

I. Подготовка рабочего окружения (Шаги 1–10)

1. Понять задачу и собрать материалы.

Описание: Прочитайте задание ещё раз, составьте список требований (страницы, авторизация, загрузка PDF, дизайн, админка).

Время: 3 минуты

2. Узнать, что такое Node.js и зачем он нужен.

Описание: Посмотрите короткое видео или почитайте статью о Node.js (это нужно для работы с JavaScript-средой).

Время: 5 минут

3. Установить Node.js.

Описание: Перейдите на [официальный сайт Node.js](https://nodejs.org/) и скачайте LTS-версию для вашей операционной системы, затем установите её.

Время: 10 минут

4. Установить менеджер пакетов npm (он устанавливается вместе с Node.js).

Описание: Убедитесь, что npm установлен, введя в терминале команду `npm -v`.

Время: 2 минуты

5. Установить редактор кода – Visual Studio Code.

Описание: Скачайте VS Code с [официального сайта](https://code.visualstudio.com/) и установите его.

Время: 10 минут

6. Открыть VS Code и настроить базовые плагины.

Описание: Рекомендуется установить плагины для подсветки синтаксиса JavaScript/React и автодополнения (например, ESLint, Prettier).

Время: 5 минут

7. Открыть терминал (встроенный в VS Code или системный).

Описание: Ознакомьтесь, как открывать терминал в VS Code (через меню «View > Terminal»).

Время: 2 минуты

8. Создать новую папку для проекта на вашем компьютере.

Описание: Назовите папку, например, `research-portal`.

Время: 2 минуты

9. Инициализировать новый React-проект с помощью create-react-app.

Описание: В терминале выполните команду

```
lua
Copy
npx create-react-app research-portal
```

Совет нейросети: Если нужна помощь, спросите:

"Напиши инструкцию по использованию create-react-app для создания нового React-проекта."

Время: 10 минут

10. Запустить созданный проект.

Описание: Перейдите в папку проекта (`cd research-portal`) и выполните команду `npm start`, чтобы увидеть стартовую страницу в браузере.

Время: 5 минут

II. Первоначальная настройка проекта и структуры (Шаги 11–25)

11. Ознакомиться со структурой проекта.

Описание: Просмотрите файлы и папки в созданном проекте (папка `src`, `public` и т.д.).

Время: 3 минуты

12. Очистить стартовую страницу.

Описание: Удалите лишний код из `App.js`, чтобы начать с чистого листа.

Время: 5 минут

13. Установить библиотеку `react-router-dom` для навигации.

Описание: В терминале выполните команду:

```
nginx
Copy
npm install react-router-dom
```

Время: 3 минуты

14. Создать базовую маршрутизацию.

Описание: Настройте в `App.js` роуты для страниц (например, `Home`, `Login`, `Register`, `Research` и `Admin`).

Совет нейросети:

"Напиши пример базовой маршрутизации в React с использованием `react-router-dom`."

Время: 7 минут

15. Создать компоненты страниц.

Описание: Создайте отдельные файлы для страниц:

- `Home.js` – главная страница
- `Login.js` – страница входа
- `Register.js` – страница регистрации
- `Research.js` – список исследований
- `AdminDashboard.js` – панель администратора

Время: 10 минут

16. Создать общие компоненты (`Header`, `Footer`, `Navigation`).

Описание: Создайте компоненты для шапки и подвала сайта, чтобы повторно использовать их на всех страницах.

Время: 8 минут

17. Настроить навигационное меню.

Описание: Добавьте ссылки в `Header`, чтобы пользователи могли переходить между страницами.

Время: 5 минут

18. Добавить страницу «Not Found».

Описание: Создайте компонент для отображения ошибки 404 при неверном URL.

Время: 5 минут

19. Проверить работу маршрутизации.

Описание: Запустите проект и протестируйте переход между созданными страницами.

