|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |
| **«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** |
| **(МОСКОВСКИЙ ПОЛИТЕХ)** |
| Факультет информационных технологий |

Кафедра «Прикладная информатика»

Форма обучения: очная

**ОТЧЕТ**

**о выполнении проекта  
 «Проект информационной системы финансового планирования, предоставляющей услугу подбора индивидуальных инвестиционных предложений клиентам компании ZLP Technologies»**

**по дисциплине**

**«Проектная деятельность»**

Студент (Н.А. Каткова)

(И.А. Акулин)

(А.Н. Липина)

**Москва 2020**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc62871635)

[1. Общее задание, организация работ, участники проекта 5](#_Toc62871636)

[2. Информационная система и ее функции 9](#_Toc62871637)

[3. Формирования инвестиционных портфелей и сервисы 11](#_Toc62871638)

[4. Инструменты и технологии в проекте 17](#_Toc62871639)

[5. Этап формирования технического задания 19](#_Toc62871640)

[6. Этап разработки и тестирования ИС 21](#_Toc62871641)

[7. Обоснование выбора базы данных для будущих решений 26](#_Toc62871642)

[8. Подведение итогов и результаты работ 33](#_Toc62871643)

[ЗАКЛЮЧЕНИЕ 35](#_Toc62871644)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc62871645)

1. ВВЕДЕНИЕ

В современном мире финансовые инвестиции стали иметь невероятно огромную роль в развитии мирового рынка. Сложно найти работника крупной компании, который бы не занимался инвестированием или же игрой на финансовых биржах, так как это не только влияет на развитие определенных компаний, имеющих вес в тех или иных сферах жизни, но и приносит огромную выгоду лично инвестору. Однако данный вид финансовой активности имеет огромные риски, в особенности для новичка в данной сфере.

Несмотря на все это так же время стало стоить очень дорого и не всегда человек, желающий принять участие в финансовых инвестициях имеет свободное время на изучение рынка. Именно поэтому компания ZLP Tech решила предложить своим клиентам информационную систему финансового планирования, предоставляющую услугу подбора индивидуальных инвестиционных предложений.

Данный проект является актуальным, так как очень многие клиенты компании желают войти в мир инвестиций без огромных затрат времени и с использованием удобной и информативной системы. Это поможет компании расширить свою клиентскую базу и предоставить удобные возможности, в следствие чего увеличить прибыль.

Для создания первого этапа данного проекта необходимо выполнить следующие задачи:

1. Изучить текущие аналоги приложений в сфере финтех;
2. Сформировать ТЗ для создания минимально жизнеспособной версии продукта;
3. Разработать в несколько итераций первые версии проекта на данном этапе;
4. Протестировать и исправить ошибки;
5. Опубликовать исходный код проекта на GitHub;
6. Подготовить сайт-презентацию, отчет и другие материалы для защиты проекта.

Основные цели создания системы:

1. Увеличение прибыли компании;
2. Снижение временных затрат на индивидуальное составление плана пользователя;

Основные цели развития системы:

1. Снижение числа задействованной рабочей силы в создании индивидуальных инвестиционных предложений;
2. Увеличение клиентской базы компании.

Результатом работы на данном этапе проекта является frontend часть информационной системы финансового планирования, предоставляющей услугу подбора индивидуальных инвестиционных предложений клиентам компании ZLP Tech – анкеты для подбора портфеля.

1. Общее задание, организация работ, участники проекта

Общее задание проекта заключается в разработке системы для финансового планирования, предоставляющей услугу подбора индивидуальных инвестиционных предложений клиентам компании ZLP Tech.

На данном этапе планировалось реализовать frontend-часть проекта – анкету для прохождения клиентами и начальный вариант веб-сайта системы.

План работ по проекту состоит из следующих этапов:

1. Создание каналов коммуникации участников и кураторов проекта;
2. Определение системы трекинга задач для участников проекта;
3. Формулировка целей и задач проекта для его реализации;
4. Распределение ролей между участниками проекта;
5. Сбор и анализ информации об имеющихся на рынке вариантах систем;
6. Разработка технического задания;
7. Оформление страницы проекта и необходимых папок в GitHub;
8. Разработка технической части приложения (предварительно выбрав используемые технологии) в 2 этапа;
9. Создание веб-сайта проекта в качестве презентации;
10. Тестирование и исправление приложения;
11. Описание проекта в итоговом отчете и подготовка финального скринкаста защиты.

Первоначально заявленные участники проекта студенты группы 181-371:

1. Каткова Н. А. – координатор, разработчик.
2. Акулин И. А. – технический писатель.
3. Липина А. Н – технический писатель.

Индивидуальный план работ по проекту менеджера и разработчика Катковой Н.А.:

1. Координация работ проекта;
2. Оформление GitHub (доработки);
3. Составление плана-графика работ всего проекта;
4. Составление индивидуальных планов работ;
5. Составление функциональной части ТЗ;
6. Составление итоговой версии ТЗ;
7. Разработка ИС – Frontend;
8. Тестирование ИС и исправление дефектов;
9. Разработка презентационного веб-сайта;
10. Формирование отчета по проделанной работе;
11. Подготовка видеоматериала для итоговой защиты проекта.

Индивидуальный план работ по проекту технического писателя Акулина

И. А.:

1. Формирование технического задания;
2. Тестирование информационной системы;
3. Помощь в формировании отчета по проделанной работе.

Индивидуальный план работ по проекту технического писателя

Липиной А. Н.:

1. Формирование технического задания;
2. Тестирование информационной системы;
3. Помощь в формировании отчета по проделанной работе.

Коммуникация участников проекта происходила с помощью каналов

связи Discord и в чате социальной сети ВКонтакте.

Для работы над проектом был создан и оформлен по правилам проектной деятельности приватный репозиторий в GitHub, добавлены доступы для участников проекта и куратора (рис. 1):

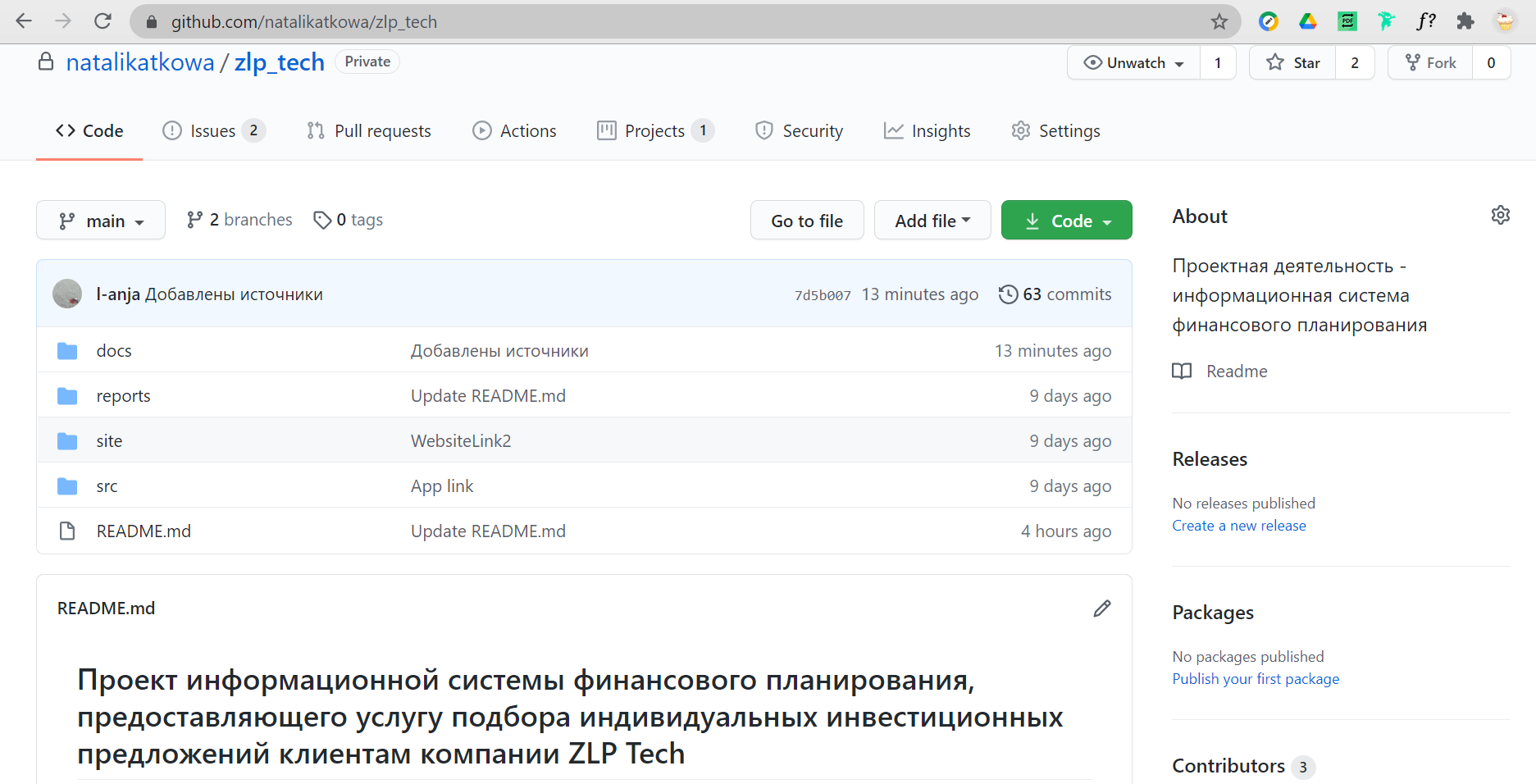


Рисунок 1. Репозиторий в GitHub

Для координации работ по проекту по итогам сравнительного анализа и личного опыта была выбрана облачная программа для управления проектами Trello - это одна из самых популярных систем управления проектами в режиме онлайн, которая пользуется особенным спросом среди небольших компаний и стартапов. Она позволяет эффективно организовать работу по японской методологии канбан-досок.

