**Введение**

Первый раздел носит название «Анализ задачи». В нем вы сможете ознакомиться с постановкой задачи, включающей исследование предметной области. Также в разделе будут описаны входные данные (например, логин пользователя, пароль пользователя) и выходные данные (учебный материал, данные об успеваемости). В подразделе «Инструменты разработки» будет рассмотрена выбранная среда программирования.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки программы. Описывается структура хранения данных (например, база данных сайта) и способы их обработки. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе будет описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации.

«Реализация» – это третий раздел отчета, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения.

Четвертый раздел – «Тестирование». В этом разделе будет описано тестирование функционала программы. Проверка охватит ввод, обработку и вывод данных, а также корректность выполнения пользовательских операций. Будут смоделированы все возможные действия пользователя при работе с web-ресурсом, начиная от входа на сайт заканчивая закрытием вкладки.

В разделе «Руководство пользователя» будет описано назначение, область применения, среда функционирования данного программного продукта.

«Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств.

В разделе «Список использованных источников» будет приведен список источников, использованных при разработке программы.

В приложении А будет представлена диаграмма вариантов использования.

В приложении Б будет представлена структура сайта.

В приложении В будет представлена модель данных.

В приложении Г будет представлена функциональная модель.

В приложении Д будет представлена диаграмма деятельности.

В приложении Е будет представлена диаграмма классов.

В приложении Ж будет представлена диаграмма объектов.

В приложении З будет представлена диаграмма последовательности.

В приложении И будут представлены UX и UI прототипы.

В приложении К будут представлены тест-кейсы.

1. **Анализ задачи**
   1. **Постановка задачи**
      1. **Организационно-экономическая сущность задачи**

Наименование задачи**:** сайт Гродненского Государственного Политехнического колледжа.

Цель разработки**:** создание новой версии сайта Гродненского Государственного Политехнического колледжа.

Назначение**:** данный программный продукт предназначен для людей любого возраста с целью получения информации о колледже.

Периодичность использования**:** по мере необходимости.

Источники и способы получения данных**:** Сайт GGPK.by, сотрудники учреждения.

Обзор существующих аналогичных ПП**:** рассмотрим сайт GGPK.by на базе которого будет разрабатываться новая версия. На этом сайте устарел дизайн, интерфейс, отсутствует авторизация для пользователя. Отсутствует среда для взаимодействия учащихся с преподавателями. Модуль «Расписание занятий» требует полного изменения.

* + 1. **Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП**:**

Гость:

1. Просмотр страниц сайта;
2. Просмотр информации о колледже;
3. Местоположение колледжа;
4. Переход в другие соц. сети такие как Instagram, Вконтакте;
5. Авторизация как пользователя;
6. Связь с администрацией;
7. График работы;
8. Блог приёмной комиссии;
9. Расписание занятий.

Пользователь: может всё тоже самое что и гость, а также появляется:

Учащийся:

1. Просмотр личных оценок;
2. Просмотр личной успеваемости;
3. Доступ к системе дистанционного обучения;
4. Получение учебного материала.

Преподаватель:

1. Выставление оценок учащимся;
2. Ведение ведомости у учащихся;
3. Доступ к системе дистанционного обучения;
4. Добавление учебной информации;

Администратор:

1. Доступ к админ. панеле;
2. Редактирование страниц сайта;
3. Доступ к базам данных;
4. Доступ к аккаунтам пользователей;
5. Изменение расписания занятий;
6. Создание аккаунтов пользователей.
   * 1. **Описание процессов с входной, выходной и условно-постоянно информацией**

Описание исходной(входной) информации:

1. Логин пользователя
2. Пароль пользователя

Описание результатной (выходной) информации:

1. Учебный материал
2. Данные об успеваемости

Описание используемой условно-постоянной информации

1. Контакты
2. Расписание занятий
3. Информация о колледже
4. Фотогалерея
5. Карта сайта
   * 1. **Нефункциональные (эксплуатационные) требования**

* Требования к применению:помогает узнать всю необходимую информацию об колледже для посетителей сайта, а также взаимодействовать преподавателям с учащимися;
* Требования к реализации:для реализации статических страниц и шаблонов должна использоваться система управления содержимым сайта WordPress;
* Требования к надёжности: система может быть недоступна не более чем 24 часа в год. У администратора сайта должна быть возможность выгрузить и загрузить копию сайта;
* Требования к интерфейсу:при разработке сайта должны быть использованы преимущественного белого цвета с оттенками синего. Основные разделы сайта должны быть доступны с первой страницы. Грамотный пользовательский интерфейс. Сайт должен адаптироваться под компьютер, телефон и планшет.
  1. **Диаграмма вариантов использования**

В качестве актеров представлен гость, учащийся, преподаватель и администратор которые взаимодействует с системой.

Гость может:

* Просматривать страницы сайта;
* Переходить в другие соц. сети;
* Связаться с администрацией;
* Авторизоваться как пользователь;
* Просмотреть местоположение колледжа;
* Написать обращение.

Учащийся может:

* Всё тоже что и гость;
* Иметь доступ к системе дистанционного обучения.

Преподаватель может:

* Всё тоже что гость и учащийся;
* Иметь доступ к системе дистанционного обучения.

Разработанная диаграмма вариантов использования представлена в приложении А.

* 1. **Выбор стратегии разработки и модели жизненного цикла**

Для разработки веб-ресурса «сайт Гродненского Государственного политехнического колледжа» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 1 **–** Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории требований** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да |  |  |  |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да |  |  |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? |  |  |  |  | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет |  | Нет |  |  |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет |  | Нет |  |  |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? |  |  |  | Да | Да | Да |

Продолжение таблицы 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории требований** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? |  |  | Да | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V- образную, 3 за RAD, 5 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 3 подходящей является инкрементная модель.

