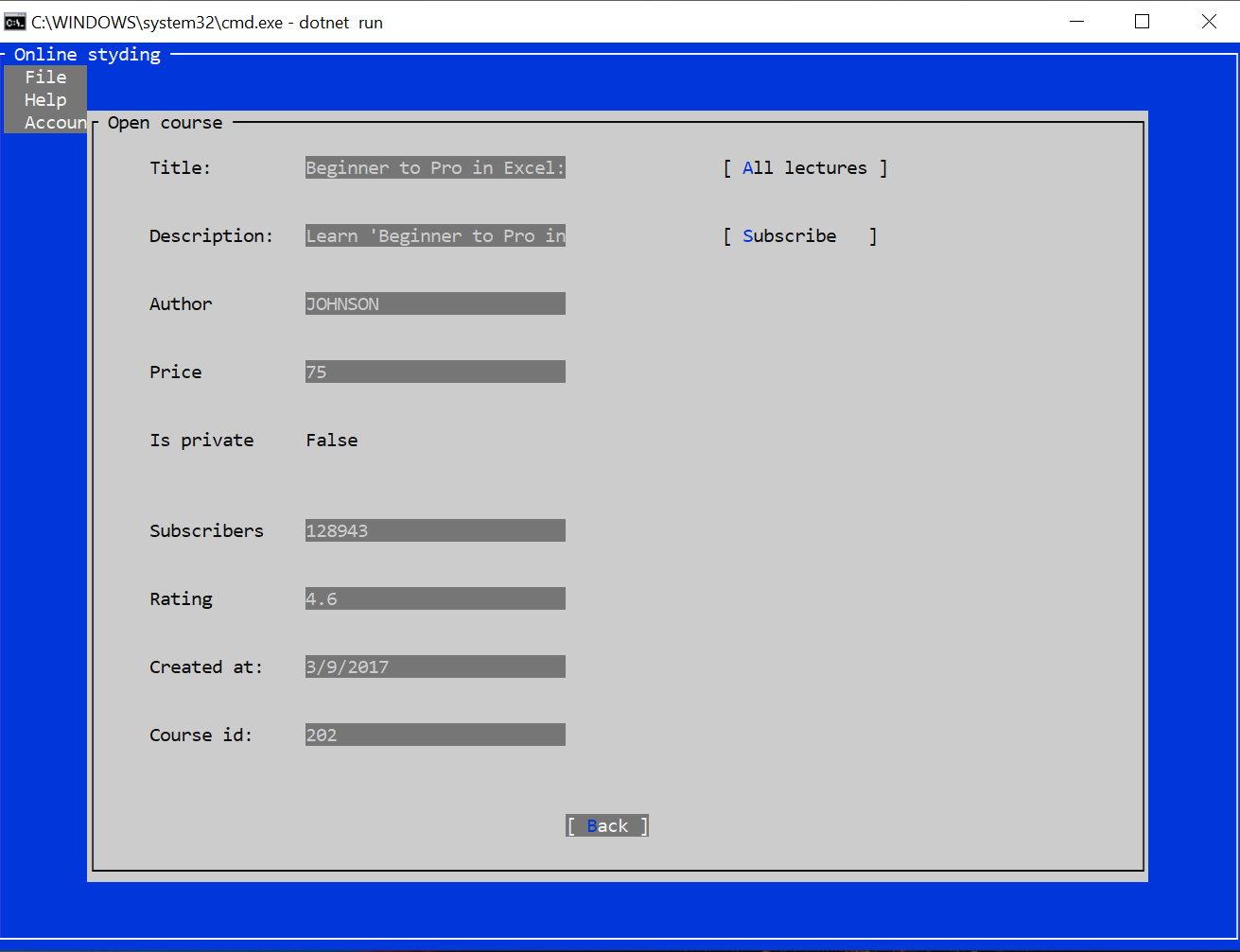


Рис. Вікно перегляду курса

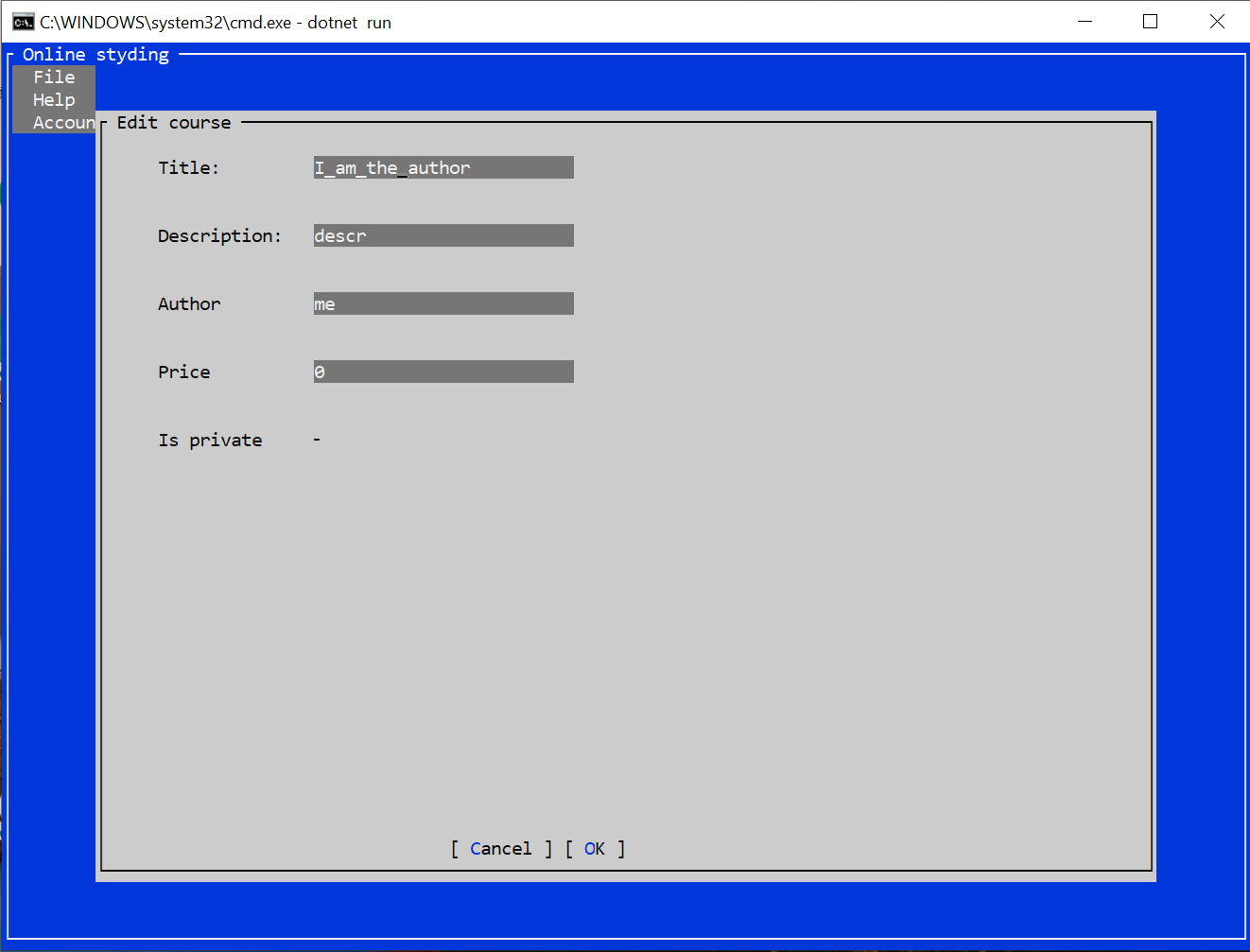


Рис. Вікно редагування курсу

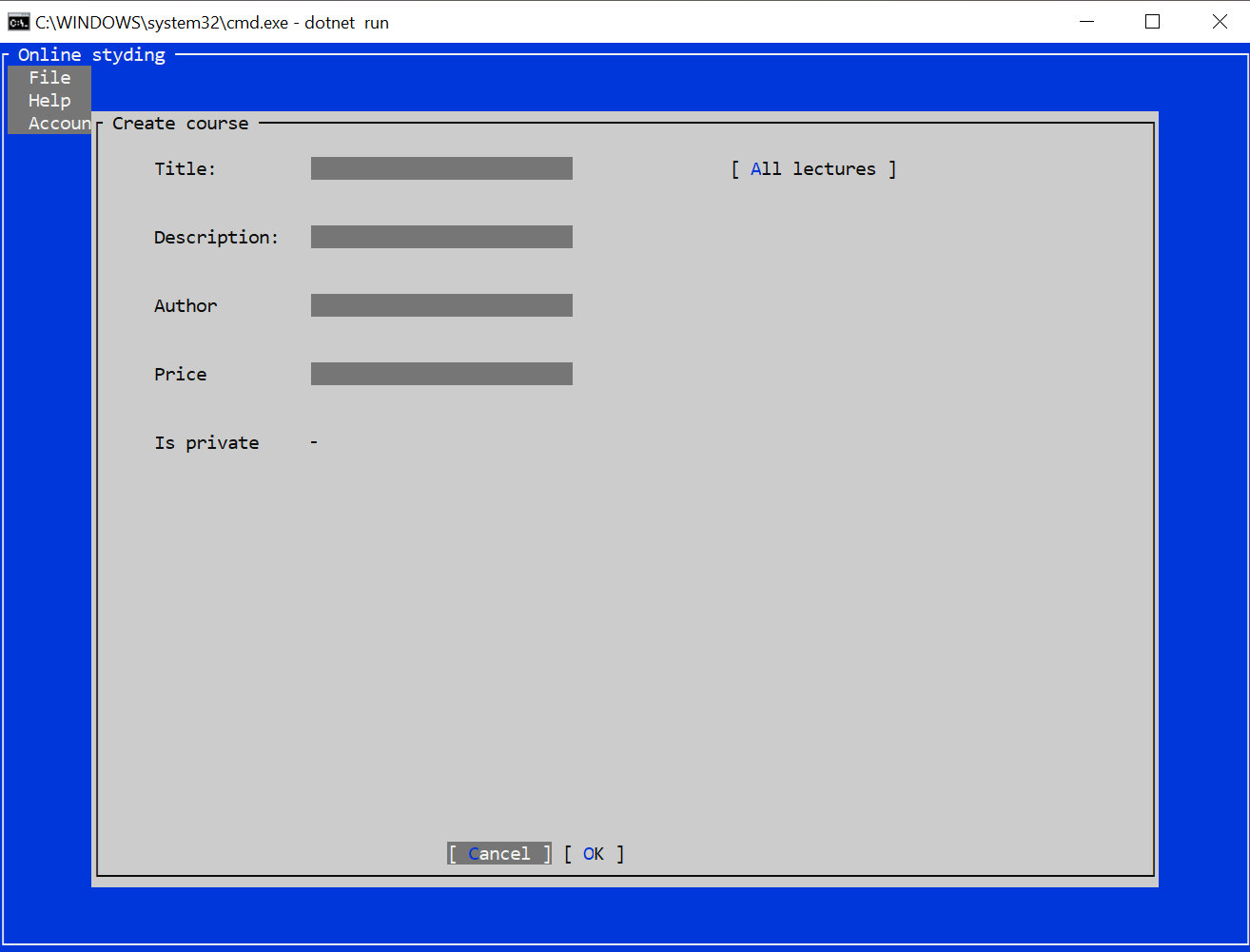