В текущей канбан доске установлен синхронизированный удобный трекинг проекта:

1. Объявлены TO DO листы общий и каждого участника проекта;
2. Объявлен лист IN PROGRESS для отслеживания текущих выполняемых задач участниками проекта;
3. Объявлены листы DONE для каждой из двух итераций проекта;
4. К каждой карточке задачи устанавливается дедлайн, чекбокс о выполнении, прикрепляется участник задачи, добавлены описания и комментарии по необходимости;
5. Дополнительно объявлены листы ресурсов и чек-лист необходимых задач для допуска к защите;
6. При выполнении задачи участник должен сделать коммит в GitHub с комментарием, оповестить об этом команду и перенести выполненную задачу в колонку DONE соответствующей итерации.

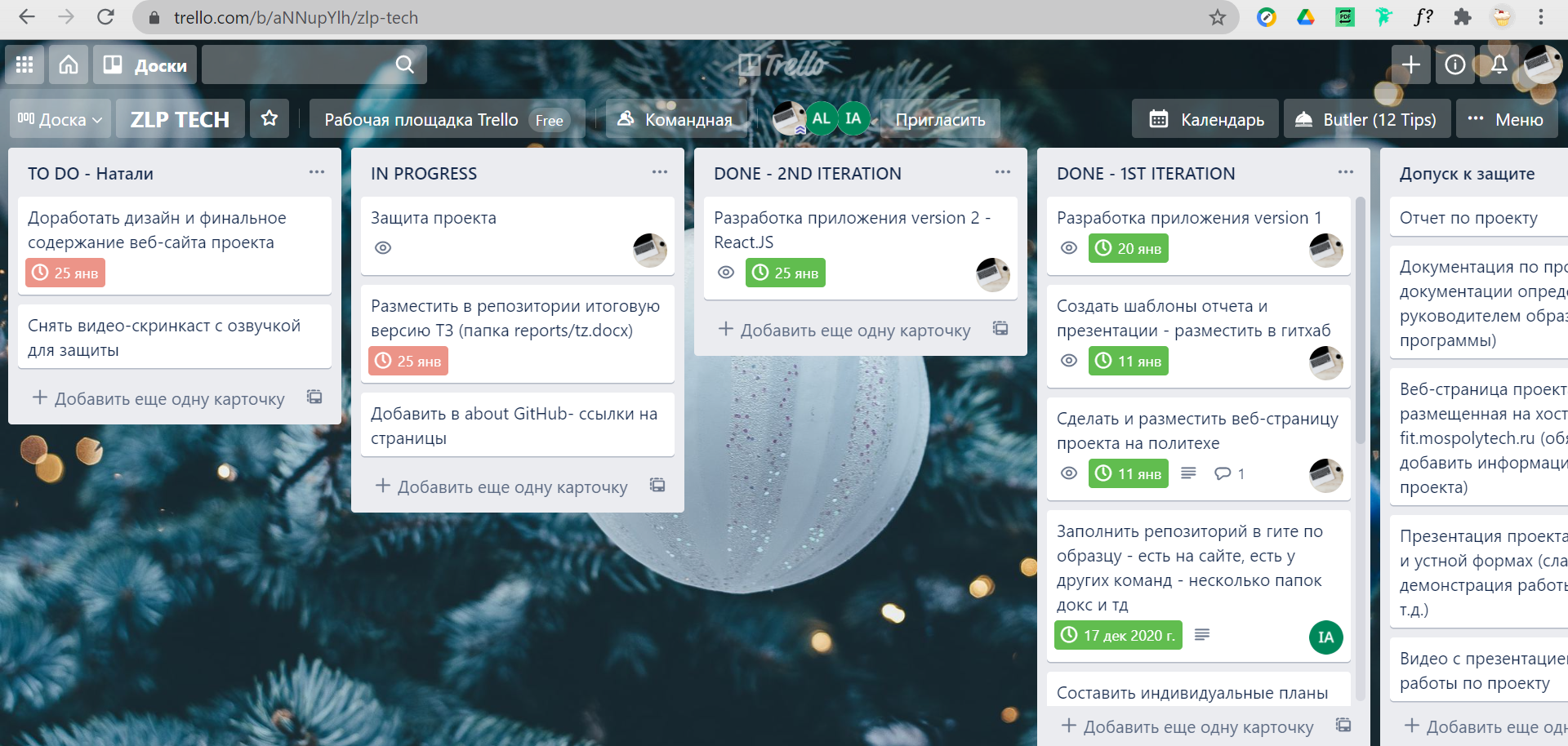


Рисунок 2. Трекинг проекта в TRELLO

1. Информационная система и ее функции

Информационная система — это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации для достижения цели управления. В современных условиях основным техническим средством обработки информации является персональный компьютер. Большинство современных информационных систем преобразуют не информацию, а данные. Поэтому часто их называют системами обработки данных.

Основные функции информационной системы (ИС) — сбор, передача и хранение информации, а также операции обработки — ввод, выборка, корректировка и выдача информации.

К информационным системам предъявляются следующие требования:

1. Полнота и достаточность информации для реализации функций управления;
2. Своевременность представления информации;
3. Обеспечение необходимой степени достоверности информации в зависимости от уровня управления;
4. Экономичность обработки информации (затраты на обработку данных не должны превышать получаемый эффект);
5. Адаптивность к информационным потребностям пользователей.

Цель внедрения информационных систем — повышение эффективности производственно-хозяйственной деятельности фирмы за счет новых методов управления, основанных на моделировании деятельности специалистов фирмы при принятии решений (методы искусственного интеллекта, экспертные системы, нейронные семиотические сети, нейротехнологии и т. п.), использования современных средств телекоммуникаций (электронная почта, телеконференции, видеоконференции) и вычислительных сетей.

На практике, при подборе системы управления, предприятие рассматривает возможность ее создания собственными силами, опираясь на внутренние наработки и компетенций сотрудников, собирает систему из набора готовых решений, каждое из которых решает отдельную задачу (управление складом, управление продажами, транспортом и т.п.), либо внедряет существующие на рынке комплексные системы управления (например, ERP) [1].

Компании в рамках своей деятельности осуществляют ряд действий во внутренней и внешней среде, которые в свою очередь, направлены на достижение долгосрочных и краткосрочных целей.

Под внутренней средой понимается контур компании, к которому относятся любые отношения, взаимодействия, бизнес-процессы направленные на обеспечение жизнедеятельности компании. Например, внутренние регламенты, система мотивации, корпоративная культура, ведение бухгалтерского и управленческого учета, планирование, контроль, производственные операции, кадровый учет.

Внешняя среда – это отношения, взаимодействия, бизнес-процессы, которые осуществляет компания с внешними контрагентами. Такие как, продажи, работа с поставщиками, отношения с клиентами и инвесторами. Все действия компании, которые осуществляются во внешней среде, имеют прямое влияние на деятельность во внутренней среде.

Информационные системы предприятия отлично решают задачи, стоящие перед современными предприятиями, которым необходима возможность доступа к большим объемам разноплановых данных, а также возможность их автоматической обработки [2].

1. Формирования инвестиционных портфелей и сервисы

Традиционно считается что для инвестиций на фондовом рынке требуются серьезная подготовка и значительные суммы денег. Но времена меняются и в современном мире, в том числе и в России, уже появились финансовые сервисы, которые дают возможность заработать на инвестициях даже без каких-либо серьезных профессиональных знаний и с небольшим количеством капитала [3].

Благодаря технологиям микроинвестиций и краудлендинга порог входа на финансовые рынки существенно снизился и появились удобные и понятные инструменты для обычных людей - людей без специального образования. Формирование инвестиционного портфеля — важная задача для каждого инвестора. Правильно составленный портфель — ключ к успеху на фондовом рынке.

Инвестиционный портфель — это набор финансовых инструментов, направленный на сохранение и увеличение денежных средств. Такой набор может состоять из акций, облигаций, ПИФов и других инструментов. Они отличаются не только базовыми характеристиками, но и уровнем риска, доходностью. Как правило, чем выше риск, тем выше потенциальная доходность вложения. Например, наиболее подвержены риску инвестиции в акции молодых компаний. Самыми низкорисковыми инструментами считаются государственные облигации (ОФЗ) [7].

Существует много различных типов инвестиционных портфелей; их можно различать по различным критериям, но основные виды инвестиционных портфелей различаются по типу соотношения риск/прибыль [8]:

* **Консервативный -** состоит из государственных ценных бумаг, акций [«голубых фишек»](https://fortrader.org/birzhevoj-slovar/birzhevie-termini/obyknovennye-akcii.html), золота и обеспечивает высокую защищенность составляющих и портфеля в целом, а доходность инвестиционного портфеля сохраняется на необходимом инвестору уровне.
* **Умеренный -** оптимизация по доходности и степени риска — характерная черта умеренного инвестиционного портфеля. В таком портфеле находятся как высокодоходные бумаги с высокой степенью риска, так и низкодоходные надежные бумаги, типа государственных облигаций.
* **Агрессивный -** портфель укомплектован высокодоходными ценными бумагами, включая их дериваты. Эти ценные бумаги имеют достаточно высокую степень риска. Поэтому, инвестор сам активно управляет своим портфелем или постоянно контактирует с управляющим его портфелем брокером.

