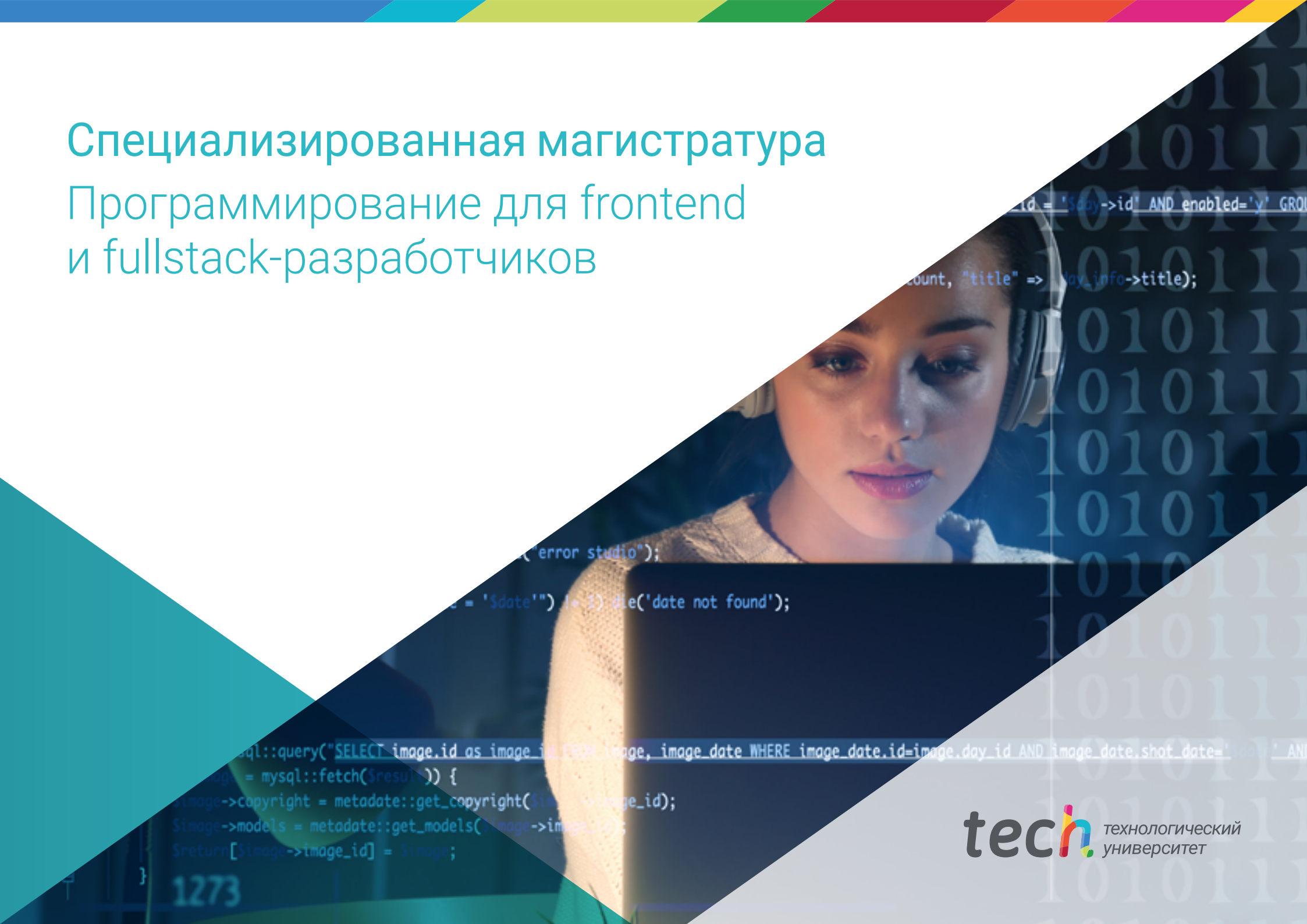
Специализированная магистратура

Программирование для frontend и fullstack-разработчиков



tech технологический университет

tech технологический университет

Специализированная магистратура

Программирование для frontend и fullstack-разработчиков

- Формат: **онлайн**
- Продолжительность: **12 месяцев**
- Учебное заведение: **TECH Технологический университет**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Веб-доступ: www.techtitute.com/ru/information-technology/professional-master-degree/master-full-front-end-programming-stack-developer

Оглавление

01

Презентация

Frontend-разработчики являются незаменимой частью любой веб-разработки. Их природа как связующего звена между графическим дизайном и *Back End*-программированием требует от них очень специфических и развитых знаний, которые, в свою очередь, необходимо постоянно обновлять, чтобы соответствовать все более требовательным задачам современности. Поскольку специальность предлагает множество возможностей на профессиональном уровне, TECH собрал лучшую команду преподавателей для разработки обширной, полной и методичной университетской программы. Опираясь на самые передовые инструменты и методологии в разработке веб-решений, эта программа дает необходимый импульс для того, чтобы сделать шаг к *fullstrack и frontend*- программированию.

Станьте неотъемлемой частью любой веб-разработки, возглавляя и координируя ИТ-команды благодаря отточенной методологии работы и лидерства этой Специализированной магистратуры"

Fullstack-разработка – это особенно привлекательный вариант для всех ИТ-специалистов, желающих значительно улучшить свою карьеру. Навыки, необходимые для успешной работы в этом секторе, обширны, что означает, что возможности процветания и даже руководства группами разработчиков многообразны.

Благодаря всестороннему подходу к разработке содержания этой программы студент сможет направить свою карьеру в сторону *frontend*-разработки, верстки страниц, специалиста по работе с клиентами или DevOps. Имея 360º видение всего процесса создания приложения/веб-страницы, специалист по информатике будет способен справиться с любым типом проекта, а также обеспечить разработку в соответствии с последними достижениями во всех процессах жизненного цикла программного обеспечения.

Уникальная академическая возможность получить доступ к знаниям, сочетающим новейшую компьютерную теорию с профессиональной передовой практикой, предоставляемая командой преподавателей самого высокого уровня. Опыт работы преподавателей во главе многочисленных проектов в области цифрового банкинга или телекоммуникаций делает дидактическое содержание более насыщенным, предоставляя большое количество реальных примеров и дополнительных материалов.

Большая гибкость этого обучения – еще одна из его самых отличительных характеристик. Здесь нет фиксированного расписания или очных занятий, студент сам решает, когда, где и как проходить всю учебную нагрузку. Все содержимое виртуального класса доступно для скачивания и может изучаться с любого устройства, имеющего подключение к Интернету.

