**Стратегия выполнения веб-приложения «GrindStone»**

**Цель:** Поэтапно разработать, протестировать и запустить полнофункциональный минимально жизнеспособный продукт (MVP).  
**Общий срок:** ~6-8 месяцев.

**Фаза 1: Проектирование и дизайн (Срок: ~3-4 недели)**

**Цель:** Создать детальный план и визуальную концепцию продукта. Полностью определить, *что* мы строим.

**Ключевые действия:**

1. **Проектирование архитектуры:**
   * Разработка схемы базы данных (диаграмма сущностей-связей для PostgreSQL). Определение таблиц (users, projects, tasks, files etc.) и связей между ними.
   * Проектирование API: составление списка всех необходимых эндпоинтов (например, GET /api/tasks, POST /api/projects), их входных и выходных данных в формате JSON.
2. **Создание прототипов в Figma:**
   * **Low-fidelity wireframes:** Быстрые наброски всех ключевых страниц (Лендинг, Дашборд, Страница проекта, Настройки) для определения компоновки элементов.
   * **High-fidelity макеты:** Детальные, pixel-perfect макеты в цвете, с типографикой и иконками. Создание дизайн-системы: кнопки, поля ввода, модальные окна, цветовая палитра (для светлой и темной темы).
   * **Интерактивный прототип:** Настройка переходов между экранами в Figma для симуляции пользовательского опыта.
3. **Планирование проекта:**
   * Разбиение всей функциональности из ТЗ на мелкие, executable задачи. Создание backlog в Trello, Notion или Jira.
   * Расстановка приоритетов для реализации MVP (минимальный набор функций для запуска).

**Результат фазы:**

* Утвержденные макеты всех экранов в Figma.
* Готовая схема базы данных.
* Полное описание API.
* Приоритизированный бэклог задач для разработки.

**Фаза 2: Подготовка среды и разработка ядра (Срок: ~2-3 месяца)**

**Цель:** Настроить всю инфраструктуру для разработки и реализовать фундаментальные, нефункциональные модули.

**Ключевые действия:**

1. **Настройка окружения и репозитория:**
   * Создание monorepo на GitHub.
   * Настройка окружений (development, production).
   * Инициализация Next.js и Nest.js проектов.
   * Настройка Docker-контейнеров для БД и приложений.
2. **Разработка базовых модулей:**
   * **Система аутентификации и авторизации:** Реализация регистрации, входа, JWT-токенов, защищенных маршрутов на бэкенде (Nest.js) и их интеграция с фронтендом (NextAuth.js).
   * **Работа с базой данных:** Написание схемы Prisma, создание сервисов для базовых CRUD-операций.
   * **Базовая UI-библиотека:** Создание набора переиспользуемых React-компонентов (кнопки, инпуты, модалки) на основе макетов из Figma с использованием Tailwind CSS.
3. **Настройка инфраструктуры:**
   * Настройка CI/CD (например, GitHub Actions) для автоматического тестирования и деплоя.
   * Настройка линтеров (ESLint) и форматтеров (Prettier) для поддержания качества кода.

**Результат фазы:**

* Рабочее локальное окружение.
* Пользователь может зарегистрироваться и войти в систему.
* Готовая библиотека UI-компонентов.
* Настроенный конвейер CI/CD.
* Пустое приложение с работающей базой данных.

**Фаза 3: Разработка ключевого функционала (MVP) (Срок: ~2-3 месяца)**

**Цель:** Реализовать основные функции продукта, определенные в ТЗ.

**Ключевые действия (Разработка ведется параллельно по модулям):**

1. **Модуль проектов:**
   * Реализация создания, редактирования, удаления проектов.
   * Разработка интерфейса списка проектов.
2. **Модуль задач (Kanban-доска):**
   * Создание, редактирование, удаление задач.
   * Реализация Drag-and-Drop функциональности для изменения статуса задачи (с помощью библиотеки @dnd-kit или react-beautiful-dnd).
   * Добавление в задачи атрибутов: исполнитель, срок, приоритет.
3. **Модуль базы знаний:**
   * **Загрузка файлов:** Реализация компонента для загрузки, бэкенд-логика для хранения файлов.
   * **Работа со ссылками:** Реализация парсинга мета-данных (Open Graph) для создания превью.
   * **Текстовый редактор:** Интеграция WYSIWYG-редактора (например, **Tiptap**) для создания wiki-страниц.
4. **Вспомогательные функции:**
   * Реализация глобального поиска.
   * Добавление переключения светлой/темной темы.

**Результат фазы:**

* **Готовое MVP:** Пользователь может создать проект, добавлять в него задачи на Kanban-доску, организовывать их, загружать файлы и создавать страницы.

**Фаза 4: Интеграционное тестирование и отладка (Срок: ~3-4 недели)**

**Цель:** Выявить и исправить ошибки, обеспечить стабильность и безопасность работы приложения.

**Ключевые действия:**

1. **Всестороннее тестирование:**
   * **Функциональное тестирование:** Пошаговая проверка всего функционала по тест-кейсам из «Программы и методик испытаний».
   * **Юзабилити-тестирование:** Привлечение 2-3 друзей/коллег для выполнения типовых сценариев (найти, создать, отредактировать). Сбор обратной связи по удобству.
   * **Нагрузочное тестирование:** Проверка работы API под нагрузкой (с помощью k6 или Artillery.io).
   * **Тестирование безопасности:** Проверка на уязвимости (например, OWASP ZAP).
2. **Исправление ошибок и оптимизация:**
   * Приоритизация и фикс всех обнаруженных багов.
   * Оптимизация медленных запросов к БД.
   * Оптимизация сборки и кода фронтенда (например, с помощью lighthouse-ci).

**Результат фазы:**

* Стабильная, протестированная версия приложения.
* Список исправленных критических и значительных ошибок.
* Отчеты по тестированию.

**Фаза 5: Деплой, запуск и обратная связь (Срок: ~2 недели)**

**Цель:** Вывести приложение в прод и начать сбор обратной связи от первых реальных пользователей.

**Ключевые действия:**

1. **Деплой в продакшен:**
   * Выбор хостинга (VPS like Selectel, или платформа like Railway).
   * Настройка сервера/контейнеров, доменного имени, SSL-сертификата.
   * Полное развертывание приложения и БД на продакшен-сервере.
2. **Подготовка к запуску:**
   * Наполнение базы демо-данными (примеры проектов и задач).
   * Написание краткого «Руководства по началу работы».
3. **Запуск и мониторинг:**
   * Открытие доступа для первого круга пользователей (альфа-тестеры).
   * Мониторинг ошибок (с помощью Sentry) и производительности.
   * Активный сбор и анализ обратной связи.

**Результат фазы:**

* **GrindStone** доступен в интернете по доменному имени.
* Первые пользователи работают с приложением.
  + Сформирован план дальнейших итераций и развития продукта на основе обратной связи.

**Итог:** Эта стратегия превращает сложный проект в последовательность manageable steps. Каждая фаза имеет четкую цель и измеримый результат, что позволяет контролировать прогресс и минимизировать риски.