Таблица 2 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории команды разработчиков**  **проекта** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет |  |  |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? |  |  | Нет | Нет | Нет |  |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет |  |  |  |

Продолжение таблицы 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории требований** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да |  | Да |  |  |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? |  |  |  |  | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да |  | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V-образную, 3 за RAD, 4 за инкрементную, 3 за быстрого прототипирования и 2 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящими являются каскадная, V-образная и инкрементная модели.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории коллектива пользователей** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да |  | Да |  | Да |

Продолжение таблицы 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории требований** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? |  |  |  | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет |  | Нет |  | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? |  |  |  |  | Да | Да |

Вычисления: 2 за каскадную, 2 за V-образную, 0 за RAD, 3 за инкрементную, 2 за быстрого прототипирования и 4 за эволюционную.

Итог: На основе результатов заполнения табл. 5 подходящей является эволюционная модель.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории типов проекта и рисков** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет |  |  |  |

Продолжение таблицы 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ критерия** | **Критерии категории типов проекта и рисков** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипирования** | **Эволюционная** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да |  |  |
| 3. | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет |  |  |  |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да |  | Да |  | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? |  | Да |  | Да |  | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? |  |  |  | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? |  |  |  | Да | Да | Да |
| 8. | Является ли график сжатым? | Нет | Нет |  |  |  |  |
| 9. | Предполагается ли повторное использование компонентов? |  |  | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? |  |  |  |  | Да | Да |

Вычисления: 5 за каскадную, 6 за V-образную, 4 за RAD, 6 за инкрементную, 4 за быстрого прототипирования и 6 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является инкрементная, V-образная и эволюционная модели.

Общий итог: в итоге заполнения табл. 3 – 6 наиболее подходящей является инкрементная модель

**1.4 Инструменты разработки**

Для разработки и настройки новой версии сайта гродненского государственного политехнического колледжа, мы используем WordPress, Moodle.

Разработка будет осуществляться с использованием конструкторов:

– Moodle: система управления обучением (LMS), предназначенная для создания и управления учебными курсами и образовательным контентом. Moodle специально разработан для образовательных учреждений и организаций, предлагая функции для управления курсами, заданиями и оценками. Moodle позволяет создавать и управлять большим количеством курсов и пользователей, что идеально подходит для учебных заведений и крупных организаций. Платформа включает инструменты для форумов, чатов, тестов и оценок, что позволяет создать полноценное образовательное пространство. Moodle поддерживает интеграции с различными внешними сервисами и системами, что расширяет его функциональность.

Иные инструменты, которые будут использованы при разработке и оформлении проекта:

– Microsoft Office Word: для подготовки и написания документации проекта;

– DRAW.IO: для создания структурной схемы сайта и навигации;

– Microsoft Office Visio: для создания диаграмм;

– Google Chrome: для тестирования работы сайта в разных браузерах.

Простой сайт не требует высокой производительности, поэтому работа будет осуществляться на компьютере со следующими характеристиками:

– Процессор: Intel® Core I5-11400H 2.7GHz;

– Оперативная память: 16.00 GB;

– Жёсткий диск: 500 GB;

– Видеокарта: NVIDIA® RTX 3050 4GB laptop version;

– Операционная система: Windows 10 Pro.

**2 Проектирование**

**2.1 Разработка системы меню и навигации по программе**

Система меню и навигации является важным компонентом пользовательского интерфейса, обеспечивающим удобный доступ к функциональности и разделам приложения. Ее главная задача — помочь пользователю быстро находить необходимую информацию и перемещаться между различными страницами. Для этого меню должно быть интуитивным, доступным и логичным.

При разработке такой системы необходимо учитывать адаптивный дизайн, чтобы меню корректно отображалось на любых экранах. Также важно документировать структуру и работу навигации, чтобы пользователи могли быстро освоиться. Хорошо продуманная система меню способствует повышению эффективности работы с приложением и улучшению общего пользовательского опыта.

Полную структуру сайта можно рассмотреть в приложении Б.

**2.2 Разработка UML-диаграмм**

**2.2.1 Модель данных**

Модель данных — это логическая структура, описывающая, как данные хранятся, организованы и связаны в рамках информационной системы. Она определяет основные сущности, их атрибуты, взаимосвязи и ограничения, обеспечивая основу для обработки данных. Использование модели данных помогает эффективно управлять информацией, гарантировать целостность данных и упрощать работу с ними в процессе разработки и эксплуатации программного обеспечения.

Разработанная модель данных представлена в приложении В.

**2.2.2 Функциональная модель**

Функциональная модель — это описание процессов и функций системы, показывающее, как она обрабатывает данные и взаимодействует с внешней средой. Она определяет, какие операции выполняются, их последовательность, входные и выходные данные, а также связь между различными функциями. Функциональная модель используется для анализа и проектирования системы, чтобы понять, как она должна работать и какие задачи выполнять для достижения поставленных целей.

В функциональной модели изображен процесс добавления нового курса на сайт.

С функциональной моделью можно ознакомиться в приложении Г.

**2.2.3 Диаграмма деятельности**

Диаграмма деятельности — это графическое представление последовательности действий, потоков управления и данных в процессе или системе. Она показывает, как выполняются задачи, переходы между действиями и возможные параллельные процессы. Диаграмма деятельности помогает визуализировать логику работы системы, выявлять точки взаимодействия и оптимизировать процессы. Чаще всего используется при моделировании бизнес-процессов или проектировании программного обеспечения.

На диаграмме изображено добавление редактирование курсов на сайте.

Диаграмма изображена в приложении Д.

**2.2.4 Диаграмма классов**

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними.

На диаграмме показаны используемые в программе классы.

С диаграммой можно ознакомиться в приложении Е.

**2.2.5 Диаграмма объектов**

Диаграмма объектов — это графическое представление состояния системы в конкретный момент времени, показывающее объекты и их взаимосвязи. Она фокусируется на экземплярах классов, их атрибутах и связях, а не на общих структурах. Диаграмма объектов помогает понять, как данные взаимодействуют в определенной ситуации или этапе работы системы, и часто используется для уточнения и проверки моделей данных или диаграмм классов.

На диаграмме показано взаимодействие администратора с панелью управления пользователями.

Диаграмма объектов показана в приложении Ж.