При формировании инвестиционного портфеля можно выделить несколько основных этапов:

1. Определение себя как инвестора: консерватор, умеренный или агрессивный инвестор;
2. Определение цели инвестирования: максимум доходности, минимум риска, быстрый рост капитала, быстрый возврат инвестиций или сочетание этих целей;
3. Анализ финансовых рынков исходя из целей инвестирования;
4. Выбор активов для инвестиций и определение их соотношения для заданного уровня доходности и минимума степени риска;
5. Приобретение активов и начало текущего мониторинга сформированного портфеля.

Портфельные инвестиции могут управляться инвестором непосредственно или передаться в доверительное управление брокерской компании. Передача в доверительное управление не означает полную передачу прав на управлением портфелем. Изменение состава портфеля, увеличение или снижение стоимости активов портфеля происходит с обязательным согласованием с инвестором [9].

Сущность управления портфелем для инвестора состоит в поддержании его доходности на определенном уровне. Здесь существует два основных подхода:

* Формирование высоко диверсифицированного портфеля с заданным уровнем соотношения доходы/риски.
* Формирование высокодоходного портфеля с высоким уровнем риска.

При этом управление инвестиционным портфелем может быть двух видов:

Активное управление - здесь речь идет о постоянном отслеживании рынка, покупке и оперативной продаже тех активов, которые теряют свою ликвидность. В таком виде управления состав инвестиционного портфеля часто подвергается изменениям.

Пассивное управление - здесь подразумевается сбор качественного диверсифицированного портфеля, его сохранение и получение прибыли.

Главными параметрами при формировании и управлении инвестиционным портфелем, являются его ожидаемая доходность и риск. В связи с отсутствием возможности точного определения динамики вышеуказанных параметров, данные величины оцениваются, в первую очередь, на основе статистической информации за предыдущие периоды времени.

Ожидаемая доходность портфеля рассчитывается на основе ожидаемой доходности содержащихся в нём активов. Инвестор должен ясно понимать, что риск и доходность инвестиционного портфеля – это связующие звенья одной цепи. Потенциальное увеличение доходности влечет за собой увеличение риска, и наоборот.

Риски могут определяться факторами, не подвластными инвестору в момент выбора объектов инвестирования. Подобные риски относятся к изменениям ситуации во внешнеэкономической деятельности, их принято называть систематическими [9].

**Систематические риски делятся на:**

* политический риск – угроза негативного воздействия на рынок из-за смены правительства, войны и т.д.;
* экологический риск инвестиционного портфеля подразумевает под собой возможные убытки в случае природных катастроф или ухудшения экологической обстановки;
* инфляционный риск возникает в случае высокой [инфляции](https://fortrader.org/birzhevoj-slovar/inflyaciya_inflation.html), которая обесценит капитал инвестора;
* валютный риск может возникнуть по причине политических и экономических факторов, которые складываются в стране;
* изменение процентной ставки – это риск, при котором происходит снижение или повышение процентной ставки центрального банка страны, влекущее за собой изменения на рынке инфляций.

Суммарный риск инвестиционного портфеля складывается из всех вышеперечисленных. Для инвестора важна оценка не только отдельно взятых ценных бумаг, но и общий уровень риска на фондовом рынке и в экономике.

Помимо систематических рисков, на которые инвестор влиять не может, есть и несистематические риски, которые вызваны ошибками в управлении портфелем. Причиной этих рисков может быть непрофессиональная оценка инструментов инвестиционного портфеля в момент выбора инвестиционной политики, нерациональная структура инвестируемых средств. Повысив эффективность управления инвестиционной деятельности, можно избежать негативных последствий.

**Несистематические риски делятся на:**

* кредитный риск наступает в случае несоблюдения обязательств заемщиком или поручителями;
* отраслевой риск может быть связан с изменениями в определенной отрасли экономики;
* деловой риск связан с ошибками в управлении компаниями, в которые вложены деньги.

На российском рынке уже набрали популярность множество сервисов для подбора индивидуальных портфелей клиентов, ниже представлен список некоторых из них.

Мастер вкладов и инвестиций banki.ru. Воспользоваться сервисом просто: ответить на вопросы, указать сумму, которую готовы инвестировать, выбрать стратегию — и получить вариант уже сформированного портфеля. Портфель можно донастроить вручную, а начать инвестировать — практически сразу: «Мастер» сделает все необходимые заявки.

Сервис Yammi - простой инвестиционный портфель с умным помощником. Сервис был создан в результате партнерства «Яндекс.Денег» и FinEx. Умный помощник сервиса на основе ответов на вопросы в анкете разработает индивидуальную, оптимальную стратегию инвестирования. Минимальная сумма для входа — от 5000 рублей, комиссия 1,5% в год. Имеется возможность получить налоговый вычет с доходов.

Умный помощник для инвестиций сервиса Fins.money от БКС - сервис для получения пассивного дохода. Он анализирует финансовый рынок и выдает индивидуальные рекомендации и подбирает оптимальную стратегию каждому клиенту под планируемую им сумму инвестиций. Специальные знания не нужны, инвестировать и открывать вклады можно онлайн.  
Порог входа — 10 тысяч рублей.

Простые инвестиции Сбербанк - автоматический финансовый советник. Сервис автоматически анализирует информацию на фондовых рынках, рассчитывает план инвестиций и обеспечивает управление вложенными деньгами. В то же время эксперты следят за работой сервиса и готовы ответить на вопросы.

Яндекс.Деньги / Инвестиции БКС - совместный проект проект Яндекса и брокера БКС. Предоставляет возможность покупать акции российских и зарубежных компаний, валюту и облигации прямо в приложении Яндекс.Денег. В основе нового сервиса — инвестиционная платформа компании «БКС Брокер». Услуга доступна идентифицированным пользователям Яндекс.Денег с российским гражданством. Чтобы ею воспользоваться, нужно в приложении Яндекс.Денег открыть раздел «Инвестиции БКС» и создать брокерский счет.

1. Инструменты и технологии в проекте

Для реализации frontend-части проекта была выбрана JavaScript-библиотека React с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений.

React — это инструмент для создания пользовательских интерфейсов. Его главная задача — обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты – компоненты. Компонент React — это, если по-простому, участок кода, который представляет часть веб-страницы. Каждый компонент — это JavaScript-функция, которая возвращает кусок кода, представляющего фрагмент страницы. Для формирования страницы мы вызываем эти функции в определённом порядке, собираем вместе результаты вызовов и показываем их пользователю.

React использует язык разметки, называемый JSX, который похож на HTML, но работает внутри JavaScript, что отличает его от HTML.

С целью сделать разработку быстрее и эффективнее решено было использовать библиотеку Material UI. Material-ui — reactJS's фрэймворк, предоставляющий готовые Google решения для быстрой и довольно простой web разработки. Material-UI достаточно крупная библиотека, где ключевой частью react компонентов и стилизации является @material-ui/core (целью ее использования должен быть немаленький проект). В React JS material-ui — это часть Material Design. Material Design — это язык дизайна, впервые представленный Google в 2014 году. Это визуальный язык, который использует макеты на основе сетки, гибкую анимацию и переходы, дополнения и эффекты глубины, такие как освещение и тени. Цель Material Design сводится к трем вещам: Создание, Унификация и Настройка.

Создание — Material Design направлен на предоставление визуального языка, который синтезирует классические принципы дизайна. Унификация — он нацелен на разработку единой базовой системы, которая объединяет пользовательский интерфейс на разных платформах, устройствах и методах ввода. И Настройка обеспечивает визуальный язык и гибкую основу для инноваций и брендинга. Material design - язык визуальных образов, который не так давно создала корпорация Google для унификации интерфейсов всех ее продуктов и сервисов. Брендбук, который включает в себя все элементы данного направления в дизайне, постоянно развивается и дополняется, при этом сохраняя фундаментальные основы неизменными. Поверхности и края элементов в данном направлении дизайна сочетают визуальные образы и сигналы, создавая подсказки, помогающие пользователю интуитивно ориентироваться на сайте (как если бы все это происходило в реальном мире).

Кроме того, Material Design использует принципы полиграфического дизайна для эффектной расстановки акцентов (то есть фокусировки внимания на нужном элементе), упрощения навигации в интерфейсе, интуитивной передачи смысла его элементов. Для Material Design характерны насыщенные, ровные цвета, резкие, очерченные края, крупная типографика и немалые отступы между элементами. В совокупности эти элементы не просто складываются в приятную для глаз картинку, а создают новую реальность с концептуальным смыслом и множеством функций, которые дарят пользователю уникальный UX [10].

В случае с этим видом дизайна действия пользователя находятся в центре внимания. Все взаимодействия происходят в одном окружении, интерактивные объекты без прерывания последовательности переходят из одной среду в другую.

1. Этап формирования технического задания

Разработку любого проекта необходимо начинать с анализа его целей и задач и планирования реализации этапов. Поэтому мы на начальном этапе необходимо составить техническое Задание (ТЗ).

Создание информационной системы подбора индивидуальных инвестиционных предложений избавляет клиента от необходимости самостоятельного анализа рынка, автоматически определяя уровень риска. Для этого клиенту предлагается пройти тест с вопросами, который займет около двух минут. После прохождения теста сервис предложит наиболее простые инвестиционные инструменты.

