

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ  
КЫРГЫЗСКО-ГЕРМАНСКОГО ИНСТИТУТА ПРИКЛАДНОЙ ИНФОРМАТИКИ**

# **ОТЧЕТ**

о производственной практике №1

Место прохождения практики: Департамент Развития УНПК(МУКР)

Выполнил:

студент группы АIN-2-21

Молчанов Кирилл Витальевич \_\_\_\_\_

Руководитель практики:

к.т.н., доцент Джалбиев Э.А. \_\_\_\_\_

**Бишкек 2025**

## Содержание

<b>Содержание.....</b>	<b>2</b>
<b>Введение.....</b>	<b>3</b>
Роль цифровизации в образовании.....	3
Развитие национальных образовательных систем.....	4
Соответствие современным требованиям:.....	4
<b>Основная часть.....</b>	<b>6</b>
Организационные вопросы охраны труда.....	6
Получение индивидуального задания.....	7
Выполнение индивидуального задания.....	7
Выполнение задания по теме ВКР.....	8
<b>Выводы.....</b>	<b>9</b>
<b>Приложение 1.....</b>	<b>10</b>
<b>Приложение 2.....</b>	<b>11</b>
<b>Приложение 3.....</b>	<b>12</b>
<b>Приложение 4.....</b>	<b>14</b>

## Введение

В ходе производственной практики была поставлена задача разработки серверной части Learning Management System (LMS) для Департамента развития НОУ УНПК Международного Университета Кыргызской Республики. Данная система направлена на автоматизацию управления образовательными процессами, оптимизацию взаимодействия между преподавателями и административным составом, а также управление доступом к образовательным материалам.

Актуальность исследования: Современные образовательные учреждения сталкиваются с рядом вызовов: необходимостью автоматизации управления учебным процессом, оптимизацией взаимодействия между преподавателями и студентами, а также анализом и мониторингом успеваемости. Разработка Learning Management System (LMS) отвечает этим вызовам, предлагая цифровые решения для повышения эффективности образовательного процесса.

- Роль цифровизации в образовании:

Как отмечается в отчёте ЮНЕСКО *"Education in a post-COVID world: Nine ideas for public action"*, цифровизация образовательных процессов стала ключевым элементом для адаптации системы образования к современным вызовам. LMS способствует обеспечению доступности учебных материалов и поддержке дистанционного обучения, что особенно актуально в условиях глобальных кризисов, таких как пандемия COVID-19.

<https://www.unesco.org/en/articles/education-post-covid-world-nine-ideas-public-action>

- Эффективность образовательных платформ:

В статье *"The Role of LMS in Enhancing Learning Outcomes"* исследуется, как внедрение LMS повышает эффективность преподавания и усвоения материала. Системы управления обучением

помогают структурировать курсы, обеспечивают обратную связь и автоматизируют рутинные задачи преподавателей.

<https://www.sciencedirect.com/science/article/abs/pii/S0360131519301369>

- Развитие национальных образовательных систем:

В Кыргызстане, как и в других странах Центральной Азии, внедрение LMS является важным шагом для модернизации образовательной системы. Исследование на примере внедрения платформы Moodle в Нарынском государственном университете показало, что использование LMS улучшает взаимодействие между студентами и преподавателями, повышает доступность образовательных ресурсов и способствует внедрению инновационных методов обучения.

<https://cyberleninka.ru/article/n/opyt-ispolzovaniya-lms-moodle-dlya-obucheniya-fizike>

- Соответствие современным требованиям:

Компания Codify Lab, разработавшая отечественную LMS, подчёркивает, что такие системы необходимы для управления учебным процессом в условиях цифровой трансформации. Их разработки используются в образовательных учреждениях Кыргызстана, подтверждая актуальность и востребованность подобных решений.

<https://lms.codifylab.com/>

#### Анализ международной и отечественной практики

На международной арене LMS широко используются для автоматизации образовательных процессов, таких как управление курсами, отслеживание успеваемости и взаимодействие с учащимися. Системы, подобные Moodle и Blackboard, предоставляют гибкие инструменты для администрирования и обучения.

В Кыргызстане использование LMS также становится все более актуальным. Например, Codify Lab разработала отечественную систему,

которая успешно применяется в образовательных учреждениях. Эти проекты доказывают эффективность автоматизации обучения в условиях локальных реалий.