**2.2.6 Диаграмма последовательности**

Диаграмма последовательности используется для визуализации взаимодействия между объектами в системе. Она показывает порядок сообщений, которыми объекты обмениваются друг с другом во времени и их жизненный цикл. Это самый простой и удобный инструмент для демонстрации всех интеграций и взаимодействий в рамках проектируемого бизнес-процесса.

На диаграмме показана последовательность действий прохождения курса.

Диаграмма представлена в приложении З.

**2.3 Разработка пользовательского интерфейса**

Одним из важнейших элементов проектирования является описание внешнего интерфейса разрабатываемого программного продукта.

Для обеспечения эффективной работы пользователя нужно создать целостное приложение, охватывающее конкретную предметную область, где все компоненты будут сгруппированы по их функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс. Приложение должно позволять пользователю решать задачи с минимальными усилиями по сравнению с работой с разрозненными объектами.

Прототип – это модель пользовательского интерфейса. В сущности, это «черновик», созданный на основе представления разработчика о потребностях пользователя.

В создании интерфейса был задействован шрифт Inter. Для текста были использованы черные, серые и белые цвета. Для создания цветовой стилистики основными цветами были выбраны: белый, синий, и серый. Логотип сайта был взят со старой версии сайта Гродненского государственного политехнического колледжа.

Прототипы UX и UI представлены в приложении И.

Со всеми разработанными UX-прототипами можно ознакомиться по ссылке:

<https://www.figma.com/design/MQjJZLYuHbHysX9JJZXdqT/UX-SDS?node-id=0-1&t=T0JIwNa0vaewgRh7-1>.

Со всеми разработанными UI-прототипами можно ознакомиться по ссылке:

<https://www.figma.com/design/WZZqcVUXF1OOtBRwaF78H0/UI-SDS?node-id=0-1&t=L42isETm10GyO2E7-1>.

**3 Реализация**

**3.1 Руководство программиста**

Программный продукт разработан с использованием системы управления содержимым сайта Moodle. Этот ресурс представляет собой удобный и простой в использовании инструмент, позволяющий создавать сайта сразу с готовой системой дистанционного обучения. Для начала работы необходимо установить проверенную версию Moodle, настроить администратора, параметры сервера и безопасность. Далее добавляются курсы, материалы (документы, видео, презентации) и пользователи с разделением на группы (администраторы, преподаватели, студенты). Платформа поддерживает установку плагинов для тестирования, общения и других задач. После настройки система готова к использованию и легко адаптируется под любые учебные или корпоративные потребности.

**3.1.1 Создание сайта**

После запуска Moodle создать учетную запись администратора, настроить основные параметры сайта (название, описание, язык интерфейса). Создать структуру курсов, загрузить учебный материал (документы, презентации, видео) и настроить интерактивные элементы (тесты, форумы, чаты). Данные для входа на сайт администратора:

– E-mail: evgenniydasshkevich@gmail.com;

– Пароль: D!k@@L#R\_6Cij\*2.

После авторизации попадаем на сайт.

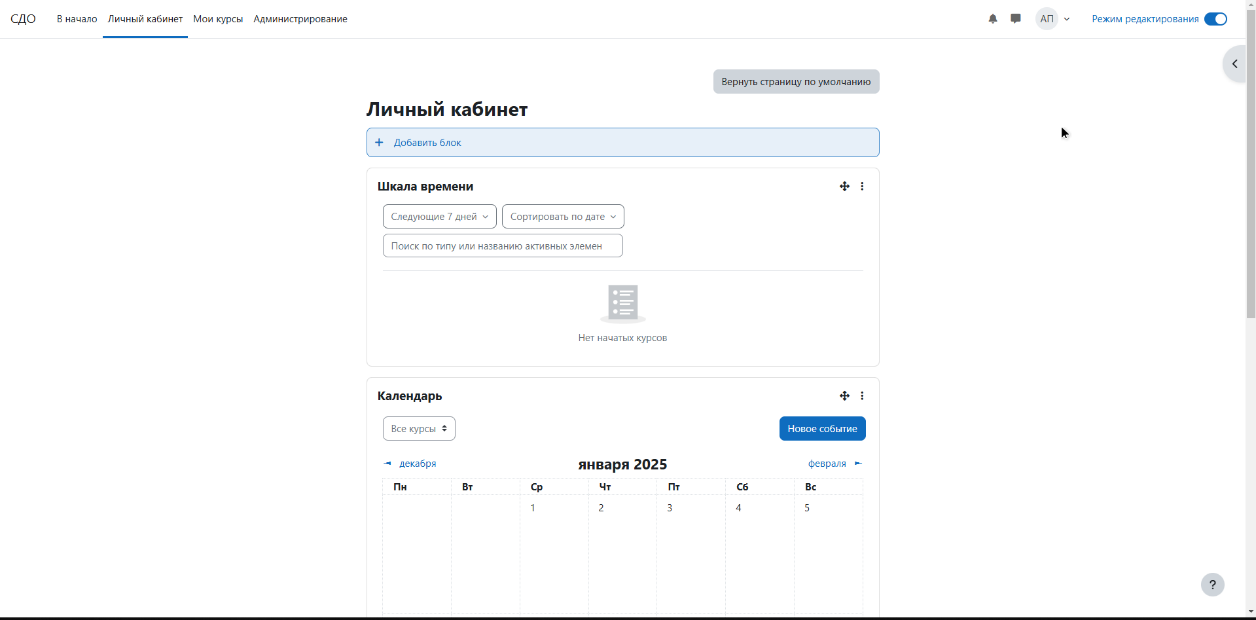
Меню сайта показана на рисунке 1.

Рисунок 1 – Меню сайта

**3.1.2 Настройка функциональности и дизайна**

Все элементы, которые имеются на экране, могут быть изменены, перемещены или удалены.

Moodle дает множество тем для создания сайта, которые могут быть отредактированы (изменена тема, поля ввода и т.д), дополнены новыми элементами и удалены. На рисунке 1 можно рассмотреть меню сайта.