Данная **Специализированная магистратура в области программирования для frontend и fullstack-разработчиков** содержит самую полную и современную образовательную программу на рынке. Основными особенностями обучения являются:

- Разработка практических кейсов, представленных специалистами в области программирования для frontend и fullstack-разработчиков
- Наглядное, схематичное и исключительно практичное содержание курса предоставляет научную и практическую информацию по тем дисциплинам, которые необходимы для осуществления профессиональной деятельности
- Практические упражнения для самооценки, контроля и улучшения успеваемости
- Особое внимание уделяется инновационным методологиям
- Теоретические занятия, вопросы эксперту и самостоятельные работы
- Учебные материалы курса доступны с любого стационарного или мобильного устройства с выходом в интернет

Зарегистрируйтесь сейчас и не упустите возможность, которая поднимет вас в высшую точку лидерства и развития самых амбициозных ИТ-проектов"

Вы достигнете продвинутого уровня экспертизы и сможете создавать любые необходимые веб-решения с учетом современных требований клиентского опыта и адаптированные к текущему рынку"

В преподавательский состав программы входят профессионалы в данной области, которые применяют в процессе обучения свой опыт работы, а также специалисты из ведущих сообществ и престижных университетов.

Мультимедийное содержание программы, разработанное с использованием новейших образовательных технологий, позволит специалисту проходить обучение с учетом контекста и ситуации, т.е. в симулированной среде, обеспечивающей иммерсивный учебный процесс, запрограммированный на обучение в реальных ситуациях.

Формат этой программы ориентирован на проблемное обучение, с помощью которого специалист должен попытаться разрешить различные ситуации из профессиональной практики, возникающие во время обучения. В этом поможет инновационная интерактивная видеосистема, созданная признанными экспертами.

Вы углубите свои знания о гибкой методологии и о том, как она может быть реализована в процессе разработки, повысите свои междисциплинарные навыки и компетенции.

У вас будет доступ к широкому спектру учебных материалов, от языка программирования Javascript до таких инструментов, как CSS, Angular и ReactJS.

02

Цели

Целью данной Специализированной магистратуры в области программирования для frontend и fullstack-разработчиков, с учетом многочисленных возможностей, которые предлагает разработка, является не что иное, как предоставление самых передовых знаний и методов в этой области. Таким образом, благодаря исключительно практическому подходу всего представленного материала, программисты могут начать разработку собственных проектов или ускорить свою профессиональную карьеру еще до получения диплома.

Вы достигнете своих самых амбициозных карьерных целей благодаря отличительному подходу этой программы, которая проведет вас через все этапы frontend и fullstack-разработки"

Общие цели

- Сформировать экспертные знания по ключевым аспектам программирования
- Стимулировать развитие алгоритмического мышления
- Предоставить необходимые инструменты и навыки для разработки
- Способствовать внедрению методологий Agile для реализации проектов
- Развить специализированные знания основ веб-технологий
- Способствовать использованию современных инструментов и методов *frontend*-разработки
- Разработка веб-дизайна для правильного макетирования
- Оценивать полученные знания

Конкретные цели

Модуль 1. *Fullstack*-разработка

- Развить продвинутые навыки программирования
- Укрепить использование систем контроля версий и платформ для размещения кода
- Содействовать использованию Agile-методологий
- Сформировать знание об основных понятиях и функционировании Интернета
- Повысить уровень владения командной строкой

Модуль 2. *Frontend*-разработка в программировании

- Определять и понимать правильный синтаксис HTML и CSS
- Изучить различные элементы HTML
- Определить подход к адаптивному проектированию
- Использовать форматирование презентаций для веб-страниц с помощью каскадных таблиц стилей
- Встроить в работу препроцессор CSS
- Выделить преимущества использования препроцессора
- Сформировать специализированные знания о системах проектирования
- Установить критерии для использования системы проектирования

Модуль 3. Язык JAVASCRIPT применительно к *fullstack-разработке*

- Выделить основные и сложные типы, предлагаемые JavaScript
- Анализировать различные способы программирования на языке и правильно использовать их в каждой ситуации
- Обновить знания до последних версий
- Получить представление о функциональном программировании
- Изучить асинхронное программирование и его особенности

Модуль 4. Макет веб-сайтов применительно к *fullstack-разработке*

- Проводить оценку веб-дизайна, чтобы знать, как применить его на временной основе
- Изучить основные правила CSS
- Представить различные методики CSS для получения *адаптивных* дизайнов
- Обосновать принципы каскадной разработки CSS
- Уметь идентифицировать технологии Bootstrap в любом веб-дизайне
- Анализировать принципы работы Bootstrap
- Разработать веб-макет с использованием Bootstrap
- Определить принципы разработки в проекте SaSS

Модуль 5. Инструменты JavaScript. Библиотека ReactJs

- Определить функциональные возможности React
- Настроить проект с помощью Create React App
- Проанализировать жизненный цикл компонентов в React
- Сформировать специальные знания о современных функциях React, таких как *Hooks* и *Context*
- Установить глобальные состояния с помощью *Context*
- Формировать и осуществлять рендеринг списков и создавать формы с помощью React
- Внедрять валидацию полей в формах
- Реализовать стилизацию компонентов и элементов
- Осуществлять отладку, тестировать и развертывать приложения React

Модуль 6. *Фреймворк* для JavaScript. Angular

- Расширить специальные знания об архитектуре *Framework*
- Углубить знания о методике Angular
- Проанализировать концепцию компонентов
- Осуществлять правильную организацию кода