Время: 5 минут

20. Подключить CSS-фреймворк для быстрого и современного дизайна (например, `Bootstrap` или `Material-UI`).

Описание:

- Если выбираете `Bootstrap`, установите его через `npm`:

```
nginx
Copy
npm install bootstrap
```

- Подключите стили в `index.js` или `App.js`:

```
js
Copy
import 'bootstrap/dist/css/bootstrap.min.css';
```

21. *Время:* 7 минут

22. **Создать адаптивную навигационную панель с Bootstrap.**

Описание: Используйте компоненты Bootstrap для создания современного меню.

Совет нейросети:

"Напиши пример адаптивной навигационной панели на React с использованием Bootstrap."

Время: 8 минут

23. **Создать базовый шаблон для всех страниц.**

Описание: Оберните содержимое страниц в компонент `Layout`, где будут подключаться `Header` и `Footer`.

Время: 7 минут

24. **Добавить базовую систему стилей.**

Описание: Создайте файл `App.css` и подключите его для общей стилизации (цвета, отступы, шрифты).

Время: 5 минут

25. **Провести первоначальное тестирование интерфейса.**

Описание: Пройдитесь по всем страницам, чтобы убедиться, что маршрутизация и базовая структура работают.

Время: 5 минут

26. **Сохранить все изменения в системе контроля версий (например, git).**

Описание: Если не знакомы с `git`, можно создать локальный репозиторий, выполнив команды `git init`, `git add .` и `git commit -m "Initial commit"`.

Время: 5 минут

III. Реализация регистрации и авторизации (Шаги 26–45)

26. **Создать форму регистрации.**

Описание: В файле `Register.js` создайте форму с полями: имя, email, пароль.

Время: 7 минут

27. **Добавить валидацию формы регистрации.**

Описание: Используйте стандартные HTML-атрибуты (например, `required`) или простую JS-валидацию.

Совет нейросети:

"Напиши пример React-формы регистрации с базовой валидацией полей."

Время: 7 минут

28. **Настроить сохранение данных регистрации во временном хранилище (Local Storage).**

Описание: При отправке формы сохраните данные пользователя в Local Storage (это будет имитацией регистрации для MVP).

Время: 7 минут

29. Добавить уведомление после регистрации.

Описание: После регистрации покажите сообщение: «Спасибо за регистрацию, с вами скоро свяжутся менеджеры!».

Время: 4 минуты

30. Создать форму входа (Login).

Описание: В файле `Login.js` создайте форму для ввода email и пароля.

Совет нейросети:

"Напиши пример кода для формы входа в React."

Время: 7 минут

31. Настроить процесс входа.

Описание: При вводе данных сравнивайте их с данными из Local Storage, и если всё совпадает, сохраняйте состояние «авторизован».

Время: 8 минут

32. Создать контекст (Context API) для управления состоянием авторизации.

Описание: Создайте файл, например, `AuthContext.js`, который будет хранить информацию о текущем пользователе.

Время: 10 минут

33. Создать защищённый маршрут (ProtectedRoute).

Описание: Напишите компонент, который проверяет, авторизован ли пользователь, и если нет – перенаправляет на страницу входа.

Совет нейросети:

"Напиши код для защищённого маршрута в React с использованием Context API."

Время: 7 минут

34. Протестировать вход и выход.

Описание: Проверьте, что после входа пользователь получает доступ к защищённым страницам, а при выходе – доступ закрывается.

Время: 5 минут

35. Создать кнопку выхода (Logout) в Header.

Описание: Добавьте кнопку, при нажатии на которую состояние авторизации очищается.

Время: 5 минут

36. Отобразить имя пользователя в интерфейсе (если необходимо).

Описание: После входа можно показывать имя пользователя в навигации.

Время: 4 минуты

37. Добавить обработку ошибок при входе (неверный пароль/нет пользователя).

Описание: Выведите понятное сообщение об ошибке, если введены неправильные данные.