Меню состоит из 4-ех пунктов:

– «В начало» – просмотр и создание новых курсов;

– «Личный кабинет» – позволяет просмотреть информацию, связанная с их учебной или административной деятельностью;

– «Мои курсы» – позволяет просмотреть все курсы, к которым у него есть доступ;

– «Администрирование» – позволяет добавлять, удалять и редактировать сайт.

**3.1.3 Редактирование страниц**

Для создания курса необходимо в меню найти страницу «Мои курсы» далее нажать на кнопку «Создать курс».

Панель редактирования состоит из вкладок:

– «Категория» – редактирование названия курса, дата начала и окончания курса, описания, формата курса, внешний вид, файлы и загрузки, отслеживание выполнения, группы, теги;

– «Настройки» – редактирование названия категории и описании;

– «Загрузка курсов» – редактирование и загрузка файлов для курса;

– «Дополнительно» – редактирование прав, глобальных групп, фильтров, шаблонов учебного плана и банк контента.

На рисунке 2 представлена панель редактирование курса.

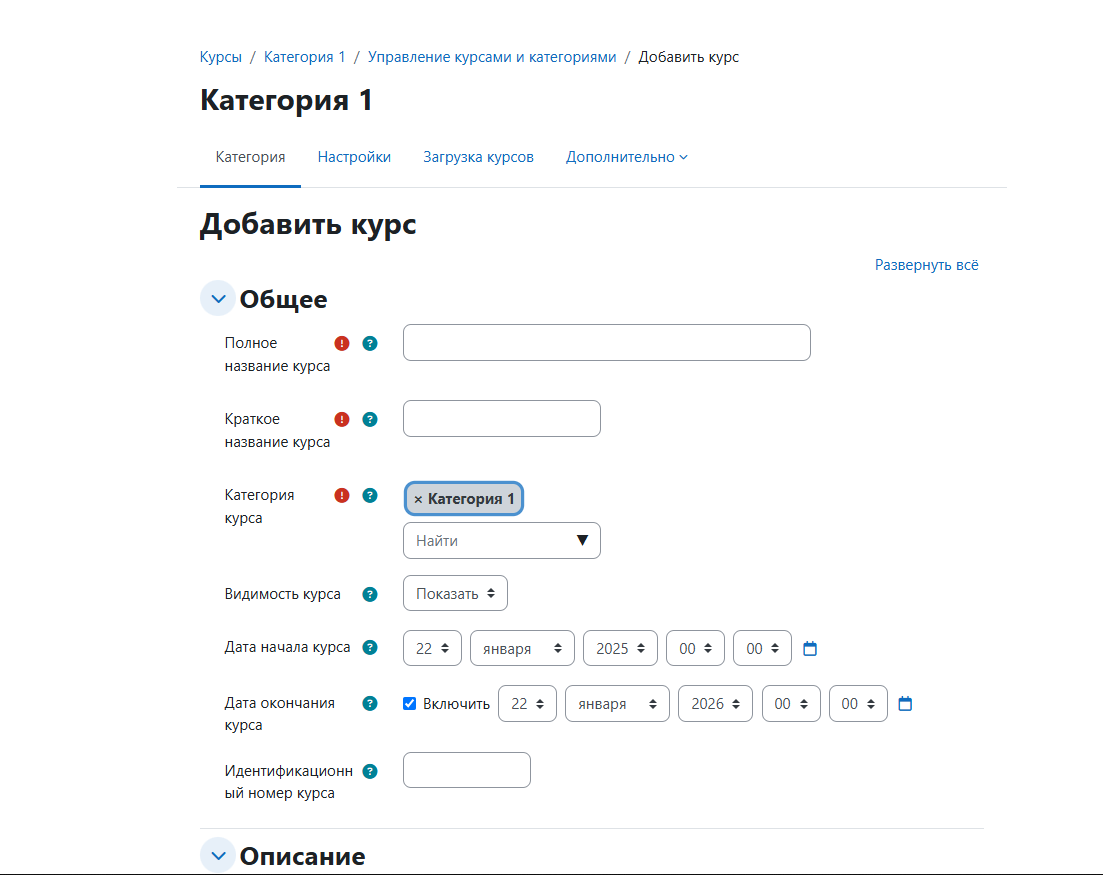
Для редактирования структуры курса необходимо перейти на вкладку «Мои курсы» далее нажать на курс, который хотим редактировать и включить режим «Режим редактирования» который находится в правом верхнем уголке сайта.

Рисунок 2 – Панель редактирования курса

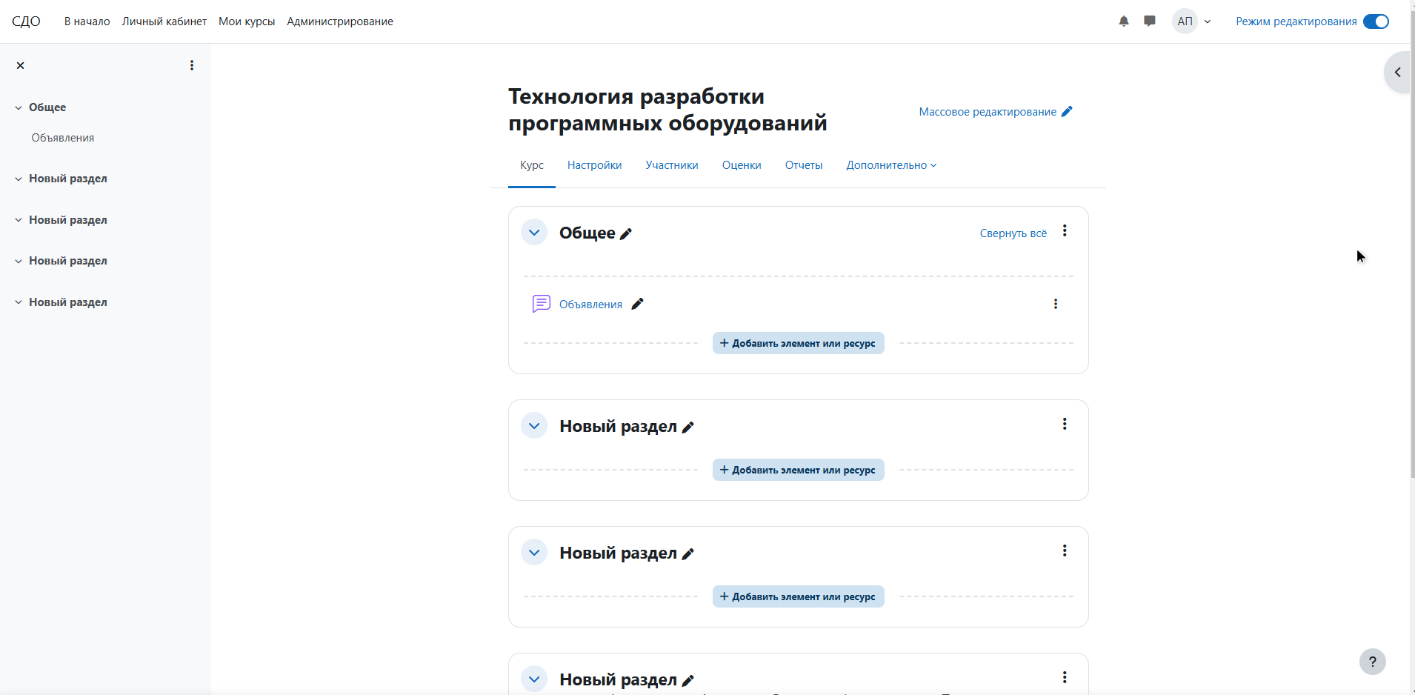
На рисунке 3 представления панель редактирования структуры курса.