Модуль 7. Программирование на языке NodeJS

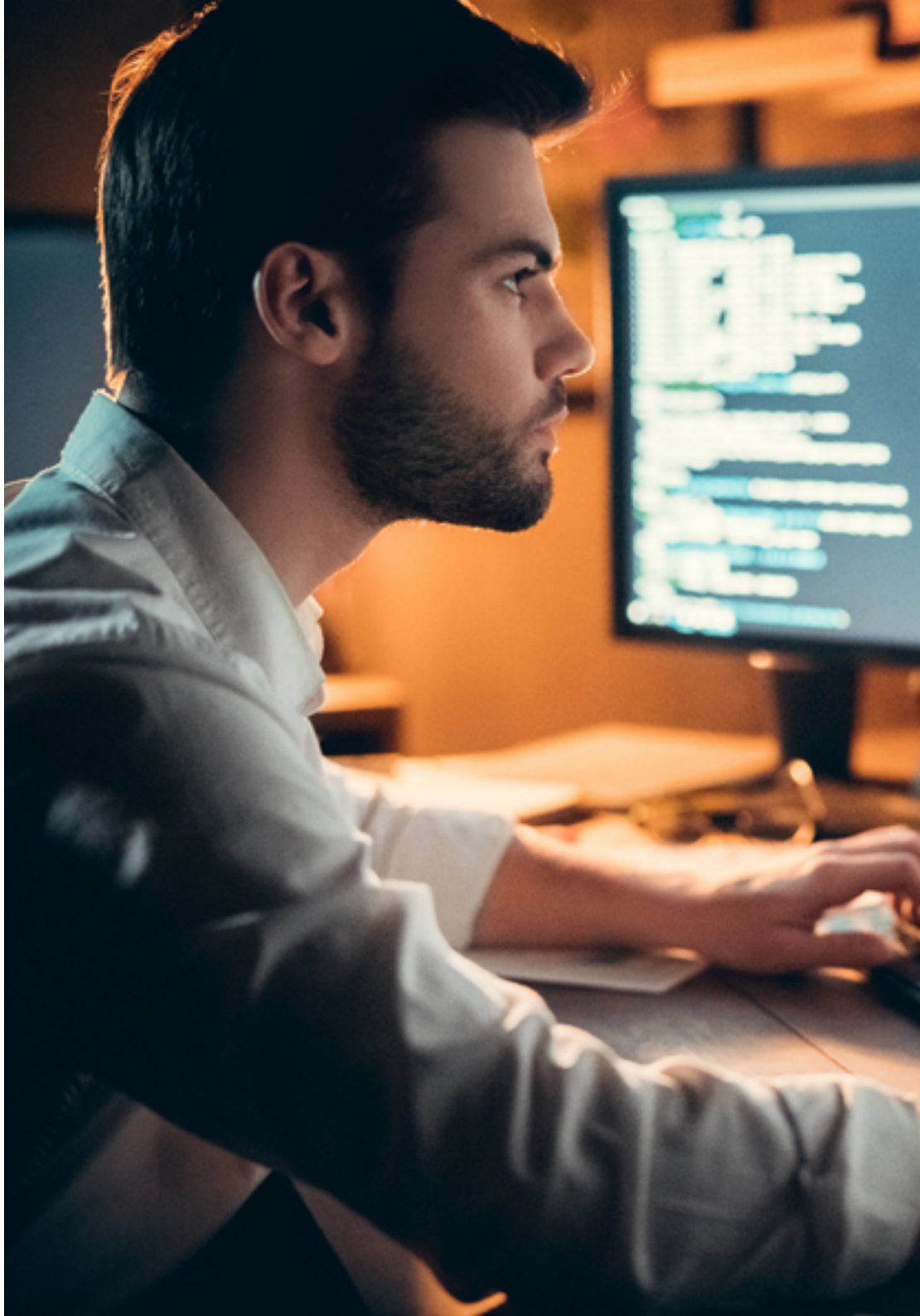
- Сформировать специальные знания о типах JavaScript и их операторах
- Проанализировать наилучшие способы программирования на этом языке
- Обновить знания до последних версий
- Изучить функциональное программирование
- Изучить асинхронное программирование и мотивацию
- Получить навык создания приложения с помощью NodeJSINDEX

Модуль 8. Базы данных для *fullstack*-разработчиков

- Определить, зачем использовать базу данных при разработке приложений
- Изучить типы доступных баз данных и их различия
- Разработать четкое представление о том, для чего используется каждый тип базы данных
- Проанализировать использование баз данных в современных парадигмах разработки

Модуль 9. UX CX. Клиентский опыт

- Анализировать важность пользователя в настоящее время и углублять культуру обратной связи
- Реализовывать омниканальные стратегии и стратегии персонализации на основе микровзаимодействий
- Изучить эволюцию веб-аналитики до поведенческой аналитики
- Определить, как искусственный интеллект вывел CX на новый уровень



- Выявить наиболее важные методы анализа веб-опыта, мобильности и доступности
- Представить методологию *Design Thinking* и процесс создания пользовательского опыта
- Представить конкретные инструменты прототипирования и *wireframing*, а также *frameworks* при *frontend*-разработке

Модуль 10. Непрерывная интеграция и развертывание приложений

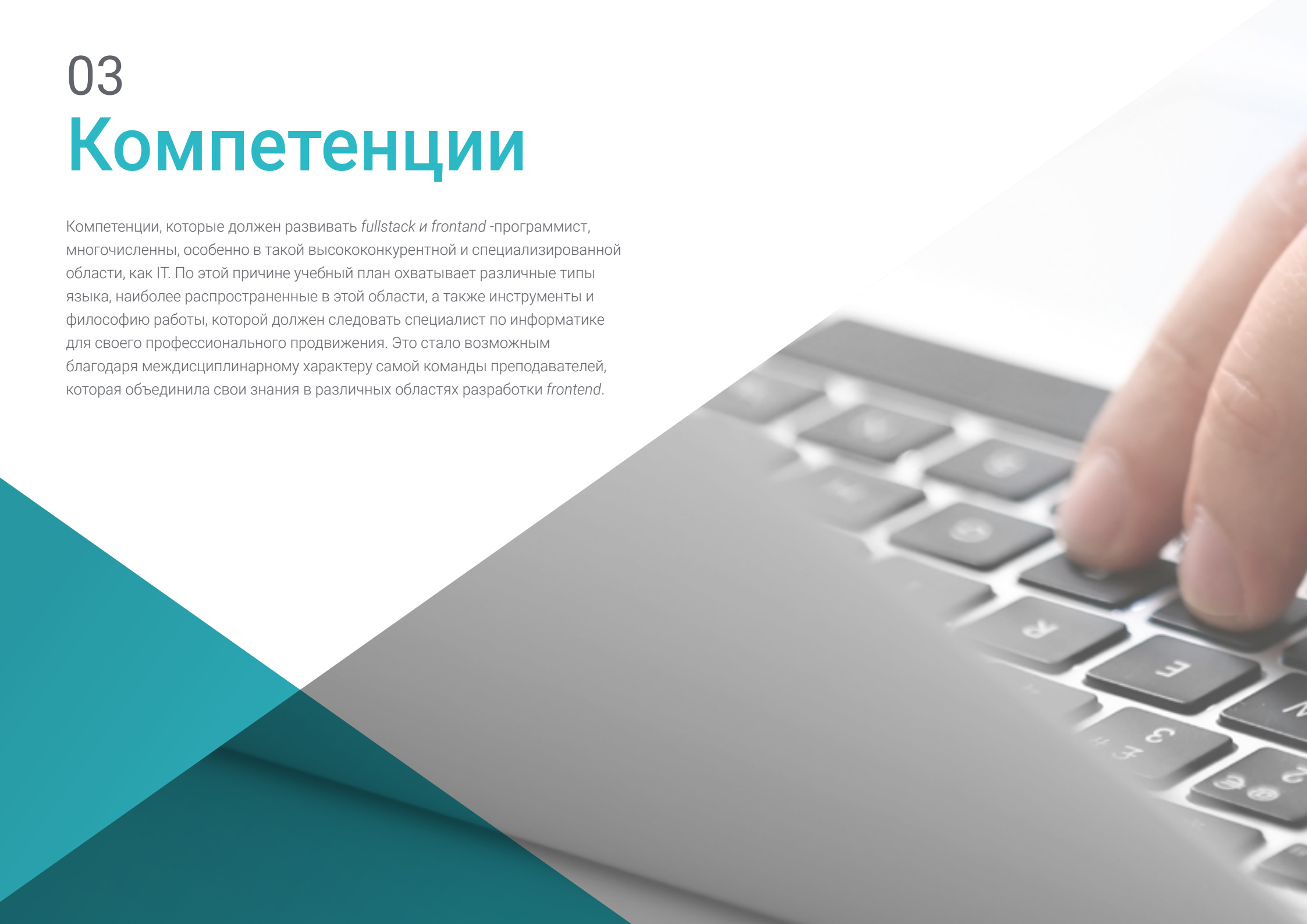
- Определить преимущества внедрения автоматизированной модели развертывания приложений
- Установить различия между непрерывной интеграцией, непрерывной доставкой и непрерывным развертыванием
- Определить основные особенности DevOps
- Произвести оценку некоторых ключевых инструментов для внедрения конвейеров CI/CD
- Определить ключевые факторы для разработки приложений, готовых к поддержке процессов CI/CD
- Изучить контейнерные технологии как фундаментальную основу практики CI/CD

