

ЛИЦЕНЗИРОВАННЫЙ УЧЕБНЫЙ ЦЕНТР • ТЕРАКОР

Профессия JavaScript-разработчик

От основ языка и браузера до TypeScript, Vue 3, React и готовых frontend-приложений.

≈ 5 месяцев

длительность

450 ак. ч.

общий объём

150 ак. ч.

с преподавателем

3 проекта

Vue · React · итоговый

Как построена программа

Сохраняется траектория действующей программы: HTML/CSS → JavaScript → браузер → инструменты → TypeScript → HTTP → Vue → React → архитектура → качество и production. После каждого модуля студент выполняет практическое задание. За программу создаются три больших проекта.

150 ак. ч. занятий с преподавателем + 300 ак. ч. самостоятельной практики = 450 ак. ч. Формат: живые онлайн-занятия и самостоятельная работа между встречами.

Основные проекты: PulseDesk на Vue, CourseBoard на React и итоговая frontend-система.

Практические задания первых модулей готовят к разработке трёх проектов; следующие этапы показывают развитие инженерных навыков.

JavaScript · TypeScript · Vue 3 · React · REST API · Vitest · Docker

Модули 1-4

Каждый модуль завершается практическим заданием с разбором кода; крупных проектов за программу три.

МОДУЛЬ 01 / 8 АК. Ч.

HTML и CSS для разработчика

- Семантическая структура страниц, доступность, формы и валидация.
- Box model, Flexbox, CSS Grid, адаптивная вёрстка и DevTools.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Свёрстаный адаптивный экран с формой и проверкой доступности.

МОДУЛЬ 02 / 16 АК. Ч.

JavaScript: основы

- Типы, функции, область видимости, условия и циклы.
- Объекты, массивы, методы коллекций, деструктуризация, spread/rest, ES-модули.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Набор функций расчёта стоимости с проверкой входных данных.

МОДУЛЬ 03 / 16 АК. Ч.

JavaScript: углублённый уровень

- Execution context, замыкания, this, прототипы и классы.
- Event loop, Promise, async/await, обработка ошибок и декомпозиция кода.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Решение задач на замыкания, асинхронность и обработку событий.

МОДУЛЬ 04 / 10 АК. Ч.

DOM и браузер

- DOM, делегирование событий, формы и Browser API.
- Storage, Fetch, CORS, отладка запросов в Network и основы производительности.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Работающая интерактивная форма с DOM, событиями и localStorage.

Модули 5-8

Каждый модуль завершается практическим заданием с разбором кода; крупных проектов за программу три.

МОДУЛЬ 05 / 6 АК. Ч.

Git и инструменты разработки

- Git, GitHub, ветки, pull request, разрешение конфликтов и code review.
- npm, Vite, ESLint, Prettier, переменные среды и сборка.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Репозиторий практической работы: Vite, lint, история изменений и README.

МОДУЛЬ 06 / 12 АК. Ч.

TypeScript

- Типы, interfaces, unions, generics, narrowing и type guards.
- Utility types, типизация доменной модели, форм и ответов API.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Типы для заявок, пользователей, форм и ответов API будущих проектов.

МОДУЛЬ 07 / 6 АК. Ч.

HTTP и REST API

- Методы HTTP, статусы, заголовки, JSON, CORS и контракт API.
- Access/refresh tokens на концептуальном уровне, загрузка, ошибки, пагинация и фильтры.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Учебный клиент REST API с обработкой ошибок, загрузки и пагинации.

МОДУЛЬ 08 / 20 АК. Ч.

Vue 3

- SFC, Composition API, script setup, props, emits, слоты и composables.
- Vue Router, Pinia, формы, интеграция с API и защищённые маршруты.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ PulseDesk: панель заявок с ролями, CRUD, фильтрами, пагинацией, Pinia и адаптивным интерфейсом.

Модули 9-12

Каждый модуль завершается практическим заданием с разбором кода; крупных проектов за программу три.

МОДУЛЬ 09 / 22 АК. Ч.

React

- Компоненты, hooks, custom hooks, формы и маршрутизация.
- Context, Redux Toolkit, интеграция с API, разделение кода и основы производительности.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ CourseBoard: кабинет образовательного сервиса с программами, прогрессом, поиском и состоянием приложения.

МОДУЛЬ 10 / 10 АК. Ч.

Архитектура frontend

- Страницы, функции, сервисы, API-слой и границы компонентов.
- Локальное, глобальное и серверное состояние; SOLID, DRY и feature-based структура.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Архитектурная переработка проекта CourseBoard с описанием решений.

МОДУЛЬ 11 / 8 АК. Ч.

Тестирование

- Unit и component tests: Vitest, React Testing Library, Vue Test Utils.
- Тестирование форм, асинхронных состояний, ошибок и ключевых пользовательских сценариев.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Тесты критических сценариев одного из двух созданных проектов.

МОДУЛЬ 12 / 4 АК. Ч.

Безопасность frontend

- XSS, CSRF, cookies, токены и безопасный вывод пользовательских данных.
- Границы ответственности браузера и сервера при аутентификации.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Проверка безопасности существующего проекта и исправления в его коде.

Модули 13-14

Каждый модуль завершается практическим заданием с разбором кода; крупных проектов за программу три.

МОДУЛЬ 13 / 4 АК. Ч.

Производительность

- Lighthouse, Core Web Vitals, profiling, lazy loading и code splitting.
- Оптимизация изображений, загрузки и лишних перерисовок.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Замеры и оптимизация производительности CourseBoard.

МОДУЛЬ 14 / 8 АК. Ч.

Production, интервью и защита

- Docker и Nginx для SPA, базовый CI/CD и развёртывание — 4 ак. ч.
- Техническое собеседование: JavaScript, TypeScript, браузер, Vue/React — 2 ак. ч.
- Презентация и защита итогового проекта — 2 ак. ч.

ПРАКТИЧЕСКИЙ РЕЗУЛЬТАТ Итоговая frontend-система сервиса: React или Vue, TypeScript, роли, CRUD, учебный REST API, тесты, README и доступное демо.

Три проекта программы

01 / PulseDesk — Vue 3

Панель управления заявками для сервисной команды. Авторизация через учебный API, роли, создание и изменение заявки, статусы, поиск, фильтры, пагинация, формы, Vue Router и Pinia. Сдача: репозиторий, README, адаптивный интерфейс, опубликованное демо и разбор кода.

02 / CourseBoard — React

Личный кабинет образовательной платформы. Каталог программ, прогресс обучения, поиск и сортировка, маршруты, формы, React hooks, Redux Toolkit и работа с API. После модуля React приложение развивается: улучшаются архитектура, тесты, безопасность и производительность. Сдача: репозиторий, README, демо и краткое описание решений.

03 / Итоговая frontend-система

Индивидуальный сервис на TypeScript и выбранном фреймворке: Vue или React. Роли, CRUD, интеграция с учебным REST API, загрузка и ошибки, тесты, адаптивный интерфейс, production-сборка и публикация. На защите студент показывает сценарии работы, объясняет архитектуру и отвечает на вопросы по своему коду.

Объём программы

150 ак. ч. с преподавателем + 300 ак. ч. самостоятельной практики = 450 ак. ч. Сохраняются 14 модулей; после каждого — задание или этап одного из трёх проектов.

ТЕРАКОР • terakor.ru • info@terakor.ru