Рисунок 3 – Панель редактирования структуры курса

Также можно добавлять или редактировать разделы курса. На рисунке 3 можно увидеть с левой стороны сайта панель редактирования разделов.

**4 Тестирование**

**4.1 Тесты на использование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Разработанные тест-кейсы и статус их выполнения представлены в приложение К.

Расписание работ над проектом представляю в таблице 5.

Таблица 5 – Расписание работы над проектом

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Дата | Деятельность | Продолжительность, ч |
| Дашкевич Евгений | 06.12.2024 | Разработка тестов | 2 |
| Дашкевич Евгений | 06.12.2024 | Тестирование программы | 2 |
| Дашкевич Евгений | 08.11.2024 | Составление отчетов о найденных дефектах | 1 |
| Дашкевич Евгений | 09.11.2024 | Исправление найденных ошибок | 3 |
| Дашкевич Евгений | 11.11.2024 | Проведение регрессионного тестирования | 2 |
| Дашкевич Евгений | 11.11.2024 | Составление отчета о результатах тестирования | 2 |

**4.2 Отчёт о результатах тестирования**

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

Статистика по всем дефектам представлена в таблице 6.

Таблица 6 – Статистика по всем дефектам

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Важность | | | |
| Статус | Количество | Низкая | Средняя | Высокая | Критическая |
| Найдено | 0 | 0 | 0 | 0 | 0 |
| Исправлено | 0 | 0 | 0 | 0 | 0 |
| Проверено | 0 | 0 | 0 | 0 | 0 |
| Открыто заново | 0 | 0 | 0 | 0 | 0 |
| Отклонено | 0 | 0 | 0 | 0 | 0 |

**5 Руководство пользователя**

Цель данного проекта является разработка системы дистанционного обучения для сайта «Гродненский Государственный Политехнический Колледж», который позволит пользователям получить доступ к просматриванию и изучению курса.

Для того, чтобы открыть сайт необходимо перейти по ссылке.

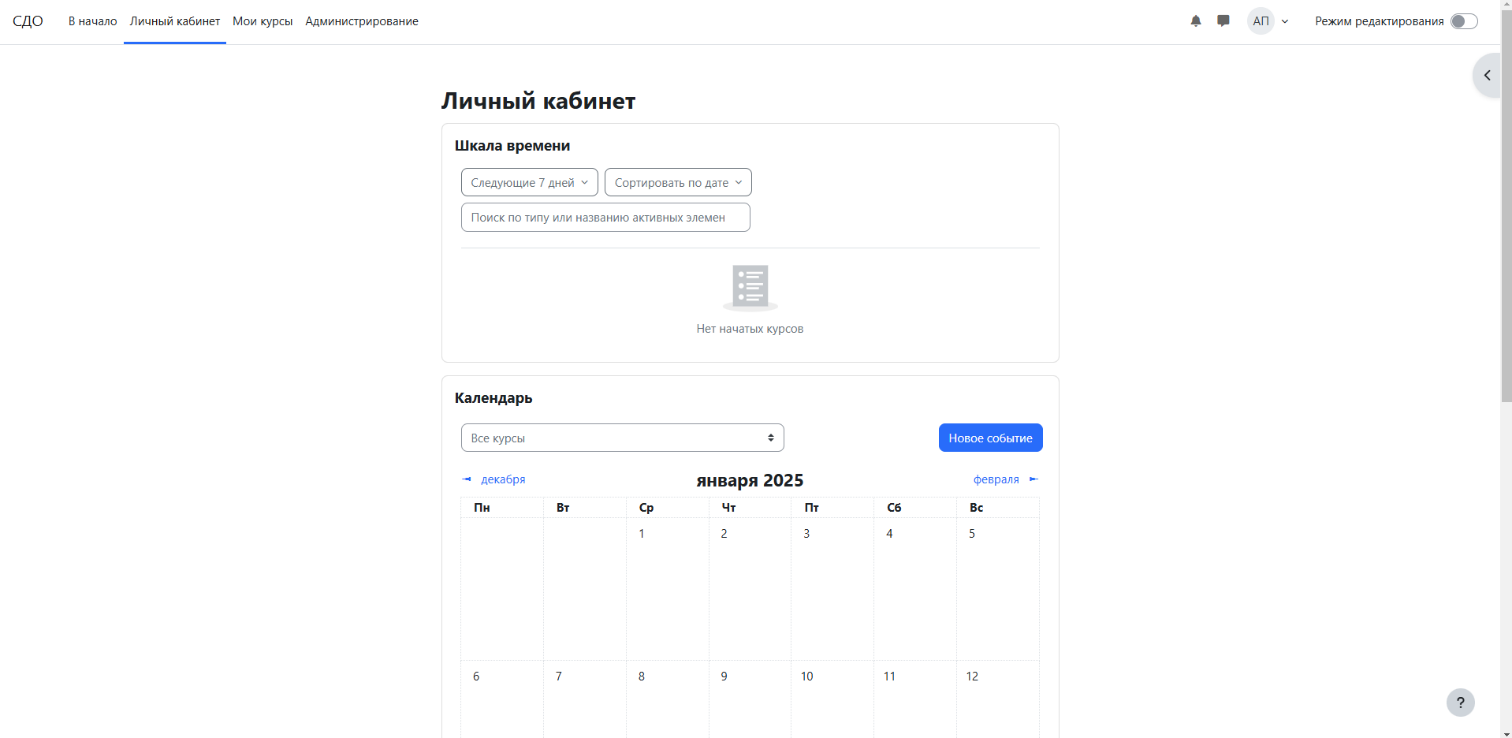
После открытия сайта загружается главная страница, представленная на рисунке 4.