Вы будете совершенствовать свои навыки и компетенции постепенно, в течение 10 модулей, разработанных на основе самых глубоких знаний и проверенного опыта всех преподавателей"

03

Компетенции

Компетенции, которые должен развивать *fullstack и frontand* -программист, многочисленны, особенно в такой высококонкурентной и специализированной области, как IT. По этой причине учебный план охватывает различные типы языка, наиболее распространенные в этой области, а также инструменты и философию работы, которой должен следовать специалист по информатике для своего профессионального продвижения. Это стало возможным благодаря междисциплинарному характеру самой команды преподавателей, которая объединила свои знания в различных областях разработки *frontend*.



Вы придадите своему резюме высококачественный импульс, включив в него множество навыков и компетенций, необходимых на самом высоком уровне разработки программного обеспечения"

Общие профессиональные навыки

- Правильно распознавать синтаксис языков HTML и CSS
- Разработать критерии передовой практики для веб-разработки
- Сформировать специализированные знания о языке JavaScript
- Уметь разрабатывать любые приложения с помощью JavaScript
- Проанализировать библиотеку Bootstrap
- Реализовывать проекты по планировке с SaSS (*Syntactically Awesome Stylesheets*)
- Определить синтаксис React и как программировать с его помощью
- Применять передовой опыт к языку
- Изучить процесс загрузки и доступа в каждом из ведущих типов баз данных в вашей области
- Оценить наиболее важные инструменты и методы анализа CX и *'стек технологий'*, обычно используемые компаниями

Профессиональные навыки

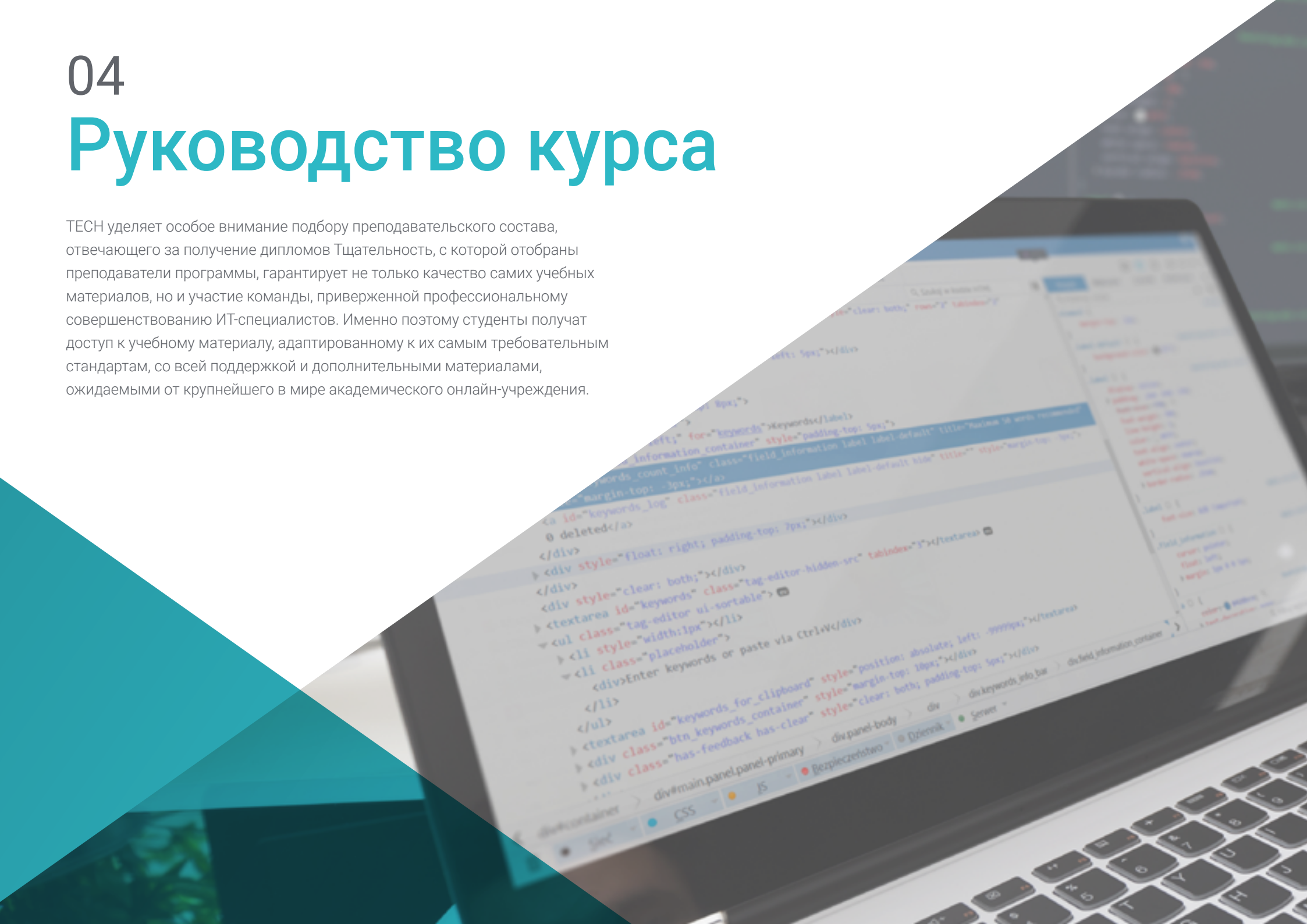
- Проанализировать различные структуры данных
- Изучить методы разработки и интерпретации алгоритмов
- Подготовить среду разработки
- Клонировать веб-сайт
- Создать сайт с помощью Bootstrap
- Компилировать код CSS с помощью SaSS
- Разработать собственный CSS *framework* на основе Bootstrap с помощью SaSS
- Разработать проект и воплотить его в жизнь
- Определить, как подключаться и загружать/извлекать данные из различных типов баз данных
- Определить практику, примеры использования, технологии и инструменты экосистемы CI/CD, необходимые для поддержки всего процесса

