

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г.

ЧЕРНЫШЕВСКОГО»

Кафедра дифференциальных уравнений и математической экономики

**СТРАТЕГИЯ РЕАЛИЗАЦИИ ПРОЕКТА В СЕЛЬСКОМ ХОЗЯЙСТВЕ С
ИСПОЛЬЗОВАНИЕМ ВЕКСЕЛЕЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 451 группы

направления 38.03.05 — Бизнес-информатика

механико-математического факультета

Зайнутдиновой Зарины Фаридовны

Научный руководитель

доцент к. ф.-м. н.,

доцент

И. Ю. Выгодчикова

Заведующий кафедрой

зав. кафедрой д. ф.-м. н.,

профессор

С. И. Дудов

Саратов 2025

ВВЕДЕНИЕ

Актуальность темы исследования. Сельское хозяйство часто сталкивается с ограничениями в доступе к традиционным кредитным инструментам. Высокие процентные ставки, недостаточная ликвидность, а также сложные процедуры получения кредитов делают традиционное банковское финансирование неэффективным для многих аграриев. В условиях нестабильной экономики и изменения климата аграрный сектор требует более гибких и адаптивных финансовых инструментов. Векселя могут стать одним из таких инструментов, позволяя фермерам привлекать средства под более выгодные условия и без необходимости предоставления залога.

Безопасность и предсказуемость доходности векселей делают их привлекательным инструментом для инвестиций в агросектор.

Целью дипломной работы является разработка веб-приложения для инвестиций в сельское хозяйство с использованием векселей, направленного на использование векселей как инновационного финансового инструмента для повышения доступности финансирования и улучшения финансовой устойчивости аграрных предприятий.

Задачи дипломной работы:

1. анализ теоретических основ;
2. исследование текущего состояния сельского хозяйства;
3. разработка стратегии реализации проекта;
4. определение функциональных возможностей веб-приложения AgroInvest;
5. разработка пользовательского интерфейса;
6. программирование основных функций приложения, включая регистрацию пользователей, выпуск и управление векселями.

Объектом исследования является инвестиционный процесс в сельском хозяйстве, включая механизмы финансирования и использования финансовых инструментов, таких как вексели.

Предметом исследования являются функциональные возможности и пользовательский интерфейс платформы AgroInvest как факторы, влияющие на активность инвесторов и эмитентов.

Основное содержание работы

Бакалаврская работа состоит из: введения, двух теоретических и практической глав, заключения, списка использованных источников, приложений.

Введение содержит основные положения: актуальность темы исследования (цель, задачи исследования, объект, предмет).

Первый раздел «Теоретические основы финансирования сельскохозяйственных проектов и роль векселей в финансовой системе» описывает финансовые аспекты, связанные с сельскохозяйственными проектами, и анализирует роль векселей как потенциального инструмента для облегчения финансирования в этой отрасли.

Особенности финансирования сельскохозяйственных проектов. Сельское хозяйство сталкивается с рядом негативных факторов, таких как сезонность производства и зависимость от внешнего финансирования, что делает актуальным бюджетное финансирование отрасли. Это финансирование осуществляется через прямое субсидирование (например, компенсационные выплаты) и косвенные меры (страхование урожая, субсидированные кредиты). В России государственная поддержка сельского хозяйства является важным приоритетом экономической политики, с акцентом на "Государственную программу развития сельского хозяйства". Традиционные источники финансирования, такие как кредиты и лизинг, важны для аграрного сектора, однако многие предприятия сталкиваются с проблемами доступа к финансам, включая высокие процентные ставки и сложности с залогами.

Вексель как инструмент финансирования: экономическая сущность и виды. Вексель — это ценная бумага, представляющая собой безусловное обязательство одного лица выплатить определенную сумму другому лицу. Основное назначение векселя — кредитование, позволяющее оформлять финансовые обязательства, такие как оплата товаров и возврат займов. Вексель выгоден для покупателя, так как обеспечивает отсрочку платежа без участия банков.