Рисунок 4 – Главная страница

Переход по страницам можно осуществлять по горизонтальному меню.

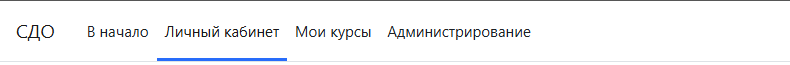
Header сайта представлена на рисунке 5.

Рисунок 5 – Меню сайта

Страница «В начало» включает в себя доступные курсы, которые можно зайти и узнать информацию.

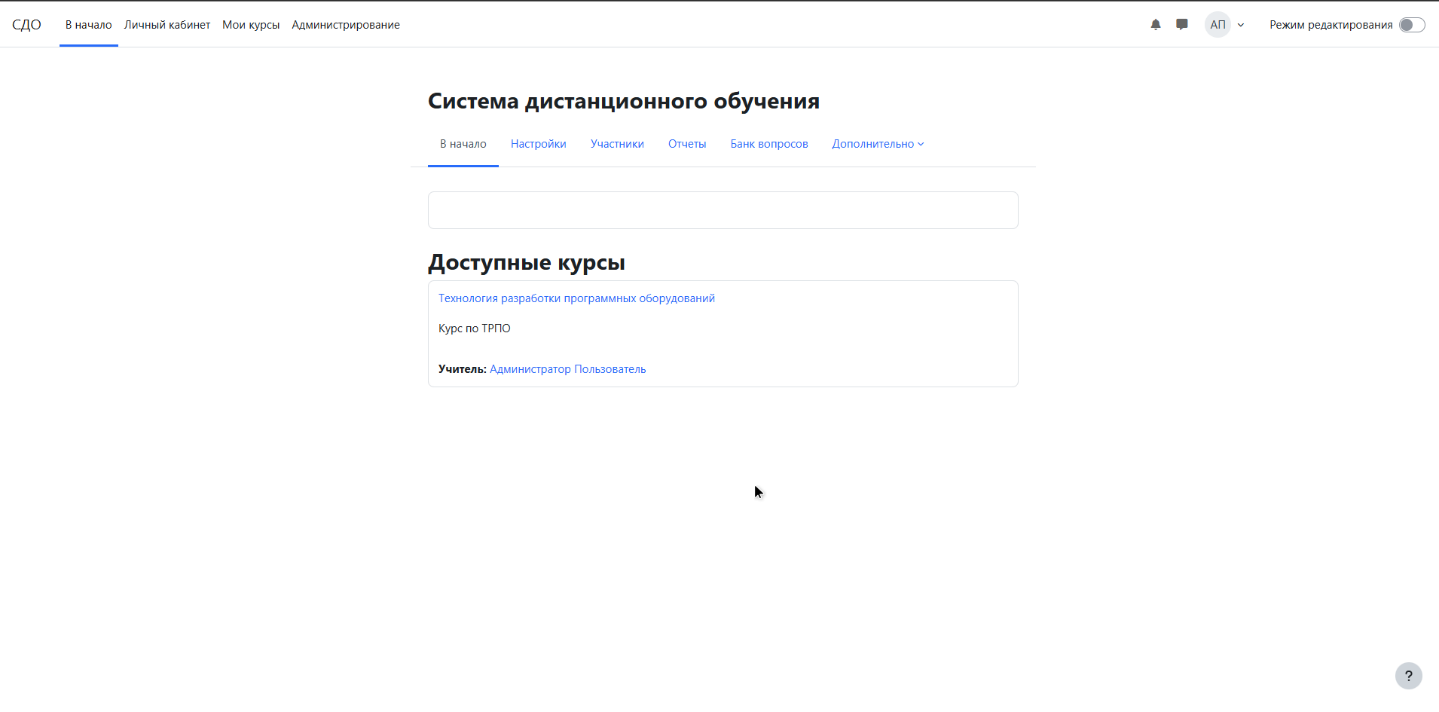
Страница «В начало» представлена на рисунке 6.

Рисунок 6 – Страница «В начало»

Страница «Личный кабинет» включает в себя просмотр начатых курсов, которые можно зайти и узнать информацию.