На протяжении всего курса вы будете развивать необходимые навыки для успешной frontend-разработки, повышая не только свои собственные знания, но и всеобъемлющие компетенции"

04

Руководство курса

TECH уделяет особое внимание подбору преподавательского состава, отвечающего за получение дипломов Тщательность, с которой отобраны преподаватели программы, гарантирует не только качество самих учебных материалов, но и участие команды, приверженной профессиональному совершенствованию ИТ-специалистов. Именно поэтому студенты получат доступ к учебному материалу, адаптированному к их самым требовательным стандартам, со всей поддержкой и дополнительными материалами, ожидаемыми от крупнейшего в мире академического онлайн-учреждения.



Вы сможете обсудить все возникающие вопросы непосредственно с преподавательским составом, получив образовательное наставничество, адаптированное к вашим потребностям"

Руководство

Г-н Олайа Бональ, Мартин

- Технический специалист по *блокчейну* в IBM SPGI
- Специалист технических продаж в области *блокчейна* IBM
- Директор по архитектуре. *Blocknitive*
- Технический специалист в области цифровой электроники
- *Архитектор блокчейна - Архитектор инфраструктуры* IT - Руководитель проектов IT. Области деятельности: Программное обеспечение, инфраструктура, телекоммуникации

Преподаватели

Г-н Кальсада Мартинес, Хесус

- Старший инженер-программист в компании Devo
- *Fullstack* -разработчик в Blocknitive
- Ответственный за *frontend* в Infinia
- *Fullstack*-разработчик в Resem
- Java-разработчик в Hitec
- Степень бакалавра в области программирования

Г-н Фриас Фаверо, Педро Луис

- CTO - Swearit Technologies
- COO - Key Identification
- Эксперт в области *блокчейна* и децентрализованных приложения, Университет Алкалы
- Fullstack-разработчик в Ironhack
- Степень бакалавра в области промышленной инженерии Университета Якамбу

Г-н Герреро Диас-Пинтадо, Артуро

- Технический инженер по предпродажной подготовке портфеля Watson *Customer Engagement (*решения для маркетинга *и клиентского опыта)* в Испании, Португалии, Греции и Израиле в IBM
- Инженер сетей НИОКР в Telefónica
- Консультант в области профессиональных знаний, сотрудничающий с передовыми организациями в Европе, Латинской Америке и на Ближнем Востоке
- Степень бакалавра в области инженерии в телекоммуникациях Университета Алькала и *Датского технического университета*
- Сотрудничество с известными университетами и центрами высшего образования по таким технологическим дисциплинам, как искусственный интеллект, *интернет вещей, облака, клиентский опыт и цифровая трансформация*

Г-н Пинтадо Сан Клаудио, Бруно

- Координатор разработок в IDavinci
- Java-разработчик в Национальной библиотеке Испании
- Разработчик поддержки и сетевой техник N1 в Sanitas
- Техник по системной поддержке в городском совете Алькобендас
- Техник связи N1 для ADIF в Телекоммуникационном центре Аточа
- Степень бакалавра в области технических средств телекоммуникаций Мадридского политехнического университета по специальности "Электронные системы"
- Степень бакалавра в области инженерии коммуникационной электроники Политехнического университета Мадрида

Г-н Рейес Олива, Луис

- Разработчик и архитектор облачных вычислений в IBM
- Технический клиентский менеджер по интегрированным счетам BBVA в IBM
- *Cloud Executive Selling* в IBM
- Архитектор облачных вычислений и DevOps в IBM
- Архитектор программного обеспечения для клиентов в компании Telefónica
- Архитектор технических решений для Rational
- *Менеджер по разработке программного обеспечения* в Borland
- Менеджер по разработке программного обеспечения и обеспечению качества в Altana Consulting
- Степень бакалавра в области компьютерной инженерии в Папском университете Саламанки, Мадрид