Рис. Вікно створення курсу

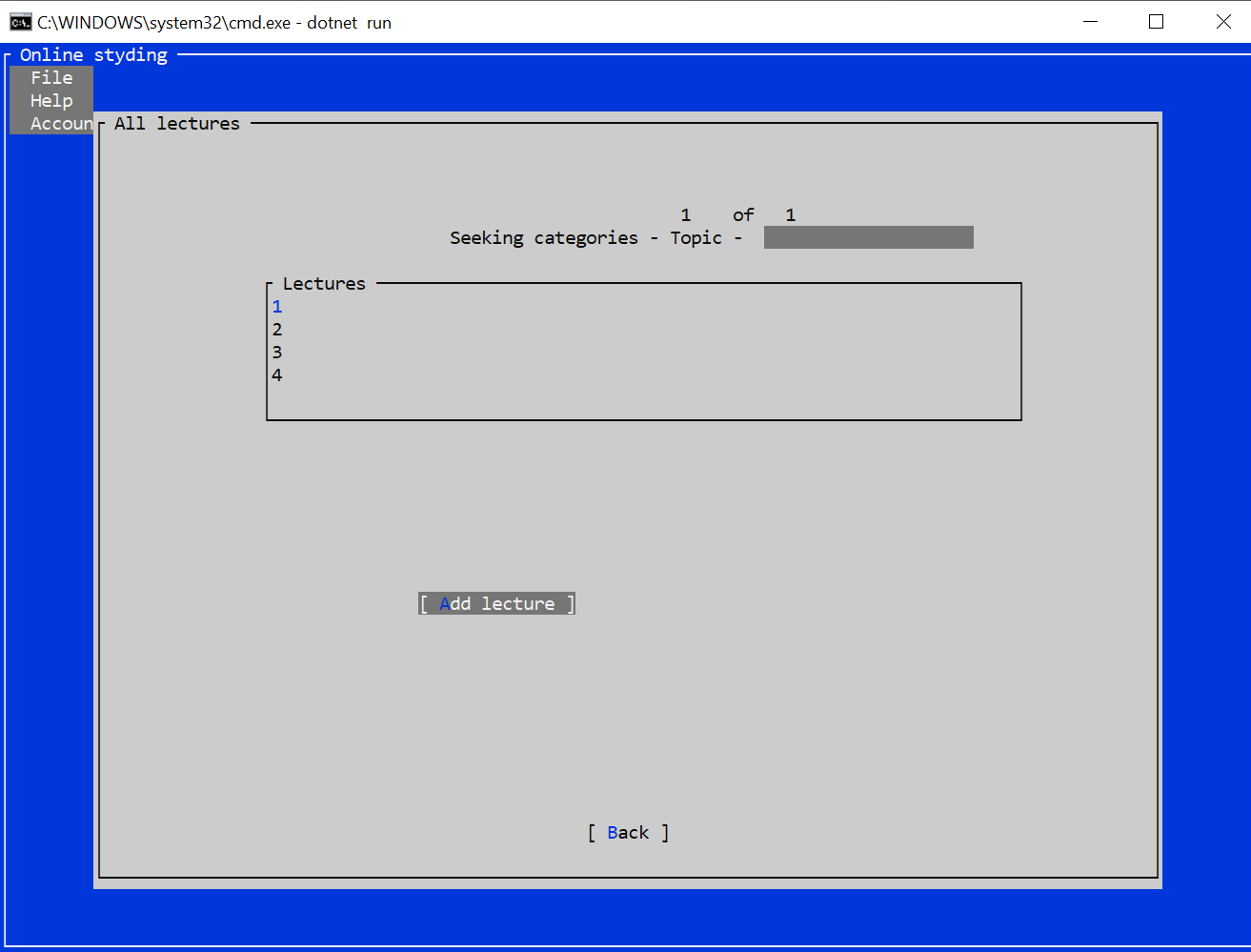


Рис. Вікно перегляду лекції при створенні курсу

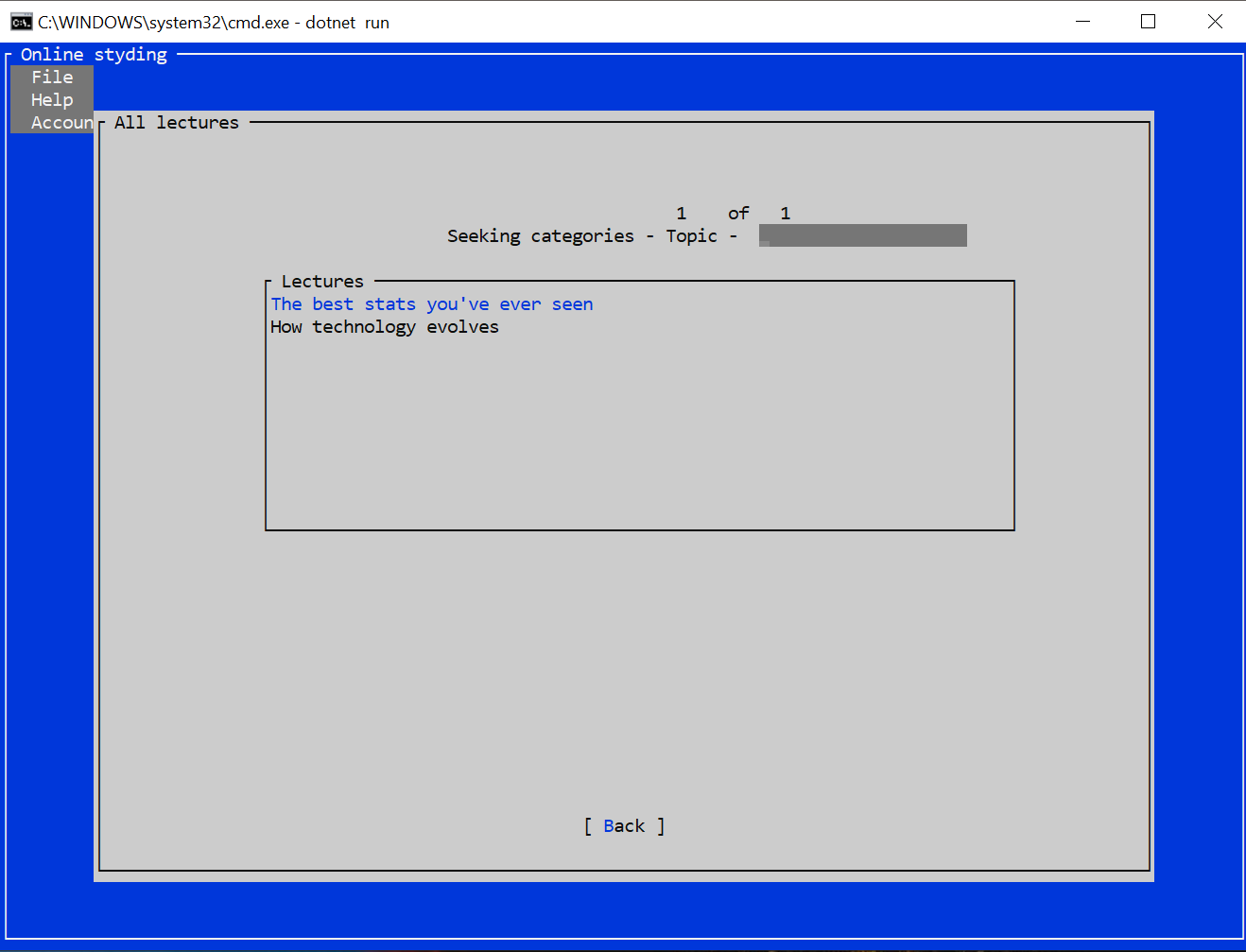


Рис. Вікно перегляду лекцій курсу