В бизнесе векселя используются для краткосрочного финансирования, особенно при временном дефиците средств. Компании могут выпускать векселя после отгрузки товаров, чтобы кредиторы могли получить деньги после

поступления оплаты от покупателей. Векселя могут быть выданы как юридическими, так и физическими лицами, хотя последние чаще выступают как инвесторы.

Также будут рассмотрены различные виды векселей, применяемых в сельском хозяйстве.

Второй раздел «Текущее состояние сельского хозяйства и финансирования проектов» предоставляет комплексное понимание текущего состояния сельского хозяйства и механизмов его финансирования, а также выявляет ключевые проблемы и возможности для дальнейшего развития.

Обзор современного состояния сельского хозяйства в России. Сельское хозяйство играет ключевую роль в российской экономике, обеспечивая продовольственную безопасность и обладая значительным экспортным потенциалом. После кризиса 1990-х годов отрасль начала устойчивый рост в 2000-х, благодаря активной поддержке правительства, которое рассматривает сельское хозяйство как драйвер экономического развития.

Сегодня Россия является одним из крупнейших экспортеров зерна в мире, а также демонстрирует рост в птицеводстве и свиноводстве, снижая зависимость от импорта. Однако молочное животноводство и производство крупного рогатого скота остаются проблемными и требуют дополнительных инвестиций и поддержки.

Основой модели развития является государственно-частное партнерство, где агрохолдинги получают финансирование от государства, в том числе через «Россельхозбанк». Несмотря на успехи, отрасль сталкивается с вызовами: низкая рентабельность, нехватка инфраструктуры, дефицит квалифицированных кадров, зависимость от импорта семян и риски изменения климата. Эти проблемы требуют комплексного подхода к решению.

Проблемы финансирования сельскохозяйственных проектов. Сельское хозяйство в России сталкивается с серьезными вызовами в привлечении финансирования. Сезонность, высокие риски и длительные циклы производства делают аграрный бизнес уязвимым и менее привлекательным для инвесторов. Будет подробнее рассмотрено то, как погодные условия, нехватка залогового обеспечения и высокие транзакционные издержки влияют на

финансовые возможности фермеров и малых предприятий в этой важной отрасли.

Возможности использования векселей для решения финансовых проблем. В мире финансов, где инновации и новые технологии занимают центральное место, вексель остается недооцененным, но мощным инструментом. Этот традиционный вид ценной бумаги предлагает уникальные возможности для бизнеса и частных лиц, обеспечивая гибкость в расчетах и финансировании. Однако, несмотря на свои преимущества, вексель также несет определенные риски, такие как кредитный риск и сложности с ликвидностью. Важно правильно использовать вексели, тем самым минимизировать риски и максимально эффективно применять этот инструмент в современных финансовых реалиях.

Третий раздел «Разработка веб-сайта» посвящён ключевым этапам и аспектам разработки веб-сайта.

Типовая структура web-сайта. Платформа, соединяющая фермеров и инвесторов, предоставляет удобные инструменты для управления векселями. В каталоге можно найти подробную информацию о каждом векселе, а также есть возможность создавать и публиковать собственные предложения в личном кабинете. Безопасная регистрация и аутентификация гарантируют защиту ваших данных.

Обоснование выбора средств разработки web-сайта. В этом разделе рассматривается выбор технологического стека для создания веб-сайта, который будет управлять вексельным финансированием сельскохозяйственных проектов. Выбор стека основан на критериях простоты разработки, гибкости, масштабируемости, безопасности и низкой стоимости, что позволяет создать функциональный и безопасный веб-сайт с минимальными затратами и временем на разработку.

Python обеспечивает простоту и широкий спектр применения, а Flask как микрофреймворк предоставляет необходимые инструменты без излишней нагрузки. HTML и CSS формируют структуру и стиль страниц, а Bootstrap ускоряет разработку адаптивного интерфейса. SQLite, встраиваемая база данных, упрощает развертывание благодаря отсутствию необходимости в отдельном сервере.

Этапы разработки web-сайта.

Перед непосредственным созданием веб-сайта необходимо подготовить папки, в которых будут содержаться необходимые документы, такие как шрифты, изображения, логотип и другие статические файлы для стилизации.