Время: 5 минут

38. Сохранить изменения в системе контроля версий.

Описание: Сделайте коммит с сообщением, например, «Добавлена регистрация и авторизация».

Время: 3 минуты

39. Проверить работу регистрации и авторизации на разных браузерах (при необходимости).

Описание: Протестируйте работу функций в Chrome, Firefox или другом браузере.

Время: 5 минут

40. Создать базовую структуру для хранения данных исследований.

Описание: В проекте создайте файл (например, `data.js` или `researchData.json`) с примерными данными исследований.

Время: 7 минут

41. Наполнить файл исследованиями.

Описание: Добавьте несколько примеров отчётов с полями: название, тема (рынок пшеницы, стали, макрорепорт и т.д.), периодичность, ссылка на PDF.

Совет нейросети:

"Напиши пример JSON-структуры с данными исследований для research-портала."

Время: 7 минут

42. Создать страницу для отображения списка исследований.

Описание: В файле `Research.js` напишите компонент, который выводит список исследований из файла данных.

Время: 10 минут

43. Реализовать фильтрацию исследований по темам и периодичности.

Описание: Добавьте кнопки или выпадающие списки для фильтрации списка отчётов.

Время: 10 минут

44. Добавить ссылку для просмотра отдельного исследования.

Описание: Сделайте так, чтобы при клике по исследованию открывалась детальная страница (например, с отображением PDF).

Время: 7 минут

45. Создать страницу детального просмотра исследования.

Описание: Создайте компонент (например, `ResearchDetail.js`), который получает данные отчёта и показывает кнопку для открытия PDF-файла.

Совет нейросети:

"Напиши React-компонент, который принимает URL PDF-файла через props и отображает его в iframe."

Время: 8 минут

IV. Реализация функционала для администратора (Шаги 46–65)

46. Создать отдельный раздел для администратора.

Описание: Создайте компонент `AdminDashboard.js` с навигацией для админских функций.

Время: 7 минут

47. Добавить защищённый маршрут для доступа к админке.

Описание: Убедитесь, что доступ к `AdminDashboard` имеют только авторизованные администраторы (можно для MVP задать фиксированные данные).

Время: 7 минут

48. Создать форму входа для администратора (если хотите отделить её от обычного входа).

Описание: Можно переиспользовать форму входа, добавив проверку типа

пользователя (админ/пользователь).

Время: 5 минут

49. Добавить раздел «Загрузить PDF-отчёт».

Описание: В админке создайте страницу или раздел для загрузки новых исследований.

Время: 7 минут

50. Создать форму загрузки PDF-файла.

Описание: Добавьте элемент `<input type="file" />` и поля для ввода названия, темы, периодичности отчёта.

Время: 8 минут

51. Настроить обработчик формы загрузки.

Описание: Напишите функцию, которая при отправке формы считывает данные и «загружает» отчёт (для MVP можно сохранять в состоянии приложения).

Совет нейросети:

"Напиши код для обработки загрузки файла в React, который сохраняет данные в state."

Время: 8 минут

52. Сохранить загруженные отчёты в локальном состоянии (например, в массиве или Local Storage).

Описание: Обновите данные в вашем `researchData` после загрузки нового отчёта.

Время: 7 минут

53. Добавить возможность редактирования и удаления отчётов для администратора.

Описание: Создайте кнопки «Редактировать» и «Удалить» рядом с каждым отчётом в админском интерфейсе.

Время: 8 минут

54. Реализовать функцию удаления отчёта.

Описание: Напишите функцию, которая удаляет отчёт из списка.

Совет нейросети:

"Напиши функцию на JavaScript для удаления элемента из массива в React state."

Время: 7 минут

55. Добавить подтверждение удаления (например, через модальное окно).

Описание: Перед удалением спросите у администратора, уверен ли он, что хочет удалить отчёт.