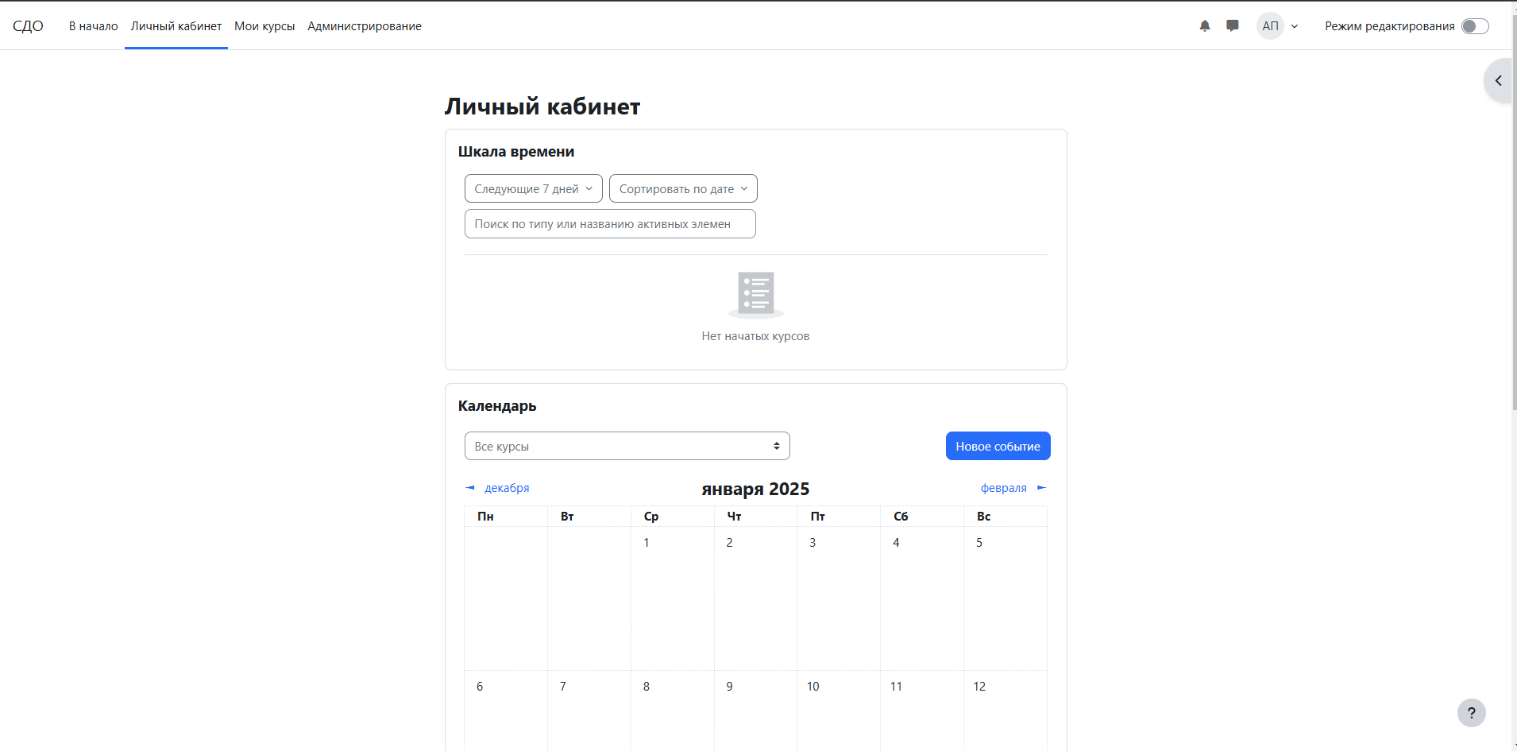
Страница «Личный кабинет» представлена на рисунке 7.

Рисунок 7 – Страница «Личный кабинет»

Страница «Мои курсы» включает в себя курсы, которые уже существуют или возможность создать новый курс.

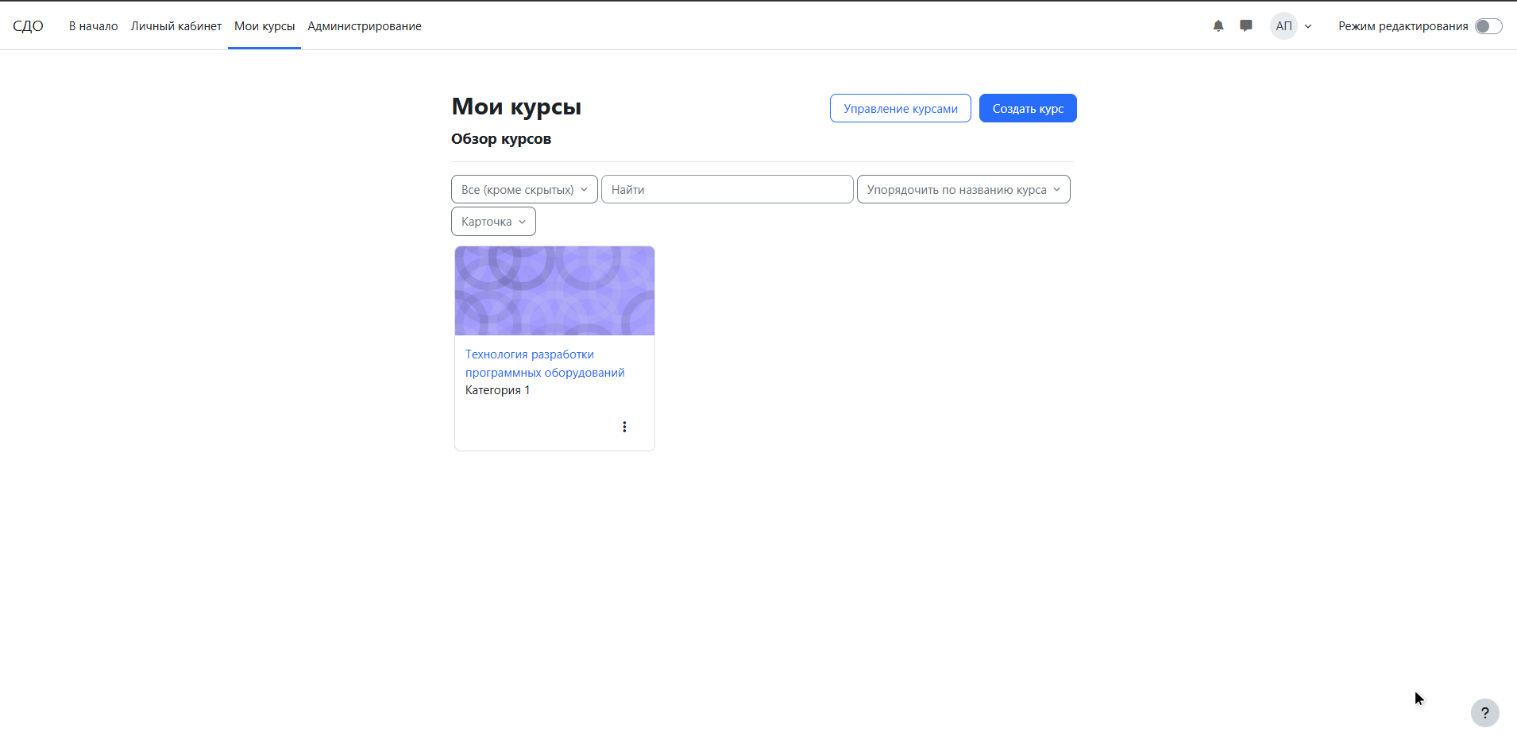
Страница «Мои курсы» представлена на рисунке 8.