Система анализирует ответы клиента, присваивает ему один из  трех уровней риска: консервативный, сбалансированный, агрессивный. На  основании риск-профиля подбираются наиболее подходящие для клиента финансовые продукты: акции, облигации, инвестиционные паи паевых инвестиционных фондов. На следующих этапах развития системы сразу после подбора продукта клиент может выбрать, открыть ли ему ИИС или брокерский счет.

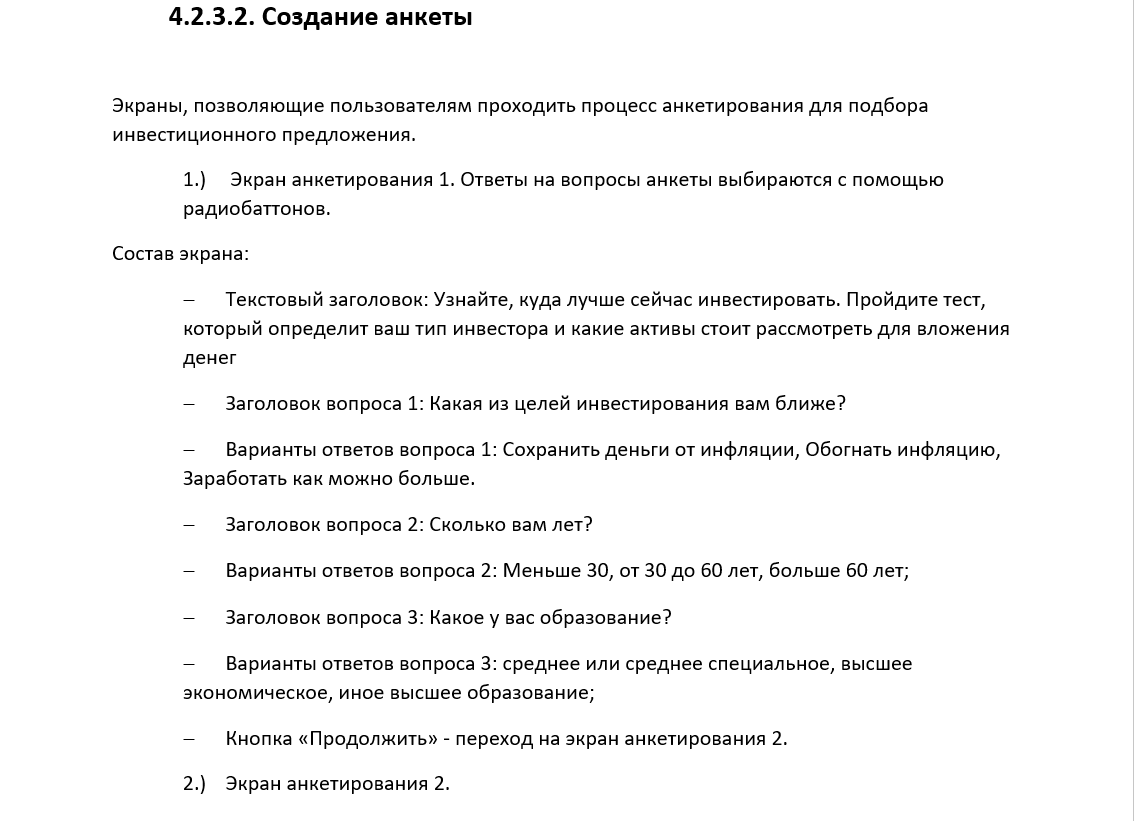


Рисунок 3. Фрагмент технического задания

Варианты типов инвесторов:

1 - Консервативный инвестор стремится избежать возможных финансовых потерь и сохранить вложенные средства, поэтому предпочтительно распределение портфеля в консервативных облигациях или структурных продуктах и золоте. Данная комбинация обеспечивает сохранность от инфляции и дает возможность получить доходность чуть выше, чем процентные ставки по депозиту;

2 - Сбалансированный инвестор имеет взвешенный подход к инвестированию, сочетая более рисковые активы - акции, либо структурные продукты без защиты капитала, и менее рисковые - облигации или золото. Такой инвестор готов принять умеренный уровень риска, чтобы потенциально заработать существенно больше, чем по вкладам. При этом сумма первоначальных вложений может меняться из-за колебаний на фондовом рынке, в том числе в отрицательную сторону в течение некоторого периода времени. Важно не забывать распределять средства в различные классы активов, движение которых идет разнонаправлено и сглаживает друг друга в моменты рыночных шоков, не давая портфелю просесть сильно;

3 - Умеренно-агрессивный инвестор готов принять высокий уровень риска, чтобы потенциально заработать в несколько раз больше, чем по вкладам. При этом сумма ваших вложений может существенно меняться из-за колебаний на фондовом рынке, в том числе в отрицательную сторону. Облигациям и инструментам с защитой капитала можно отвести меньшую долю и обратить внимание на акции и товарные рынки. Такое распределение активов поможет сгладить возможную просадку портфеля, но в случае роста рынков иметь возможность в нем полноценно участвовать.

1. Этап разработки и тестирования ИС

Для реализации frontend-части проекта была выбрана JavaScript-библиотека React с открытым исходным кодом для разработки пользовательских интерфейсов. С целью сделать разработку быстрее и эффективнее решено было использовать библиотеку Material UI.

Так как ранее опыта работы с компонентной разработкой у разработчика команды не было, предварительно и во время работы постепенно изучалась информация по работе с библиотекой при помощи официальной документации и примеров разработчиков.

Для формирования проекта библиотека устанавливается с помощью менеджера пакетов NPM. Для того, чтобы установить React с помощью npm, необходимо выполнить:

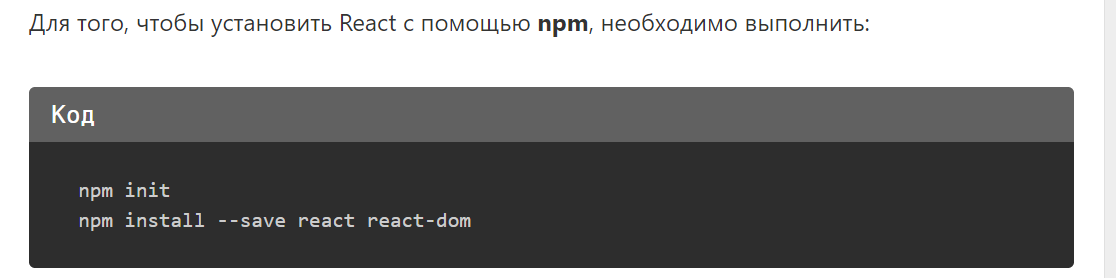


Рисунок 4. Команды для установки React

Для создания проекта со всеми необходимыми модулями, стандартными папками и сформированного файла package.json используется одна команда:

create-react-app

Итог создания приложения представлен на рисунке 5:

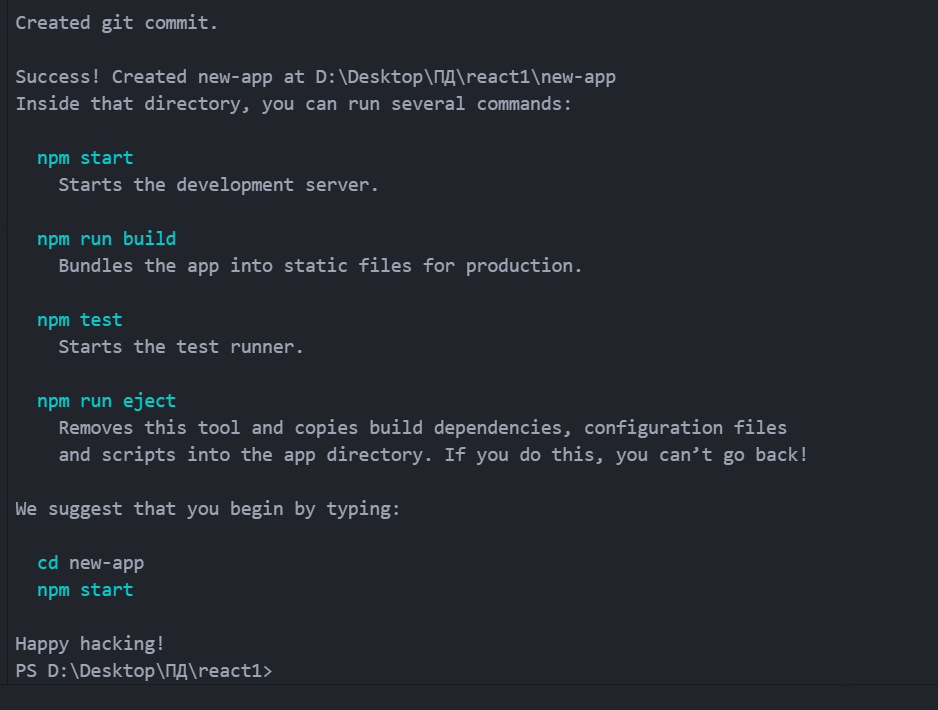


Рисунок 5. Приложение React создано

Для установки Material UI используется одна команда, всю необходимую информацию можно найти в официальной документации:

npm install @material-ui/core

Дальнейшая разработка велась в среде разработки Atom, при необходимости создавались компоненты и изменялась файловая структура приложения.