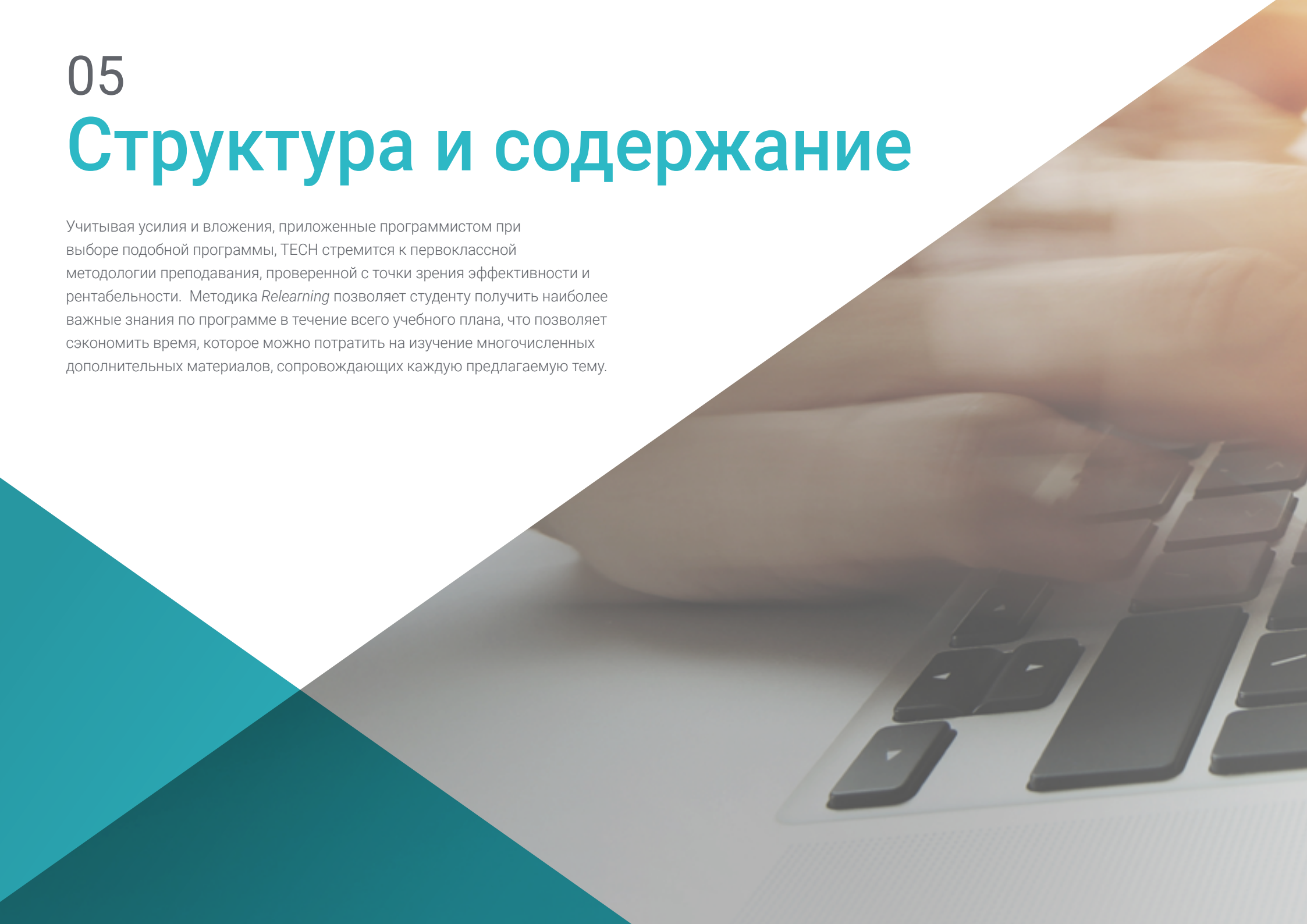
Г-н Гомес Родригес, Антонио

- Инженер по облачным решениям в компании Oracle
- Руководитель проектов в компании Sopra Group
- Руководитель проектов в компании Everis
- Руководитель проектов в государственной компании по управлению культурными программами. Министерства культуры Андалусии
- Аналитик информационных систем. Sopra Group
- Степень бакалавра в области высшей телекоммуникационной инженерии в Политехнического университета Каталонии
- Послевузовское профессиональное образование в области информационных технологий и систем в Каталонском технологическом институте
- Магистерская специализация в области *E-Business* в Бизнес-школе Ла-Салье

05

Структура и содержание

Учитывая усилия и вложения, приложенные программистом при выборе подобной программы, TECH стремится к первоклассной методологии преподавания, проверенной с точки зрения эффективности и рентабельности. Методика *Relearning* позволяет студенту получить наиболее важные знания по программе в течение всего учебного плана, что позволяет сэкономить время, которое можно потратить на изучение многочисленных дополнительных материалов, сопровождающих каждую предлагаемую тему.



Изучите различные модули и темы с помощью коротких интерактивных видео, подробные и мотивирующие видео, созданные самими преподавателями"

Модуль 1. *Fullstack*-разработка

1.1. *Fullstack-разработка* I. Программирование и языки
- 1.1.1. Программирование
- 1.1.2. Роли в программировании
- 1.1.3. Языки и *фреймворк*
- 1.1.4. Алгоритм
- 1.1.5. Характеристики алгоритма

1.2. *Fullstack*-разработка II. Типология
- 1.2.1. Переменные и константы
- 1.2.2. Типы
- 1.2.3. Операции
- 1.2.4. Заявления
- 1.2.5. Петли
- 1.2.6. Функции и объекты

1.3. Структура данных при разработке
- 1.3.1. Виды линейных структур
- 1.3.2. Виды функциональных структур
- 1.3.3. Типы древовидных структур

1.4. Разработка и интерпретация алгоритмов
- 1.4.1. Параллелизм при разработке. Разделяй и властвуй
- 1.4.2. Жадные алгоритмы
- 1.4.3. Динамическое программирование

1.5. Среда и инструменты для развития в области *fullstack-разработок*
- 1.5.1. Подготовка среды для Mac OS
- 1.5.2. Подготовка среды для Linux
- 1.5.3. Подготовка среды для Windows

1.6. Командная строка Типология и функционирование
- 1.6.1. Терминал
- 1.6.2. Эмуляторы
- 1.6.3. Интерпретатор команд
- 1.6.4. Первые команды
- 1.6.5. Навигация
- 1.6.6. Управление файлами и папками с помощью интерфейса командной строки
- 1.6.7. *Secure Shell* SSH
- 1.6.8. Продвинутые команды

1.7. Git. Хранилище программного обеспечения
- 1.7.1. Репозиторий программного обеспечения Git
- 1.7.2. Использование Git
- 1.7.3. Репозитории программного обеспечения
- 1.7.4. Ветви
- 1.7.5. Рабочий цикл
- 1.7.6. Команды

1.8. Услуга хостинга версий кода
- 1.8.1. Услуга хостинга версий кода
- 1.8.2. Поставщики
- 1.8.3. Репозиторий

1.9. Интернет
- 1.9.1. Интернет
- 1.9.2. Протоколы, используемые в WWW
- 1.9.3. Протокол HTTP

1.10. Методологии *fullstack*-разработок
- 1.10.1. *Scrum*
- 1.10.2. XP
- 1.10.3. *Дизайн-спринт*

Модуль 2. *Frontend* в программировании

2.1. Язык HTML
- 2.1.1. Документ HTML
- 2.1.2. Элемент *Head*
- 2.1.3. Элемент *Body*
- 2.1.4. Текст
- 2.1.5. Гиперссылка
- 2.1.6. Изображения
- 2.1.7. *Primer Site*