Подготовка структуры папок перед началом разработки веб-сайта – важный шаг для организации проекта, обеспечения удобства работы и поддержания порядка. Это позволяет избежать хаоса и упрощает поиск нужных файлов в дальнейшем.

В папке `css` каталога `static` содержатся следующие компоненты:

- `all.min.css` обычно относится к минифицированной версии CSS-файла, который содержит все стили для иконочного шрифта Font Awesome.

- `bootstrap.min.css` — это минифицированная версия CSS-файла (или JavaScript-файла, в зависимости от расширения) фреймворка Bootstrap, одного из самых популярных бесплатных и открытых CSS-фреймворков для разработки адаптивных, мобильных веб-сайтов и веб-приложений. Он предоставляет готовые CSS-стили, компоненты и JavaScript-плагины, которые позволяют быстро создавать современные и профессионально выглядящие веб-интерфейсы.

- `styles` - это условное обозначение для CSS-файла, который содержит пользовательские стили вашего веб-сайта.

Папка `scripts` содержит:

- `bootstrap.bundle.min.js` — это минифицированная версия JavaScript-файла, входящего в состав фреймворка Bootstrap. Она содержит в себе функциональность как Bootstrap JavaScript, так и Popper.js. Bootstrap JavaScript предоставляет интерактивные компоненты и функции, которые позволяют создавать динамичные веб-сайты и приложения.

- `jquery-3.6.0.min.js` — это минифицированная версия JavaScript-библиотеки jQuery версии 3.6.0. jQuery — это быстрая, маленькая и многофункциональная JavaScript-библиотека. Она упрощает работу с HTML-документами, обработку событий, анимацию и AJAX (асинхронные запросы к серверу) в веб-разработке. Можно сказать, что jQuery делает JavaScript более простым и удобным в использовании.

В папке uploads содержатся все необходимые картинки для формирования карточек эмитентов.

Все необходимые шрифты помещены в папку webfonts.

Далее следует подготовка инструментов для работы с базами данных и аутентификацией на сайте. Создаются экземпляры расширений SQLAlchemy и LoginManager в файле extensions.py, которые будут использоваться для управления этими аспектами веб-сайта.

Реализуется импортирование класса SQLAlchemy из библиотеки flask_sqlalchemy.

SQLAlchemy - это расширение Flask, которое упрощает работу с реляционными базами данных. Он предоставляет инструменты для определения моделей данных как классов Python и для выполнения операций CRUD с базой данных с использованием объектно-ориентированного подхода.

Реализуется импортирование класса LoginManager из библиотеки flask_login.

LoginManager - это расширение Flask, которое упрощает управление аутентификацией пользователей в ваших Flask-приложениях. Он предоставляет инструменты для управления сессиями пользователей, входа и выхода из системы, защиты определенных представлений от неавторизованных пользователей и т.д.

Также необходимо предоставить информацию о версиях установленных пакетов (библиотек) в Python-окружении, которая будет содержаться в файле requirements.txt.

Flask - это микрофреймворк для Python, который упрощает разработку веб-приложений. Он предоставляет базовые инструменты для обработки HTTP-запросов, маршрутизации, рендеринга шаблонов и т.д.

Flask-SQLAlchemy - это расширение Flask, которое упрощает работу с реляционными базами данных. Он предоставляет ORM (Object-Relational Mapper), который позволяет взаимодействовать с базой данных с использованием объектно-ориентированного подхода.

Flask-Login - это расширение Flask, которое упрощает управление аутентификацией пользователей. Он предоставляет инструменты для управления сессиями пользователей, входа и выхода из системы, защиты определенных представлений от неавторизованных пользователей и т.д.

Werkzeug - это набор инструментов для разработки WSGI-совместимых веб-приложений на Python. Flask основан на Werkzeug. Он предоставляет утилиты для обработки HTTP-запросов и ответов, маршрутизации, отладки и т.д.

Следующим шагом разработки является создание базы данных - card и user. Для этого необходимо создать модели, которые будут описывать сущности карточки проекта и пользователя, соответственно.