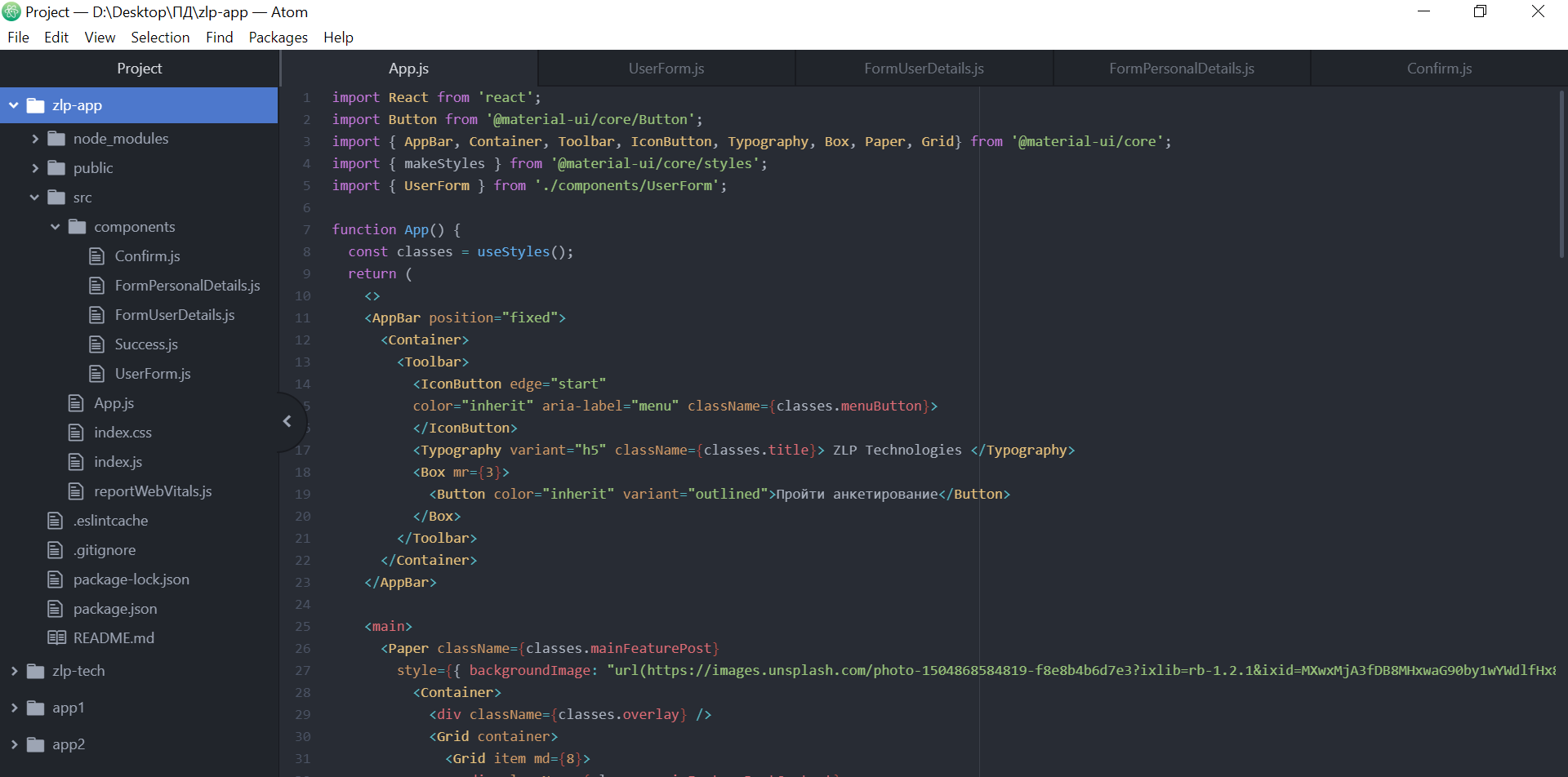


Рисунок 6. Фрагмент кода

Результат локального запуска проекта – первые два блока – это шапка сайта и краткое содержание информационной системы с предложением пройти тестирования для подбора портфеля:

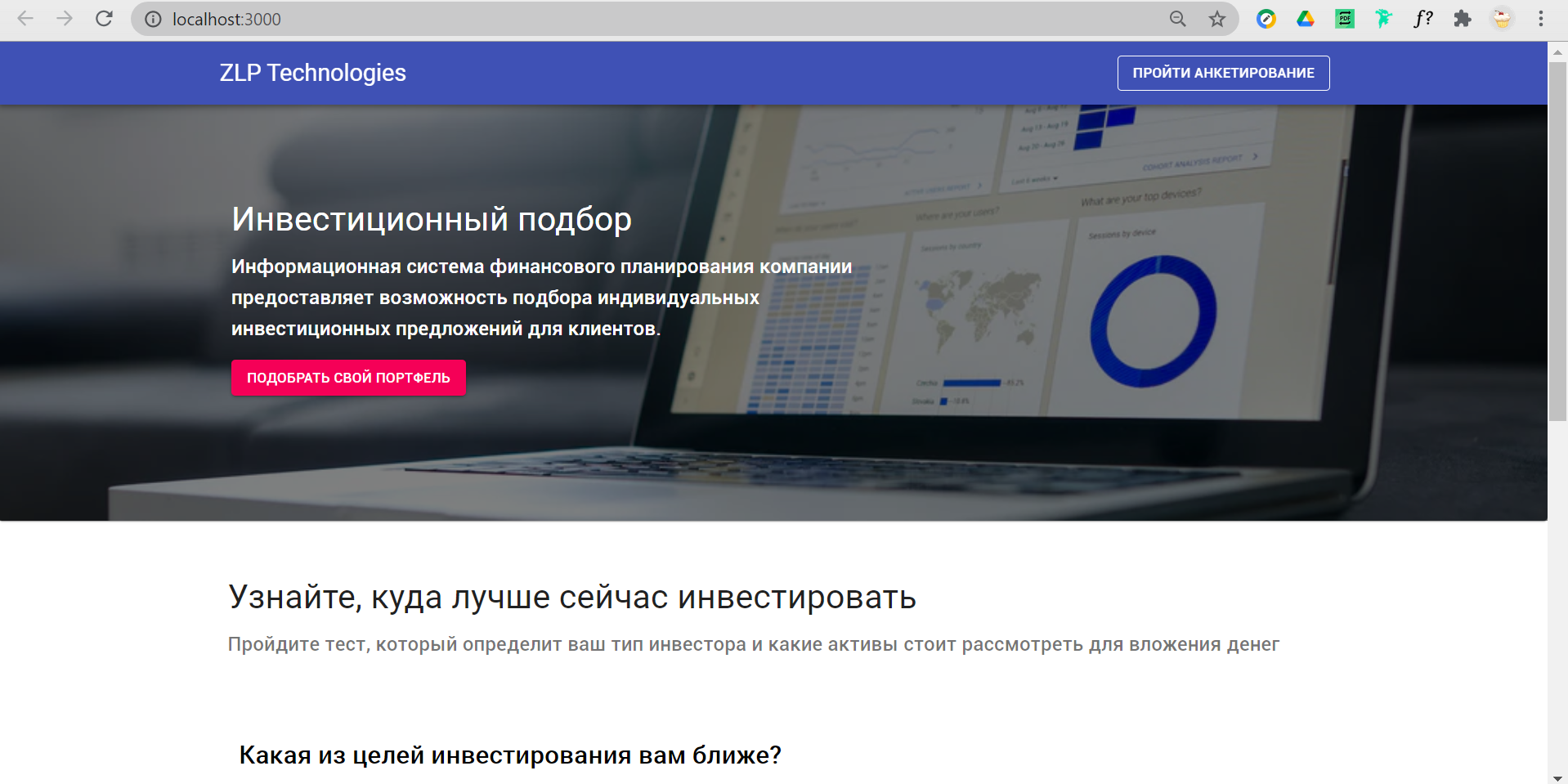


Рисунок 7. Шапка сайта

Ниже клиенту предоставляется возможность пройти анкетирование – разработано 3 шага анкетирования, с 3 вопросами в каждом, переход осуществляется по кнопкам «Следующий шаг» и «Назад»:

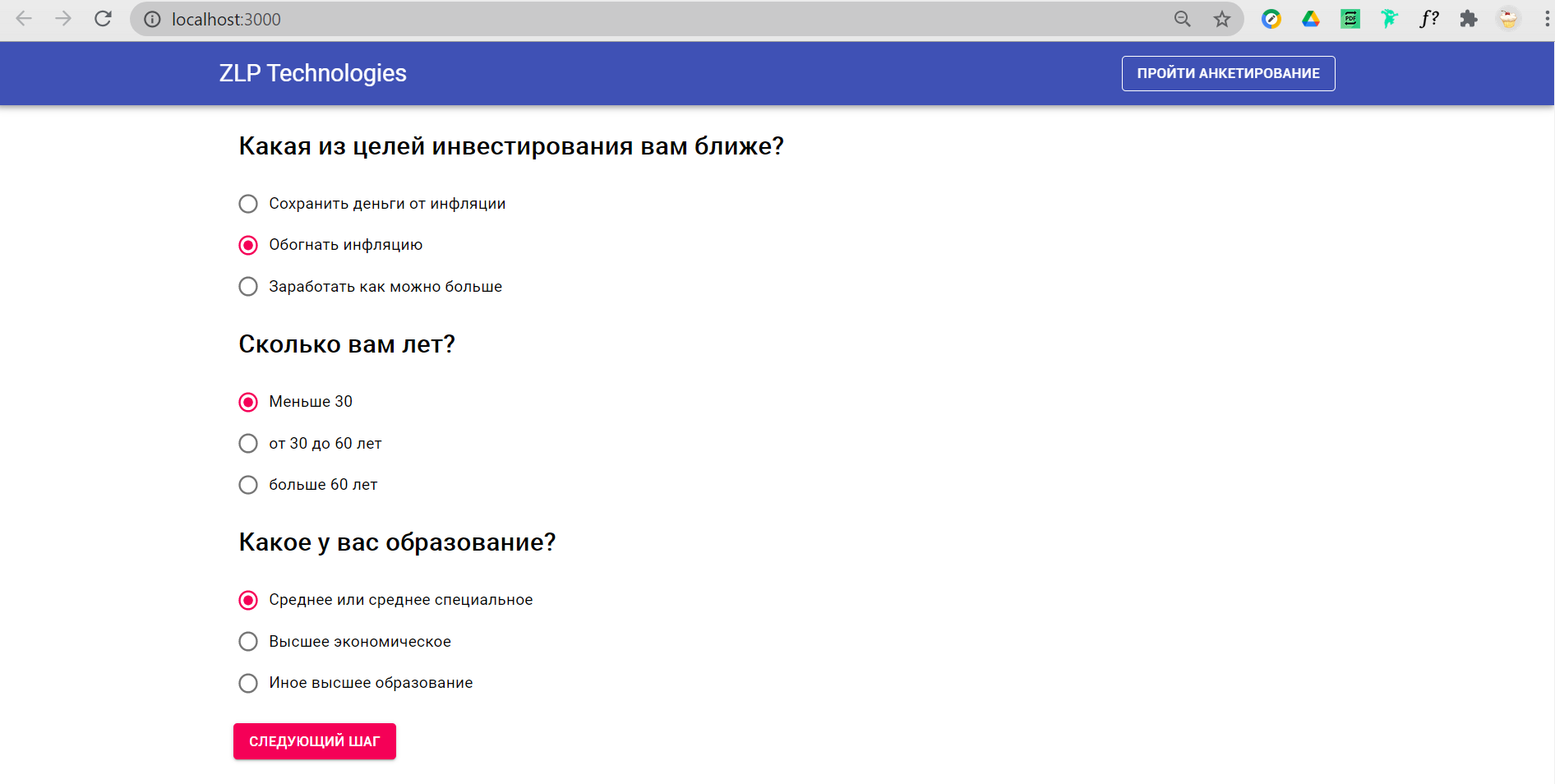


Рисунок 8. Анкета шаг 1

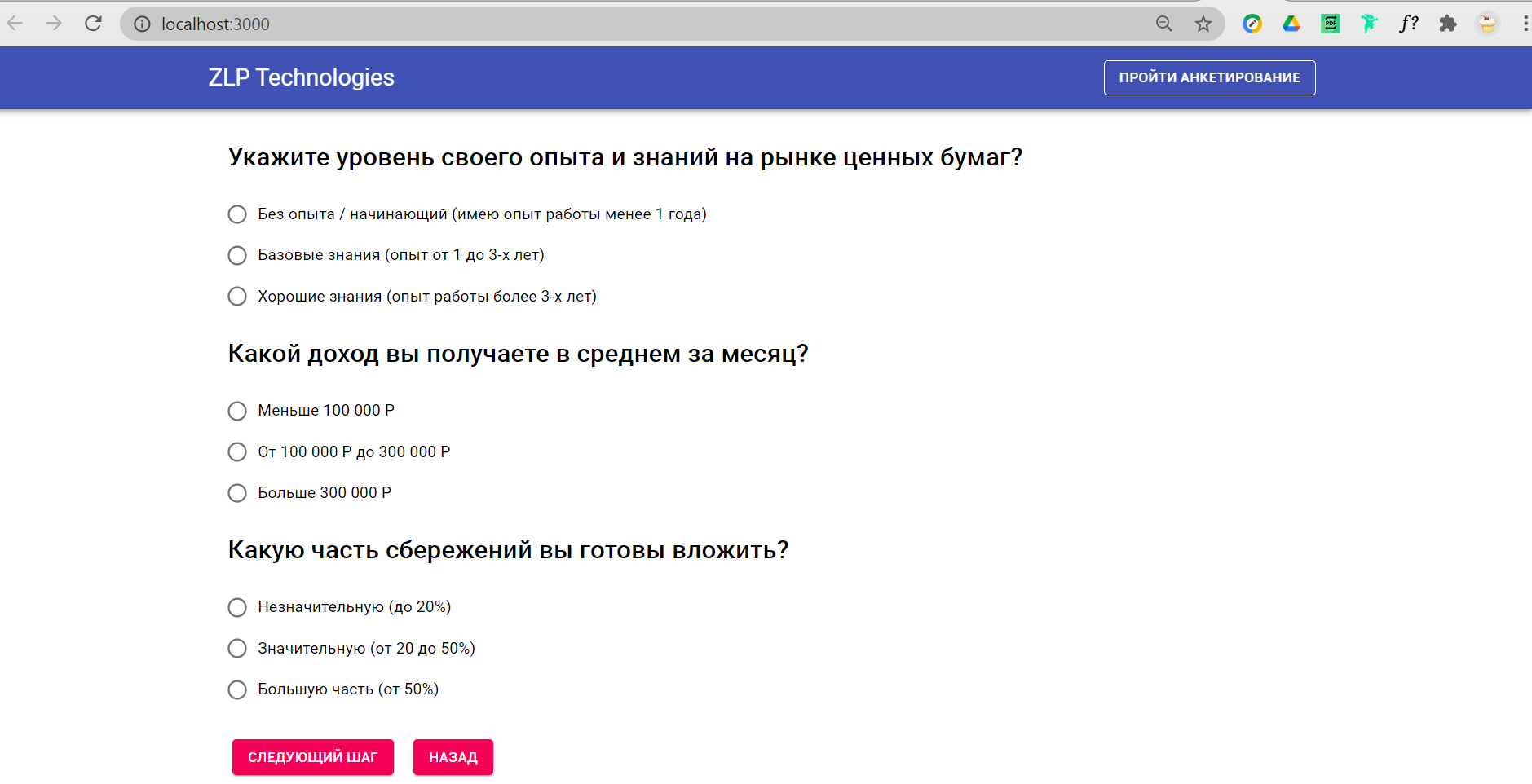


Рисунок 9. Анкета шаг 2

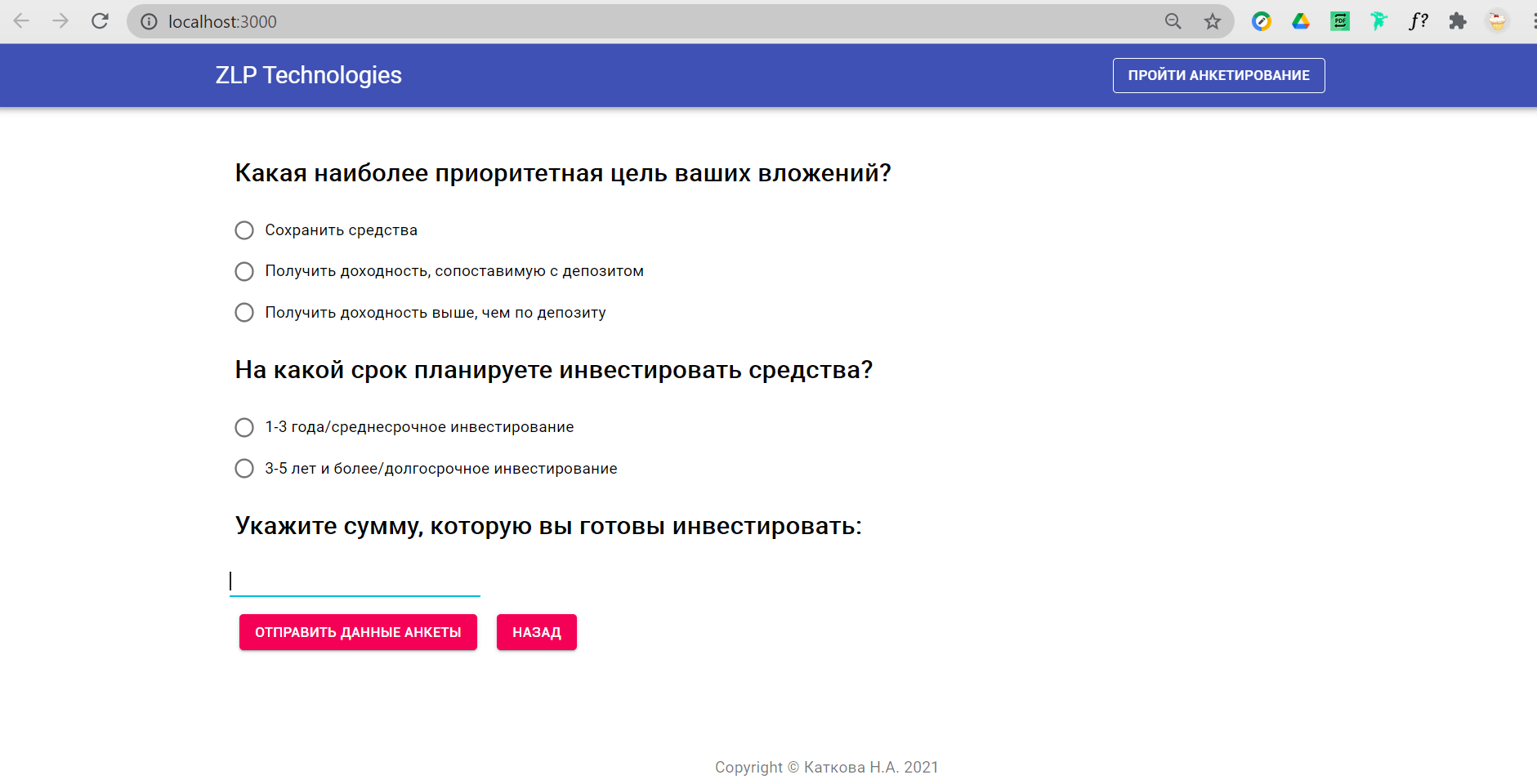


Рисунок 10. Анкета шаг 3

1. Обоснование выбора базы данных для будущих решений

На данном этапе проекта реализована фронтенд-часть анкетирования клиентов компании ZLP Tech, однако необходимо составить план дальнейших действий для развития проекта – выбрать стек backend части и используемую базу данных. Для этого мы провели сравнительный анализ и сделали выводы.