2.2. Язык HTML. *Лейаут*
- 2.2.1. Язык HTML. Элементы
- 2.2.2. Традиционный *лейаут*
- 2.2.3. Семантический *лейаут*

2.3. Каскадные таблицы стилей CSS (*Cascading Style Sheets*)
- 2.3.1. Включение CSS в HTML-документ
- 2.3.2. Комментарии
- 2.3.3. Селекторы
- 2.3.4. Расширенные селекторы

2.4. Свойства CSS (*Cascading Style Sheets*)
- 2.4.1. Цвет
- 2.4.2. Текст
- 2.4.3. Псевдоклассы
- 2.4.4. Переходы
- 2.4.5. Анимация
- 2.4.6. Анимация элементов
- 2.4.7. Продвинутая анимация

2.5. Бокс-модель
- 2.5.1. Высота и ширина
- 2.5.2. Края
- 2.5.3. Заливка

2.6. Расположение
- 2.6.1. Статическое расположение
- 2.6.2. Относительное расположение
- 2.6.3. Абсолютное расположение
- 2.6.4. Фиксированное расположение
- 2.6.5. *Тип данных floats*

2.7. Адаптационный дизайн
- 2.7.1. *Viewport*
- 2.7.2. *Медиа-запросы*
- 2.7.3. Единицы CSS
- 2.7.4. *Изображения*
- 2.7.5. *Фреймворк*

2.8. Современный *лейаут*
- 2.8.1. *Flex*
- 2.8.2. *Grid*
- 2.8.3. *Flex* vs. *Grid*

2.9. Препроцессор
- 2.9.1. Sass
- 2.9.2. Переменные
- 2.9.3. Миксины
- 2.9.4. Петли
- 2.9.5. Функции

2.10. Система дизайна
- 2.10.1. Bootstrap
- 2.10.2. Bootstrap Grid
- 2.10.3. *Header* и *Footer* нашего сайта
- 2.10.4. Формуляры
- 2.10.5. *Карты*
- 2.10.6. Модальные окна

Модуль 3. Язык JAVASCRIPT применительно к *fullstack-разработке*

3.1. Примитивные типы данных и операторов
- 3.1.1. Язык JavaScript
- 3.1.2. Числа и их операторы
- 3.1.3. Текстовые строки и их операторы
- 3.1.4. Булевы значения
- 3.1.5. Преобразование между типами

3.2. Регуляторы потока и структура
- 3.2.1. Выражения и предложения
- 3.2.2. Переменные и константы
- 3.2.3. Условие *If*
- 3.2.4. Условие *For, While*

3.3. Функции
- 3.3.1. Функции
- 3.3.2. Параметры
- 3.3.3. Функции как параметры
- 3.3.4. Область применения переменных
- 3.3.5. Вложенные области *Scopes*
- 3.3.6. *Поднятие или hoisting*
- 3.3.7. *Закрытия*
- 3.3.8. Рекурсия

3.4. Структуры данных: объекты
- 3.4.1. Тип *Object*
- 3.4.2. Создание объектов
- 3.4.3. Доступ к значениям объекта
- 3.4.4. Добавление или удаление свойств
- 3.4.5. Вложенные объекты
- 3.4.6. *Деструктуризация (destructuring)* объектов
- 3.4.7. Методы типа *Object*
- 3.4.8. *Spread Operator*
- 3.4.9. Immutable объекты

3.5. Структура данных: *Массив*
- 3.5.1. Структура данных. *Массив*
- 3.5.2. *Массив* Типология
- 3.5.3. Вложенные *массивы*
- 3.5.4. Методы *массивов*

3.6. Объектно-ориентированное программирование (ООП): *Прототип* и классы
- 3.6.1. ООП: Объектно-ориентированное программирование
- 3.6.2. Прототипы
- 3.6.3. Классы
- 3.6.4. Частные данные
- 3.6.5. Подклассы
- 3.6.6. *Call* и *Apply*

3.7. Типы JavaScript
- 3.7.1. *Set*
- 3.7.2. *WeakSet*
- 3.7.3. *Map*
- 3.7.4. *WeakMap*
- 3.7.5. Регулярные выражения

3.8. Утилиты JavaScript
- 3.8.1. *Date*
- 3.8.2. *Math*
- 3.8.3. *Symbol*
- 3.8.4. JSON

3.9. JavaScript в *браузере*
- 3.9.1. Включение JavaScript в веб-сайт
- 3.9.2. DOM
- 3.9.3. События
- 3.9.4. *Хранение данных* в браузере

3.10. Асинхронное программирование
- 3.10.1. Асинхронное программирование
- 3.10.2. *Event Loop*
- 3.10.3. *Механизм обратного вызова(callbacks)*
- 3.10.4. *Promises*
- 3.10.5. Async/Await

Модуль 4. Макет веб-сайтов применительно к *fullstack-разработке*

4.1. CSS и макет
- 4.1.1. Макет с использованием таблиц
- 4.1.2. Динамичный дизайн
- 4.1.3. Эра *Отзывчивого дизайна*
- 4.1.4. *Mobile First* vs. *Desktop First*

4.2. CSS и правила веб-дизайна
- 4.2.1. Селекторы
- 4.2.2. Псевдоклассы
- 4.2.3. Псевдоэлементы

4.3. Макетирование CSS
- 4.3.1. Правила *бокс-модели*
- 4.3.2. Типографии
- 4.3.3. Цвета
- 4.3.4. Изображения
- 4.3.5. Фоны
- 4.3.6. Таблицы
- 4.3.7. Формуляры
- 4.3.8. Показ и скрытие элементов
- 4.3.9. Варианты CSS

4.4. *Отзывчивый* дизайн и динамичный дизайн
- 4.4.1. Плавающие элементы
- 4.4.2. *Grid* CSS
- 4.4.3. *Медиа-запросы*
- 4.4.4. *Flexbox*

4.5. Каскад CSS
- 4.5.1. Приоритет правил CSS
- 4.5.2. Правила перезаписи
- 4.5.3. Классы vs. Идентификаторы

4.6. SaSS
- 4.6.1. ПО как услуга (SaSS)
- 4.6.2. Установка SaSS
- 4.6.3. Запуск и компиляция SaSS
- 4.6.4. Структура директории SaSS

4.7. Использование SaSS
- 4.7.1. Переменные в SaSS
- 4.7.2. Модулирование нашего проекта
- 4.7.3. Синтаксис SaSS