Цель и задачи от руководителя практики

Целью производственной практики является разработка серверной части LMS с использованием фреймворка Flask, обеспечивающей:

- управление доступом к образовательным материалам,
- автоматизацию процесса выдачи заданий,
- интеграцию с базой данных для хранения информации о администрации и преподавателях.

Для достижения поставленной цели решались следующие задачи:

- Изучение технологий разработки серверных приложений на Flask.
- Создание архитектуры REST API для взаимодействия с клиентской частью.
- Обеспечение безопасности данных.
- Проведение тестирования и внедрение системы.

## **Основная часть**

### **Организационные вопросы охраны труда**

- Введение и инструктаж:
  - Проведен вводный инструктаж по охране труда, пожарной безопасности и технике безопасности на рабочем месте.
  - Ознакомление с инструкциями по охране труда и пожарной безопасности, а также с правилами внутреннего трудового распорядка на предприятии.
  - Изучены схемы аварийных выходов и расположение первичных средств пожаротушения.
- Изучение инструкций и нормативных документов:
  - Освоены нормы охраны труда при работе с вычислительной техникой.
  - Получены инструкции по технике безопасности при работе с оборудованием и программным обеспечением.

### **Знакомство с деятельностью предприятия**

- Характеристика предприятия:
  - Изучена структура организации, основные направления её деятельности, а также внутренние процессы управления образовательной деятельностью.
  - Проведен обзор выделенного рабочего места, технических средств и инфраструктуры.
- Технические и организационные аспекты:

- Проведено ознакомление с используемыми в департаменте техническими средствами, информационными системами и ПО.
- Изучены нормативные акты и инструкции, регламентирующие деятельность предприятия.

#### Получение индивидуального задания

- Описание задания:
  - Руководителем предприятия предоставлено индивидуальное задание, которое включало:
    - Разработку серверной части LMS.
    - Реализацию REST API для взаимодействия с базой данных.
    - Обеспечение авторизации и разграничения прав доступа.
  - Подготовлено детализированное техническое задание (ТЗ) в соответствии с требованиями предприятия.”Приложение 1”

#### Выполнение индивидуального задания

- Процесс выполнения:
  - На основе ТЗ разработана архитектура серверной части LMS с использованием фреймворка Flask.
  - Реализованы ключевые модули системы, включая:
    - Модуль авторизации (JWT-токены).”Приложение 2”
    - Модуль документооборота (загрузка и управление электронными документами).
    - Чат для взаимодействия между подразделениями (реализация через WebSocket).

- В качестве базы данных использована PostgreSQL. Созданы основные таблицы:
  - Пользователи (с разделением прав доступа).
  - Документы (с метаданными и историей изменений).
  - Сообщения чата.
- Меры по обеспечению безопасности:
  - Реализовано шифрование данных с использованием SSL/TLS.
  - Проведены меры по защите от SQL-инъекций и внедрению механизмов безопасной аутентификации.

#### Выполнение задания по теме ВКР

- Обзор литературы:
  - Проведен анализ международной и отечественной практики применения LMS:
    - Примеры реализации LMS (проект на GitHub, Moodle).
    - Опыт разработки отечественных систем (Codify Lab).
  - Изучены методические материалы по разработке серверных приложений на Flask.
- Практическая часть ВКР:
  - Проведено внедрение разработанных решений в инфраструктуру предприятия.
  - Реализована интеграция с клиентской частью системы и базой данных.



## **Выводы**

В ходе выполнения практики был проведен анализ международной и отечественной практики применения систем управления обучением (LMS), включая изучение успешных примеров, таких как Moodle и разработки отечественной компании Codify Lab. Литературный обзор показал важность внедрения LMS для автоматизации образовательных процессов, улучшения взаимодействия между преподавателями и студентами, а также для обеспечения доступности учебных материалов.

На основе полученных знаний была разработана серверная часть LMS для Международного Университета Кыргызской Республики с использованием Flask. В ходе реализации системы были созданы ключевые модули, такие как авторизация, документооборот и чат, что способствовало улучшению эффективности образовательного процесса. Реализованные решения полностью соответствуют современным требованиям цифровизации образования и обеспечивают удобство в управлении учебным процессом.