Существует несколько видов баз данных, различающихся по структуре:

* иерархические;
* сетевые;
* реляционные.

Под иерархической понимается такая база данных, в которой хранение данных и их структурирование осуществляется по принципу разделения элементов на родительские и дочерние. Преимуществом таких баз является лёгкость в чтении запрашиваемой информации и её быстрое предоставление пользователю.

Компьютер способен быстро ориентироваться в ней. Иерархический принцип взят за основу в структурировании файлов и папок в операционной системе Windows, а реестр хранит информацию о параметрах работы тех или иных приложений в структурированном иерархическим способом виде.

Все интернет-ресурсы также построены по иерархическому принципу, так как при его использовании ориентироваться в рамках сайта очень легко.

Под сетевой базой данных понимается модифицированная иерархическая. Её особенность заключается в том, что элементы могут быть связаны друг с другом в нарушение иерархии. То есть дочерний элемент одновременно может иметь несколько предков.

Под реляционным типом баз данных понимается их представление в рамках двумерной таблицы. Она имеет несколько столбцов, в которых устанавливаются такие параметры, как, например, тип вводимых данных (текст, число, дата и др.). Таблица здесь является способом хранения введённых в неё данных и способна реагировать на любые обращения со стороны СУБД. Главная проблема в работе с реляционными базами данных состоит в их правильном проектировании.

Во время проектирования базы данных следует учесть следующие два фактора:

* база данных должна быть компактной и не содержать избыточных компонентов;
* обработка базы данных должны происходить просто.

В крупных проектах задействовано множество таблиц, которых может быть более сотни. При этом обойтись без них невозможно, если человек имеет дело с важным и сложным проектом.

Система управления базами данных представляет собой встраивыемый модуль или полноценную программу, которая способна работать с данными и вносить изменения в базы.

Существует две модели СУБД - реляционная и бессхемная. Бессхемные СУБД, основанные на принципах неструктурированного подхода, избавляют программиста от проблем реляционной модели, в число которых входит низкая производительность и трудное масштабирование данных в горизонтальном формате.

Неструктурированные базы данных (NoSQL) создают структуру по ходу и убирают необходимость в создании жёстко определённых связей между данными. Здесь можно экспериментировать с разными способами доступа к тем или иным видам данных.

К реляционным базам данных относятся:

* SQLite;
* MySQL;
* PostgreSQL.

Из них наиболее распространённой является база данных MySQL. Принцип работы таких систем заключается в слежении за строгой структурой данных, которая представлена в виде комплекса таблиц. В свою очередь, внутри таблицы есть ячейки и поля, которыми также управляет MySQL. По принципу NoSQL работает база данных MongoDB. Они хранят все данные как единое целое в одной базе. При этом данные могут быть и одиночным объектом, но в то же время любой запрос не останется без ответа.

Каждая NoSQL имеет собственную систему запросов, что требует дополнительного изучения данной системы.

Сравнение SQL и NoSQL:

1. Если SQL-системы основаны исключительно на строгом представлении данных, то NoSQL-системы предоставляют свободу и способны работать с любым типом данных.
2. SQL-системы стандартизированы, за счёт чего запросы формируются с использованием языка SQL. В то же время NoSQL-системы базируются на специфической для каждой из них технологии, что является недостатком.
3. Масштабируемость. Обе СУБД способны обеспечить вертикальное масштабирование, то есть увеличить объём системных ресурсов на обработку данных. При этом NoSQL, будучи более новой разновидностью баз данных, позволяет применять простые методы горизонтального масштабирования.
4. В плане надёжности SQL обладает уверенным лидерством.
5. У SQL-баз есть качественная техническая поддержка за счёт их продолжительной истории, в то время как NoSQL-системы весьма молоды и и решить какую-либо проблему сложнее.
6. Хранение данных и доступ к их структурам в рамках реляционных систем лучше всего происходит в SQL-системах.

Таким образом, хоть NoSQL и является стремительно развивающейся разновидностью систем управления базами данных, однако на данном этапе рекомендуется остановить свой выбор на SQL [11].

Теперь рассмотрим виды SQL-систем. Выше были упомянуты SQLite, MySQL и PostgreSQL. Кратко их можно охарактеризовать так:

* SQLite - очень мощная встраиваемая система управления;
* MySQL - самая популярная и распространённая СУБД;
* PostgreSQL - наиболее продвинутая СУБД.

SQLite — легко встраиваемая в приложения база данных. Так как это система базируется на файлах, то она предоставляет довольно широкий набор инструментов для работы с ней, по сравнению с сетевыми СУБД. При работе с этой СУБД обращения происходят напрямую к файлам (в эти файлах хранятся данные), вместо портов и сокетов в сетевых СУБД. Именно поэтому SQLite очень быстрая, а также мощная благодаря технологиям обслуживающих библиотек.

Когда стоит использовать SQLite:

* Встроенные приложения: все портируемые не предназначенные для масштабирования приложения — например, локальные однопользовательские приложения, мобильные приложения или игры.
* Система доступа к дисковой памяти: в большинстве случаев приложения, часто производящие прямые операции чтения/записи на диск, можно перевести на SQLite для повышения производительности.
* Тестирование: отлично подойдёт для большинства приложений, частью функционала которых является тестирование бизнес-логики.

Когда не стоит использовать SQLite:

* Многопользовательские приложения: если вы работаете над приложением, доступом к БД в котором будут одновременно пользоваться несколько человек, лучше выбрать полнофункциональную РСУБД — например, MySQL.
* Приложения, записывающие большие объёмы данных: одним из ограничений SQLite являются операции записи. Эта РСУБД допускает единовременное исполнение лишь одной операции записи.

MySQL - это самая распространенная полноценная серверная СУБД. MySQL очень функциональная, свободно распространяемая СУБД, которая успешно работает с различными сайтами и веб приложениями. Обучиться использованию этой СУБД довольно просто, так как на просторах интернета существует большое количество информации. Благодаря популярности этой СУБД существует огромное количество различных плагинов и расширений, облегчающих работу с системой.

Несмотря на то, что в ней не реализован весь SQL функционал, MySQL предлагает довольно много инструментов для разработки приложений. Так как это серверная СУБД, приложения для доступа к данным, в отличие от SQLite, работают со службами MySQL.

Когда стоит использовать MySQL:

* Распределённые операции: когда нужен функционал бо́льший, чем может предоставить SQLite, стоит использовать MySQL.
* Высокая безопасность: функции безопасности MySQL предоставляют надёжную защиту доступа и использования данных.
* Веб-сайты и приложения: большая часть веб-ресурсов вполне может работать с MySQL, несмотря на ограничения. Этот инструмент весьма гибок и прост в обращении, что только на руку в длительной перспективе.
* Кастомные решения: если вы работаете над очень специфичным продуктом, MySQL подстроится под ваши потребности благодаря широкому спектру настроек и режимов работы.

Когда не стоит использовать MySQL:

* SQL-совместимость: поскольку MySQL не пытается полностью реализовать стандарты SQL, она не является полностью совместимой с SQL. Из-за этого могут возникнуть проблемы при интеграции с другими РСУБД.
* Конкурентность: хотя MySQL неплохо справляется с операциями чтения, одновременные операции чтения-записи могут вызвать проблемы.
* Недостаток функций: в зависимости от выбора движка MySQL может недоставать некоторых функций.

PostgreSQL является самым профессиональным из всех трех рассмотренных нами СУБД. Она свободно распространяемая и максимально соответствует стандартам SQL. PostgreSQL или Postgres стараются полностью применять ANSI/ISO SQL стандарты своевременно с выходом новых версий.

От других СУБД PostgreSQL отличается поддержкой востребованного объектно-ориентированного и/или реляционного подхода к базам данных. Например, полная поддержка надежных транзакций, т.е. атомарность, последовательность, изоляционность, прочность (Atomicity, Consistency, Isolation, Durability - ACID). Благодаря мощным технологиям Postgre очень производительна. Параллельность достигнута не за счет блокировки операций чтения, а благодаря реализации управления многовариантным параллелизмом (MVCC), что также обеспечивает соответствие ACID. PostgreSQL очень легко расширять своими процедурами, которые называются хранимые процедуры. Эти функции упрощают использование постоянно повторяемых операций [12].