Время: 7 минут

56. Добавить уведомления об успешной загрузке или удалении.

Описание: Выведите короткие сообщения (например, с помощью всплывающих уведомлений) об успешных операциях.

Совет нейросети:

"Напиши пример кода для показа уведомлений в React."

Время: 6 минут

57. Создать отдельный стиль для админки (AdminLayout).

Описание: Разработайте другой вариант оформления для страниц администратора, чтобы их отличать от пользовательских.

Время: 7 минут

58. **Проверить работу всех функций в админке.**
Описание: Протестируйте загрузку, редактирование и удаление отчётов.
Время: 8 минут
59. **Обеспечить сохранение данных отчётов (даже если страница перезагрузится) с помощью Local Storage.**
Описание: При изменении списка отчётов сохраняйте обновлённый массив в Local Storage.
Время: 7 минут
60. **Добавить кнопку «Выход» для администратора.**
Описание: Реализуйте возможность выйти из админского режима так же, как и для обычных пользователей.
Время: 5 минут
61. **Сохранить изменения в системе контроля версий.**
Описание: Сделайте коммит с описанием админских функций (например, «Добавлена админка: загрузка, редактирование и удаление PDF-отчётов»).
Время: 3 минуты
62. **Протестировать админский раздел на разных экранах.**
Описание: Проверьте, как выглядит админка на компьютере и в режиме мобильного просмотра.
Время: 5 минут
63. **Добавить небольшой текст-подсказку для админа, как пользоваться интерфейсом.**
Описание: Вставьте краткие инструкции на странице админки.
Время: 4 минуты
64. **Добавить базовые стили для форм и кнопок в админке.**
Описание: Используйте Bootstrap-классы или свои CSS-стили для улучшения внешнего вида.
Время: 5 минут
65. **Проверить интеграцию админских функций с остальным сайтом.**
Описание: Убедитесь, что переходы между пользовательским и админским режимами корректны.
Время: 5 минут
-

V. Доработка интерфейса и отзывчивого дизайна (Шаги 66–80)

66. **Настроить общую цветовую схему и шрифты для сайта.**
Описание: Выберите приятные цвета и шрифты, чтобы сайт выглядел современно и профессионально.
Время: 6 минут
67. **Оптимизировать CSS для адаптивности.**
Описание: Используйте flexbox, grid и media queries, чтобы сайт хорошо отображался на мобильных устройствах.
Время: 8 минут
68. **Спросить у нейросети примеры современных CSS-стилей.**
Совет нейросети:

"Напиши CSS-код для современной адаптивной навигационной панели."
Время: 4 минуты

69. Внедрить полученный CSS-код в проект.

Описание: Скопируйте и адаптируйте код под свой проект.

Время: 5 минут

70. Добавить анимацию для переходов между страницами.

Описание: Можно использовать CSS-переходы или библиотеку (например, React Transition Group) для плавного появления/исчезания элементов.

Время: 7 минут

71. Добавить загрузочный спиннер при ожидании данных.

Описание: При выполнении длительных операций (например, при авторизации) показывайте анимацию загрузки.

Совет нейросети:

"Напиши компонент спиннера для React."

Время: 6 минут

72. Интегрировать спиннер в процессы регистрации/входа и загрузки отчётов.

Описание: Обновите логику, чтобы во время выполнения операций отображался спиннер.

Время: 5 минут

73. Провести тестирование интерфейса на мобильных устройствах через инструменты разработчика браузера.

Описание: Проверьте, что всё отображается корректно на разных размерах экранов.

Время: 7 минут

74. Улучшить читаемость текста, добавив отступы и контрастные цвета.

Описание: Настройте CSS-стили для заголовков, абзацев и кнопок.

Время: 5 минут

75. Добавить иконки для навигации и кнопок (например, с помощью FontAwesome).

Описание: Установите пакет и подключите иконки, чтобы улучшить визуальное восприятие.