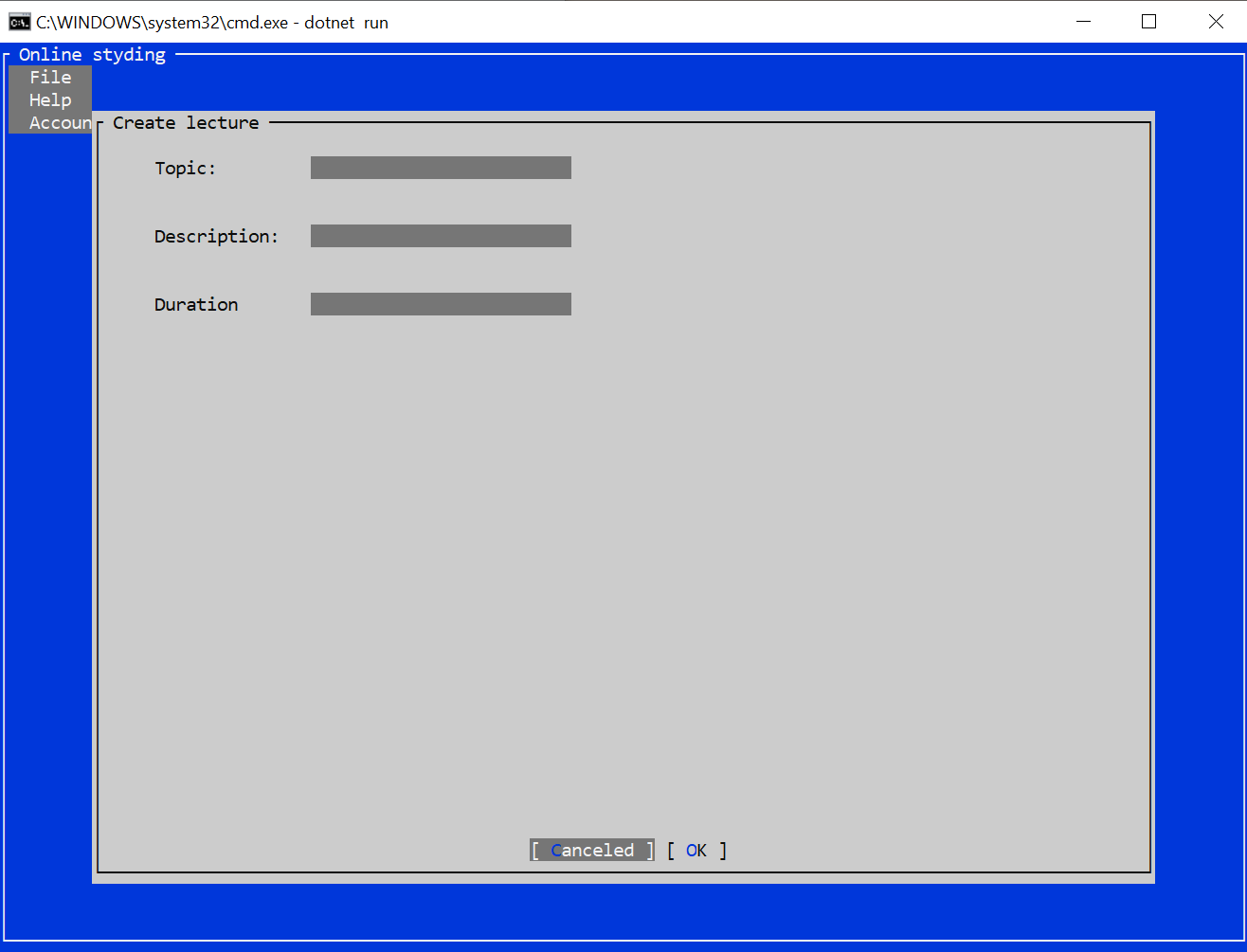


Рис. Вікно створення лекції

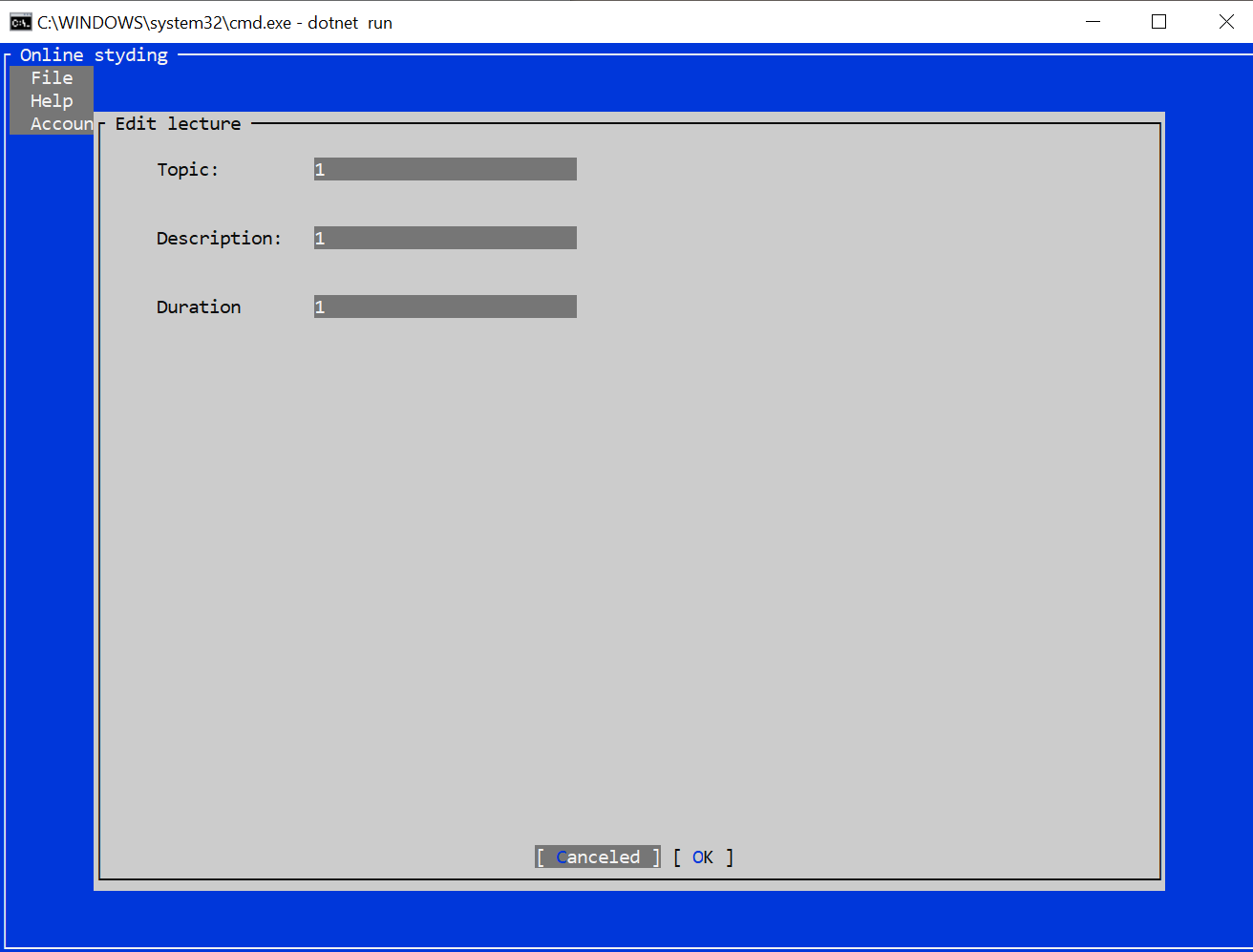


Рис. Вікно редагування лекції

# Висновки

Метою даного курсового проекту було розроблення десктопного додатку з подійно-орієнтованим інтерфейсом користувача для програмного додатку електронної бази навчальних курсів

Розроблена система:

* забезпечує авторизований доступ до управління електронною базою навчальних курсів;
* дозволяє перегляд,створення,редагування та видалення курсів та лекцій;
* дає можливість зареєструватися та увійти в додаток і використовувати його функціонал;

Особливу увагу під час розроблення даного програмного продукту було приділено графічному інтерфейсу користувача, взаємодії з базою даних та обробці помилок.  
Використання розробленого додатку дозволить стати підписником курсу, переглядати лекції, створювати, редагувати та видаляти власні курси та лекції.