Далее следует создание базовой структуры Flask-приложения в файле app.py с поддержкой работы с базой данных (SQLAlchemy) и аутентификации пользователей (Flask-Login), разделенную на логические компоненты с помощью blueprints. После запуска данного файла SQLAlchemy автоматически создаст таблицы card и user.

Таблица card хранит информацию о инвестиционном проекте эмитента: название проекта, его описание, местоположение, площадь и хозяйства, размер и срок инвестиций, ожидаемую доходность, фото, контактные телефон и почту, дополнительную информацию.

Таблица user хранит основную информацию о пользователях системы: ФИО, логин, пароль и факт, является ли пользователь эмитентом.

Создание шаблонов и форм – важная часть разработки веб-приложения на Flask. Они позволяют отображать информацию пользователю и получать данные от него. Рассмотрим создание шаблонов и форм.

В первую очередь создаем базовый шаблон base.html. Его основное содержание - общие элементы, такие как заголовок, навигация, подвал и подключение стилей или скриптов, которые будут использоваться на всех страницах сайта.

Файл auth содержит HTML-шаблоны, используемые для создания страниц входа и регистрации пользователей в веб-приложении.

Шаблон страницы входа (login.html) представляет собой интерфейс для аутентификации пользователей в Flask-приложении AgroInvest.

Шаблон register.html отвечает за создание интерфейса страницы, где новые пользователи могут зарегистрировать свои учетные записи.

В файле card находятся HTML-шаблоны, определяющие структуру и внешний вид карточек проектов в веб-приложении.

Шаблон `card_detail.html` служит для детального представления инвестиционного проекта потенциальным инвесторам.

Шаблон `create_card.html` используется для создания формы, позволяющей фермерам добавлять новые инвестиционные проекты.

Шаблон `my_projects.html` используется для создания страницы, на которой пользователи могут видеть список своих инвестиционных проектов в AgroInvest.

В файле `main` содержится шаблон `index.html` для отображения списка всех доступных инвестиционных проектов, предназначенный для просмотра потенциальными инвесторами.

Был завершен этап разработки шаблонов для создания динамических веб-страниц. В будущем Flask автоматически будет искать их директории.

Заключительным этапом создания веб-сайта является написание эндпойнтов для регистрации и аутентификации пользователей, создания и чтения карточек.

Endpoint — это конечная точка веб-сервиса, к которой клиентское приложение обращается для выполнения определённых операций или получения данных. Эндпоинты позволяют клиентам взаимодействовать с сервером или сервисом, передавая данные и получая ответы.

Для выполнения этой задачи создадим файлы `auth.py`, `cards.py` и `main.py` в каталоге `routes`.

Создание маршрута для аутентификации для управления сессиями пользователей (`auth.py`):

1. Регистрация (`/register`):

- Обработывает как GET, так и POST запросы.

При POST запросе:

- Получает данные из формы (логин, пароль, полное имя, статус фермера).
- Проверяет, существует ли пользователь с таким логином.
- Если пользователь существует, выводит сообщение об ошибке и перенаправляет на страницу регистрации.
- Если нет, создает нового пользователя с хэшированным паролем и добавляет его в базу данных.

- После успешной регистрации перенаправляет на страницу входа.

При GET запросе отображает страницу регистрации.

2. Вход в систему (/login):

- Обрабатывает как GET, так и POST запросы.

При POST запросе:

- Получает данные из формы (логин и пароль).
- Находит пользователя по логину.
- Проверяет, существует ли пользователь и соответствует ли введенный пароль хэшированному паролю.

• Если данные верны, выполняет вход пользователя и перенаправляет на главную страницу.

- Если данные неверны, выводит сообщение об ошибке.

При GET запросе отображает страницу входа.

3. Выход из системы (/logout):

• Обрабатывается только при наличии активной сессии пользователя (декоратор @login_required).

- Вызывает функцию `logout_user()` для выхода пользователя из системы.
- Перенаправляет на главную страницу после выхода.

Создание маршрута для работы с карточками, который позволяет пользователям (фермерам) создавать, просматривать и управлять своими проектами (`cards.py`):

1. Создание карточки (/create_card):

- Проверяет, является ли текущий пользователь фермером (эмитентом).