Когда стоит использовать PostgreSQL:

* Целостность данных: если приоритет стоит на надёжность и целостность данных, PostgreSQL — лучший выбор.
* Сложные процедуры: если БД должна выполнять сложные процедуры, стоит выбрать PostgreSQL в силу её расширяемости.
* Интеграция: если в будущем предстоит перемещать всю базу на другое решение, меньше всего проблем возникнет с PostgreSQL.

Когда не стоит использовать PostgreSQL:

* Скорость: если всё, что нужно — это быстрые операции чтения, не стоит использовать PostgreSQL.
* Простые ситуации: если не требуется повышенная надёжность, поддержка ACID и так далее, использовать PostgreSQL ни к чему [13].

В итоге, рассмотрев все варианты, мы решили использовать для нашего проекта MySQL. В этой системе нас привлекло обилие справочного материала, простота в обращении, совместимость со многими веб-ресурсами и приложениями, большое количество инструментов для разработки приложений и приемлемая скорость работы.

1. Подведение итогов и результаты работ

В результате выполнения проекта были выполнены следующие задачи:

1. Освоен и организован репозиторий в GitHub;
2. Разработан вариант технического задания для проекта;
3. Проведен анализ систем для подбора финансовых стратегий клиентов;
4. Разработан frontend части информационной системы – анкеты для подбора персонального портфеля, освоена библиотека React.js;
5. Выбрана база данных для дальнейшего развития проекта;
6. Разработан презентационный веб-сайт;
7. Сформирован отчет по проделанной работе;
8. Подготовлен видеоматериала для итоговой защиты проекта;
9. Все необходимые ссылки и документы проекта также размещены в репозитории.

Ссылка на репозиторий в GitHub (приватный доступ): <https://github.com/natalikatkowa/zlp_tech>;

Ссылка на презентационный веб-сайт проекта, размещенный на хостинге Московского Политеха:

<http://pd-2020-1.std-1247.ist.mospolytech.ru/index0.html>;

Ссылка на проект, размещенный на хостинге Московского Политеха: <http://zlp.std-1247.ist.mospolytech.ru/>

Скриншоты презентационного веб-сайта:



Рисунок 11. Веб-сайт проекта - шапка

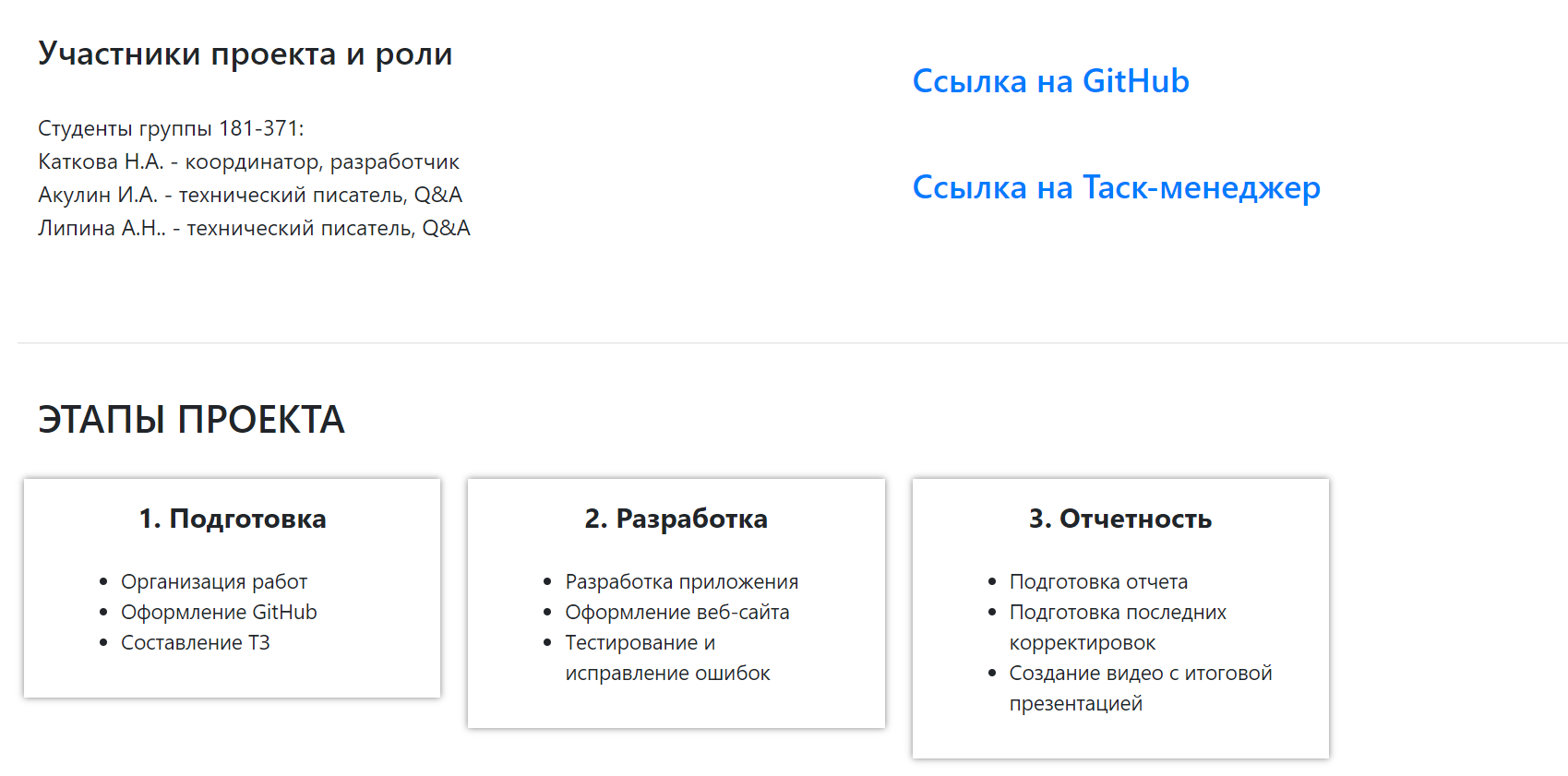


Рисунок 12. Веб-сайт проекта - содержание

# ЗАКЛЮЧЕНИЕ

В результате выполнения проекта, на данном этапе был разработан frontend информационной системы для финансового планирования, предоставляющей услугу подбора индивидуальных инвестиционных предложений клиентам компании ZLP Tech – анкета для подбора предложений клиенту и веб-сайт.

Вся работа организована в удаленном репозитории GitHub, координация этапов и задач каждого участника команды осуществлялась при помощи таск-трекера Trello с использованием канбан-доски.

Также создан презентационный веб-сайт проекта, сформировано техническое задание и итоговый отчет по проекту – все документы и ссылки размещены в репозитории GitHub.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Эффективное применение информационных систем [Электронный ресурс] – URL: <https://www.cfin.ru/itm/kis/choose/CAPE_PAEI.shtml> (дата обращения 29.01.2021);
2. Информационная система предприятия [Электронный ресурс] – URL: <http://itsiberia.ru/ru/services/details/service/information-systems> (дата обращения 29.01.2021);
3. Сервисы для инвестиций [Электронный ресурс] – URL: <https://businesgram.ru/dokhody/drugie-istochniki-dokhodov/item/292-servisy-dlya-investitsij> (дата обращения 29.01.2021);
4. Что нужно, чтобы инвестировать как профессионал [Электронный ресурс] – URL: <https://bcs.ru/blog/chto-nuzhno-chtoby-investirovat-kak-profi> (дата обращения 29.01.2021);
5. Исследование: самые популярные сервисы для инвестиций [Электронный ресурс] – URL: <https://www.sravni.ru/text/2019/11/26/issledovanie-samye-populjarnye-servisy-dlja-investicij/> (дата обращения 29.01.2021);
6. Как работают биржевые фонды [Электронный ресурс] – URL: <https://journal.tinkoff.ru/etfs/> (дата обращения 29.01.2021);
7. Инвестиционный портфель для начинающих [Электронный ресурс] – URL: <https://bcspremier.ru/knowledge/basics/investicionnyj-portfel-dlya-nachinayushchikh/> (дата обращения 29.01.2021);
8. Что такое инвестиционный портфель [Электронный ресурс] – URL: <https://berg.com.ua/basic/portfolio/> (дата обращения 29.01.2021);
9. Инвестиционный портфель [Электронный ресурс] – URL: <https://fortrader.org/birzhevoj-slovar/ekonomicheskie-ponyatiya/investicionnyj-portfel.html> (дата обращения 29.01.2021);
10. Все, что нужно знать о Material Design [Электронный ресурс] – URL: <https://lpgenerator.ru/blog/2016/06/16/vse-chto-vam-nuzhno-znat-o-material-design/> (дата обращения 29.01.2021);
11. Базы данных: какие бывают [Электронный ресурс] – URL: https://webshake.ru/post/bazy-dannyh (дата обращения 29.01.2021);
12. SQLite vs MySQL vs PostgreSQL: сравнение систем управления базами данных | DevAcademy [Электронный ресурс] – URL: https://devacademy.ru/article/sqlite-vs-mysql-vs-postgresql/ (дата обращения 29.01.2021);
13. SQLite, MySQL и PostgreSQL: сравниваем популярные реляционные СУБД [Электронный ресурс] – URL: https://tproger.ru/translations/sqlite-mysql-postgresql-comparison/ (дата обращения 29.01.2021).