Дата сдачи отчета: \_\_\_\_\_

Подпись автора: \_\_\_\_\_

## Приложение 1

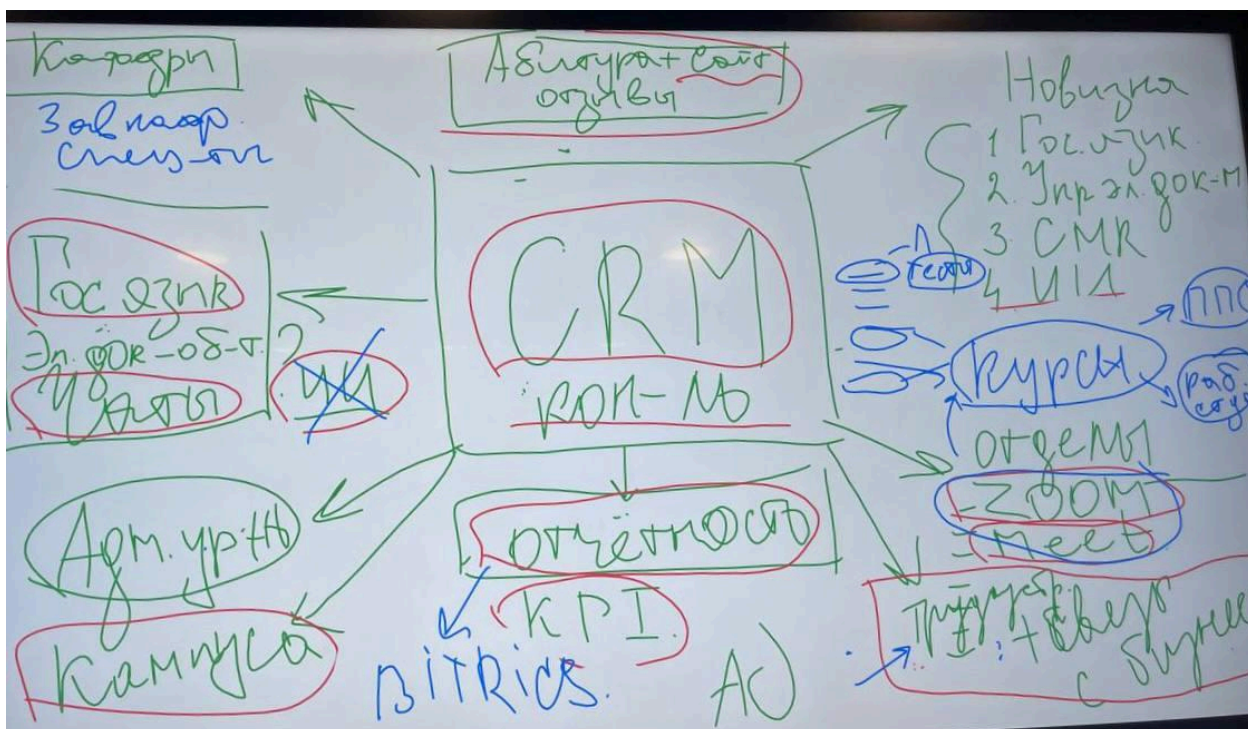


Рис. 1 - Иллюстрационное техническое задание

## Приложение 2

```
1 auth_bp.route('/login', methods=['GET', 'POST'])
2 def login():
3     if request.method == 'POST':
4         username = request.form['username']
5         password = request.form['password']
6
7         with sqlite3.connect('users.db') as con:
8             cur = con.cursor()
9             cur.execute("SELECT password FROM users WHERE username = ?", (username,))
10            user = cur.fetchone()
11
12            if user and check_password_hash(user[0], password):
13                session['username'] = username
14                flash('Вы успешно вошли!', 'success')
15                return redirect(url_for('auth.protected'))
16            else:
17                flash('Неверное имя пользователя или пароль!', 'danger')
18
19        return render_template('login.html')
```

Рис. 2.1 - Авторизация пользователя

```
1 @auth_bp.route('/register', methods=['GET', 'POST'])
2 def register():
3     if request.method == 'POST':
4         username = request.form['username']
5         password = request.form['password']
6         hashed_password = generate_password_hash(password)
7
8         try:
9             with sqlite3.connect('users.db') as con:
10                cur = con.cursor()
11                cur.execute("INSERT INTO users (username, password) VALUES (?, ?)", (username, hashed_password))
12                con.commit()
13                flash('Регистрация прошла успешно! Войдите в систему.', 'success')
14                return redirect(url_for('auth.login'))
15            except sqlite3.IntegrityError:
16                flash('Имя пользователя уже занято!', 'danger')
17
18        return render_template('register.html')
```

Рис. 2.2 - Регистрация пользователя

```
1 @auth_bp.route('/logout')
2 def logout():
3     session.pop('username', None)
4     flash('Вы вышли из системы.', 'success')
5     return redirect(url_for('auth.login'))
```