Если нет, выводит сообщение об ошибке и перенаправляет на главную страницу.

При POST запросе:

- Получает данные из формы (название, описание, сумма и т.д.).
- Обрабатывает загружаемое изображение (если есть) и сохраняет его в указанной директории.
- Создает экземпляр карточки с полученными данными и добавляет его в базу данных.

• Перенаправляет на главную страницу после успешного создания карточки.

При GET запросе отображает страницу создания карточки.

2. Просмотр деталей карточки (`/card/<int:card_id>`):

- Получает карточку по идентификатору и отображает страницу с ее деталями. Если карточка не найдена, возвращает 404 ошибку.

3. Мои проекты (`/my_projects`):

- Проверяет, является ли текущий пользователь фермером. Если нет, выводит сообщение об ошибке и перенаправляет на главную страницу.

- Получает все карточки пользователя, отсортированные по дате создания, и отображает страницу с проектами пользователя.

Создание основного маршрута для веб-сайта, который отображает главную страницу со списком карточек, выглядит следующим образом:

1. Создается объект Blueprint с именем 'main', который позволяет организовать маршруты приложения.

3. Декоратор `@bp.route('/')` связывает URL-адрес `/` с функцией `index()`.

4. В функции `index()` выполняется запрос к базе данных для получения всех карточек (`cards = Card.query.all()`).

5. Возвращается HTML-шаблон `index.html`, в который передаются данные о карточках для отображения на главной странице.

Для запуска релизованного кода необходимо выполнить следующие шаги:

1. Установка зависимостей:

```
pip install -r requirements.txt
```

Эта команда устанавливает все необходимые библиотеки и пакеты, указанные в файле `requirements.txt`. Если запуск прошел успешно, переходим к следующему шагу.

2. Запуск приложения:

```
python app.py
```

После запуска приложения необходимо открыть веб-браузер и перейти по адресу `http://localhost:5000`, чтобы увидеть веб-сайт в действии.

Заключение

Дипломная работа была посвящена разработке стратегии реализации проекта в сельском хозяйстве с использованием векселей в качестве инновационного финансового инструмента. Актуальность данной темы обусловлена

ограничениями в доступе к традиционным кредитным инструментам для аграриев, что негативно сказывается на развитии отрасли. В условиях нестабильной экономики и климатических изменений требуется более гибкий и адаптивный подход к финансированию сельского хозяйства.

В рамках исследования была достигнута поставленная цель: разработана стратегия, направленная на повышение доступности финансирования и улучшение финансовой устойчивости аграрных предприятий посредством использования векселей.

Поставленная цель была достигнута путем решения следующих задач:

1. Проведен анализ теоретических основ финансирования сельскохозяйственных проектов: Были изучены особенности аграрного сектора, его финансовые потребности и существующие инструменты финансирования, что позволило определить место и роль векселей в системе финансовых отношений.
2. Исследовано современное состояние сельского хозяйства в России: Анализ позволил выявить основные проблемы отрасли, в том числе трудности доступа к финансированию, и определить потенциальные направления для использования векселей.
3. Разработана стратегия реализации проекта в сельском хозяйстве с использованием векселей: Стратегия включает в себя обоснование выбора конкретного сельскохозяйственного проекта, разработку схемы финансирования с использованием векселей.
4. Были определены основные функции приложения. Это обеспечило создание удобного и интуитивно понятного интерфейса для пользователей.
5. Учитывая потребности целевой аудитории, был создан интерфейс, который обеспечивает удобство использования и делает приложение более доступным и привлекательным для пользователей.
6. В результате разработки были реализованы ключевые функции, включая регистрацию пользователей и управление векселями. Это стало основой для полноценного функционирования приложения.

В заключение, данная дипломная работа представляет собой вклад в разработку инновационных подходов к финансированию сельского хозяйства.

Разработанная стратегия может быть использована сельскохозяйственными предприятиями для привлечения финансирования и повышения эффективности своей деятельности, а также органами государственной власти при разработке мер поддержки сельского хозяйства.