Время: 7 минут

76. Проверить кроссбраузерную совместимость.

Описание: Убедитесь, что сайт выглядит хорошо в разных браузерах.

Время: 5 минут

77. Попросить нейросеть советов по оптимизации отзывчивости сайта.

Совет нейросети:

"Дай советы по улучшению отзывчивости интерфейса React-приложения."

Время: 3 минуты

78. Исправить замечания, найденные при тестировании.

Описание: Внесите коррективы в CSS или разметку по результатам проверки.

Время: 7 минут

79. Добавить адаптивное меню для мобильных устройств.

Описание: Если меню на ПК выглядит хорошо, добавьте его адаптивную версию (например, «гамбургер-меню»).

Время: 7 минут

80. Провести финальное тестирование дизайна и интерфейса.

Описание: Пройдите по всем страницам, протестируйте все элементы и убедитесь, что пользовательский опыт на высоком уровне.

Время: 5 минут

VI. Завершающие шаги и подготовка к деплою (Шаги 81–100)

81. Перепроверить все маршруты и переходы между страницами.

Описание: Пройдите по всем ссылкам, чтобы не было «битых» переходов.

Время: 5 минут

82. Добавить модальное окно для подтверждения важных действий (например, после регистрации).

Совет нейросети:

"Напиши код для модального окна на React для подтверждения действий."

Время: 6 минут

83. Интегрировать модальное окно в процесс регистрации (показать сообщение «С вами свяжутся менеджеры»).

Время: 5 минут

84. Провести код-ревью: просмотреть весь код, убрать ненужные комментарии и файлы.

Время: 5 минут

85. Попросить нейросеть дать рекомендации по оптимизации кода React-приложения.

Совет нейросети:

"Дай рекомендации по оптимизации React-приложения для MVP."

Время: 3 минуты

86. Протестировать функциональность на предмет ошибок в консоли браузера.

Описание: Откройте консоль разработчика и исправьте найденные ошибки.

Время: 5 минут

87. Добавить краткую документацию для администратора (как загружать и управлять отчётами).

Время: 5 минут

88. Провести финальное тестирование функционала пользователя (регистрация, вход, просмотр исследований).

Время: 5 минут

89. Провести финальное тестирование функционала администратора (загрузка, редактирование, удаление отчётов).

Время: 5 минут

90. Убедиться, что все уведомления и сообщения отображаются корректно.

Время: 4 минуты

91. Сохранить финальную версию кода в систему контроля версий (git).

Время: 3 минуты

92. Подготовить проект к деплою: выполнить команду сборки проекта.

Описание: В терминале выполните команду

```
arduino  
Copy  
npm run build
```

Время: 7 минут

93. **Спросить у нейросети: «Как собрать React-проект для продакшн?» для получения дополнительных советов по сборке.**
Время: 3 минуты
94. **Выбрать бесплатную платформу для деплоя (например, Netlify или Vercel).**
Время: 5 минут
95. **Зарегистрироваться на выбранной платформе для деплоя (если у вас ещё нет аккаунта).**
Время: 5 минут
96. **Следовать инструкциям платформы для загрузки (деплой) вашего проекта.**
Описание: Обычно это просто: загрузка содержимого папки `build`.
Время: 7 минут
97. **Дождаться завершения процесса деплоя и получить URL вашего сайта.**
Время: 5 минут
98. **Протестировать развернутый сайт в браузере по URL.**
Время: 5 минут
99. **Спросить у нейросети советы по дальнейшему расширению функционала сайта.**
Совет нейросети:

"Какие возможности можно добавить в будущем для улучшения research-портала?"
Время: 3 минуты
100. **Отпраздновать завершение проекта и составить список будущих улучшений.**
Описание: Запишите, что удалось реализовать, и какие дополнения можно внести позже (например, подключение настоящей базы данных, интеграция с облачным хранилищем для PDF и т.д.).
Время: 5 минут