Рис. 2.3 - функция выхода из аккаунта


## Приложение 3

```
1 @documents_bp.route('/translate_upload', methods=['GET', 'POST'])
2 async def translate_upload():
3     if request.method == 'POST':
4         if 'file' not in request.files:
5             return redirect(request.url)
6
7         file = request.files['file']
8         translation_direction = request.form.get('direction', 'ky-ru')
9
10        if file and allowed_file(file.filename) and file.filename.endswith('.txt'):
11            filename = file.filename
12            filepath = os.path.join(UPLOAD_FOLDER, filename)
13            file.save(filepath)
14            translated_content = await translate_file(filepath, direction=translation_direction)
15            if translated_content:
16                translated_filename = f"translated_{filename}"
17                translated_filepath = os.path.join(UPLOAD_FOLDER, translated_filename)
18                with open(translated_filepath, 'w', encoding='utf-8') as f:
19                    f.write(translated_content)
20                return redirect(url_for('documents.view_file', filename=translated_filename))
21            else:
22                return f"Translation failed: {translated_content}", 500
23        else:
24            return "Only text files (.txt) are supported for translation", 400
25
26    return render_template('translate_upload.html')
```

Рис. 3.1 - функция загрузки и перевода файла между языками


```
1 @documents_bp.route('/')
2 def documents():
3     documents = os.listdir(UPLOAD_FOLDER)
4     print(f"Documents found: {documents}")
5     if not documents:
6         documents = ["No documents available. Upload a new file!"]
7     return render_template('documents.html', documents=documents)
8 @documents_bp.route('/upload', methods=['POST'])
9 def upload_file():
10    if 'file' not in request.files:
11        return redirect(request.url)
12    file = request.files['file']
13    if file and allowed_file(file.filename):
14        filename = file.filename
15        filepath = os.path.join(UPLOAD_FOLDER, filename)
16        file.save(filepath)
17        print(f"File uploaded: {filename}")
18        return redirect(url_for('documents.documents'))
19
20    return redirect(url_for('documents.documents'))
```

Рис. 3.2 - функция проверки на наличие файлов в системе и отображения в списке




```
1 @documents_bp.route('/delete/<filename>', methods=['POST'])
2 def delete_file(filename):
3     try:
4         filepath = os.path.join(UPLOAD_FOLDER, filename)
5         os.remove(filepath)
6         print(f"File deleted: {filename}")
7         return redirect(url_for('documents.documents'))
8     except FileNotFoundError:
9         return "File not found", 404
```

Рис. 3.3 - функция удаления файлов



```
1 @documents_bp.route('/view/<filename>')
2 def view_file(filename):
3     return send_from_directory(UPLOAD_FOLDER, filename)
```

Рис. 3.4 - функция просмотра загруженного файлов



```
1 @documents_bp.route('/download/<filename>')
2 def download_file(filename):
3     return send_from_directory(UPLOAD_FOLDER, filename, as_attachment=True)
```

Рис. 3.5 - функция загрузки загруженного файлов

## Приложение 4

```
1 @chat_bp.route('/', methods=['GET', 'POST'])
2 def chat():
3     if request.method == 'POST':
4         username = request.form['username']
5         room = request.form['room']
6         add_user(username, room)
7         session['username'] = username
8         session['room'] = room
9         return redirect('/chat')
10
11     users = get_users()
12     rooms = get_rooms()
13     return render_template('chat.html', users=users, rooms=rooms)
```


Рис. 4.1 - функция чата между пользователями

```
1 @chat_bp.route('/leave', methods=['POST'])
2 def leave_room_route():
3     username = session.get('username')
4     if username:
5         remove_user(username)
6         session.pop('username', None)
7         session.pop('room', None)
8     return redirect('/chat')
```

Рис. 4.2 - функция чата между пользователями

```
1 @chat_bp.route('/room/<room_name>', methods=['GET'])
2 def room_history(room_name):
3     messages = get_messages(room_name)
4     return render_template('room_history.html', room_name=room_name, messages=messages)
```

Рис. 4.3 - функция истории чата между пользователями



```
1 @socketio.on('join_room')
2 def handle_join(data):
3     username = session.get('username')
4     if username:
5         room = data['room']
6         join_room(room)
7         print(f"{username} has joined room {room}")
```

Рис. 4.3 - функция входа в чат между пользователями