Рисунок 8 – Страница «Мои курсы»

Страница профиля включает в себя информацию о пользователе, оценки и отчеты.

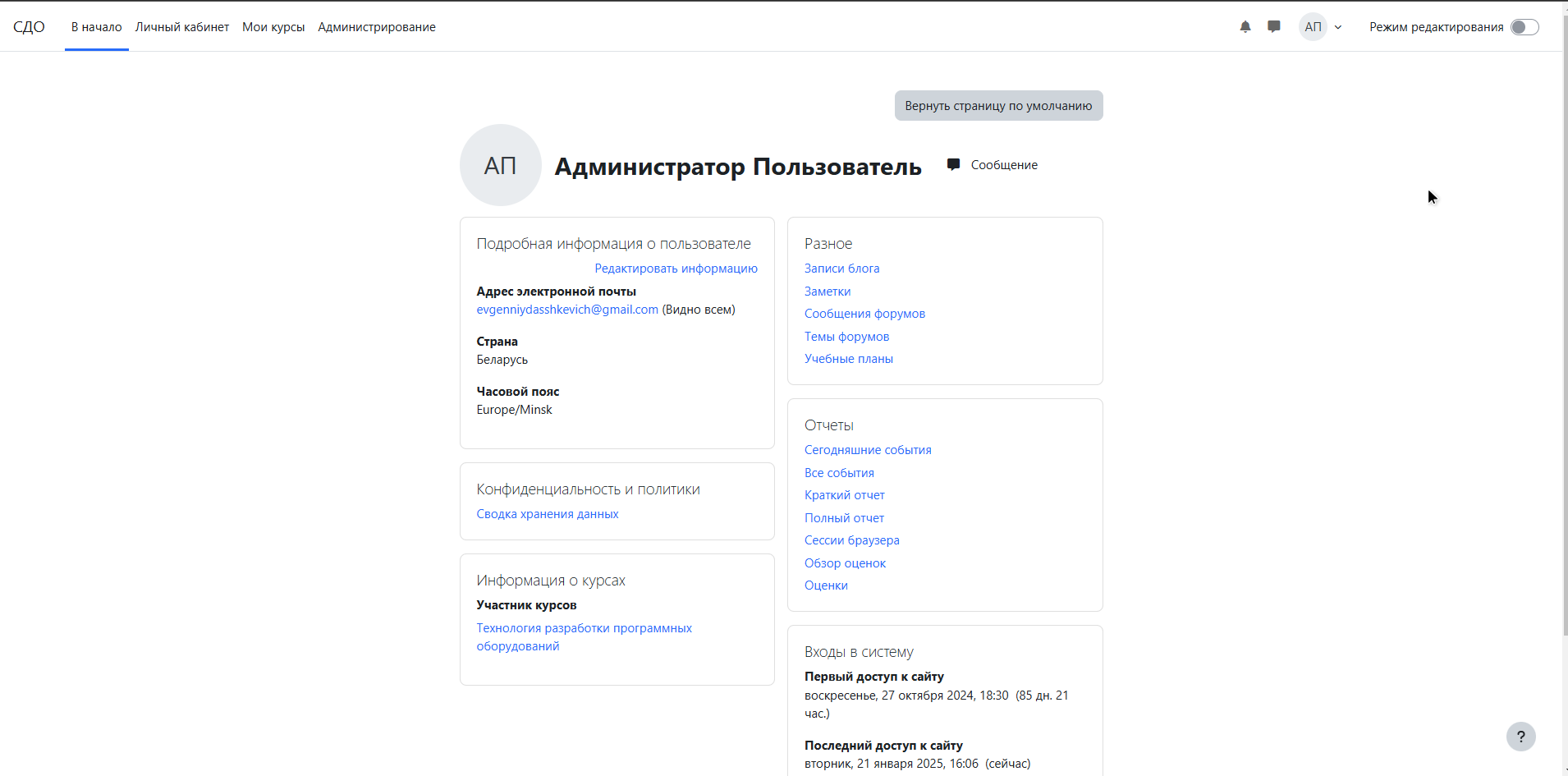
Страница профиля пользователя представлена на рисунке 9.

Рисунок 9 – Страница профиля пользователя

**Заключение**

Целью данного проекта является разработка системы дистанционного обучения для сайта «Гродненский Государственный Политехнический Колледж», который позволил дать возможность, добавлять курсы, узнавать окончания курсов, календарь, оценки.

В ходе реализации поставленной задачи были закреплены знания по использованию конструктора Moodle, а также курс предмета «Технология разработки программного обеспечения».

Поставленная задача выполнена в соответствии с требованиями, созданы и протестированы все необходимые страницы и возможности проекта.

В ходе тестирования все исключительные ситуации были отработаны. Проект работает без сбоев и ошибок. В поставленной задаче был реализован простой и понятный пользовательский интерфейс.

Исходя из этого, можно сделать вывод, что программа реализована успешно.

**Список использованных источников**

1. Форум Moodle [Электронный ресурс]. – Режим доступа: <https://moodle.org/course/> – Дата доступа: 21.12.2024.

2. TAUSOFT - компьютер доступен каждому [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/@TAUSOFTs> – Дата доступа 20.11.2024.

3. Курс по LMS MOODLE [Электронный ресурс]. – Режим доступа: <https://www.youtube.com/@lms-moodle> – Дата доступа: 20.01.2025.