4.8. Логика SaSS
- 4.8.1. Миксины
- 4.8.2. Карты
- 4.8.3. Функции и структуры управления

4.9. Макетирование в Bootstrap
- 4.9.1. Bootstrap
- 4.9.2. *Лейауты* в Bootstrap
- 4.9.3. Формуляры
- 4.9.4. *Бокс-модели* в Bootstrap
- 4.9.5. Цвета и шрифты
- 4.9.6. Линки и кнопки
- 4.9.7. Показывать и скрывать элементы в Bootstrap
- 4.9.8. *Flexbox* в Bootstrap
- 4.9.9. Компоненты

4.10. *Theming* Bootstrap
- 4.10.1. Переписывая Bootstrap с SaSS (*Software as a Service*)
- 4.10.2. Структура файлов
- 4.10.3. Формирование собственных *Framework* CSS (*Cascading Style Sheets*)

Модуль 5. Инструменты JavaScript. Библиотека ReactJs

5.1. Инструменты Javascript ReactJS
- 5.1.1. Инструмент ReactJS
- 5.1.2. Create React App
- 5.1.3. JavaScript *Syntax Extension*

5.2. Компоненты ReactJS
- 5.2.1. Компоненты
- 5.2.2. *Предметы реквизита*
- 5.2.3. Рендеринг

5.3. События в библиотеке ReactJS
- 5.3.1. Управление событиями
- 5.3.2. Управление событиями в режиме онлайн
- 5.3.3. События в библиотеке ReactJS

5.4. Конфигурация *Hooks* в ReactJS
- 5.4.1. Состояния компонента
- 5.4.2. *Hook* состояния
- 5.4.3. *Hook* эффекта
- 5.4.4. *Custom Hooks*
- 5.4.5. Другие *Hooks*

5.5. *Component Context* в ReactJS
- 5.5.1. *Component Context* в ReactJS
- 5.5.2. Использование *Context*
- 5.5.3. Структура *Context*
- 5.5.4. *React. Create Context*
- 5.5.5. *Context. Provider*
- 5.5.6. *Class. Context Type*
- 5.5.7. *Context. Consumer*
- 5.5.8. *Context.displayName*
- 5.5.9. Практическое применение *Context*

5.6. Маршрутизация в ReactJs
- 5.6.1. *Router*
- 5.6.2. *React Router*
- 5.6.3. Установка
- 5.6.4. Базовая маршрутизация
- 5.6.5. Динамичная маршрутизация
- 5.6.6. Первичные компоненты
- 5.6.7. *React Router Hooks*

5.7. Использование списков и форм в ReactJS
- 5.7.1. Списки и циклы
- 5.7.2. Формы и валидация
- 5.7.3. *React Hook Forms*

5.8. Использование стилей ReactJS
- 5.8.1. Традиционный стиль
- 5.8.2. Онлайн-стили
- 5.8.3. Добавление библиотеки систем проектирования

5.9. Реализация проб в Javascript.Инструменты
- 5.9.1. *Тестирование*
- 5.9.2. Jest JavaScript *Testing Framework*
- 5.9.3. *Visual Testing and Documentation*

5.10. Развертка кода с помощью ReactJS
- 5.10.1. *Хостинг*
- 5.10.2. Поставщики
- 5.10.3. Подготовка проекта
- 5.10.4. Развертка в Heroku

Модуль 6. *Фреймворк* для JavaScript. Angular

6.1. *Фреймворк* Angular и его архитектура
- 6.1.1. Angular CLI
- 6.1.2. Архитектура
- 6.1.3. *Workspace* и структура
- 6.1.4. Окружающая среда

6.2. Компоненты *фреймворка* Angular
- 6.2.1. Жизненный цикл
- 6.2.2. Инкапсуляция
- 6.2.3. Взаимодействие между компонентами
- 6.2.4. Проекция контента

6.3. Шаблоны *фреймворка* Angular
- 6.3.1. Интерполяция текста
- 6.3.2. Заявления
- 6.3.3. *Property Binding*
- 6.3.4. *Class, Style* и *Attribute Binding*
- 6.3.5. *Event Binding* и *Two-Way Binding*
- 6.3.6. *Pipe Framework*

6.4. Директивы *фреймворка* Angular
- 6.4.1. Директивы в Angular
- 6.4.2. Директивы атрибутов
- 6.4.3. Структурные директивы

6.5. Сервисы и инъекция зависимостей
- 6.5.1. Услуги
- 6.5.2. Внедрение зависимостей
- 6.5.3. *Service Providers*

6.6. *Маршрутизация* и навигация
- 6.6.1. Приложение с *маршрутизацией*
- 6.6.2. Базовая маршрутизация
- 6.6.3. Вложенные маршруты
- 6.6.4. Параметры
- 6.6.5. Доступ и авторизация
- 6.6.6. *Lazy Loading* модулей

6.7. RxJS
- 6.7.1. Observables
- 6.7.2. *Observers*
- 6.7.3. Подписки
- 6.7.4. Операции

6.8. Формы и HTTP
- 6.8.1. Реактивные формы
- 6.8.2. Валидация полей
- 6.8.3. Динамичные формы
- 6.8.4. Петиции
- 6.8.5. *Interceptors*
- 6.8.6. Безопасность

6.9. Анимация
- 6.9.1. Переходы и *триггеры*
- 6.9.2. Переходы по маршруту
- 6.9.3. Различия между переходами

6.10. *Testing* в *фреймворке* Angular
- 6.10.1. Тестирование услуг
- 6.10.2. Тестирование компонентов
- 6.10.3. Тестирование директив и *pipes*

Модуль 7. Программирование на языке NodeJS

7.1. NodeJS и его архитектура
- 7.1.1. NPM и управление пакетами
- 7.1.2. Реализация программы
- 7.1.3. Модули
- 7.1.4. Создание модуля
- 7.1.5. *Event Loop* или цикл событий

7.2. Servidor Backend, HTTP, *Express* и *Sockets*
- 7.2.1. Модуль HTTP
- 7.2.2. Express
- 7.2.3. Socket.io

7.3. База данных и кэш
- 7.3.1. MongoDB
- 7.3.2. Mongoose
- 7.3.3. SQL
- 7.3.4. Sequelize
- 7.3.5. Redis

7.4. Файловая система и Os
- 7.4.1. *File System Module*
- 7.4.2. Os *Module*
- 7.4.3. *Cluster Module*

7.5. События, *Buffers* и *Streams*
- 7.5.1. События
- 7.5.2. *Буфферы*
- 7.5.3. *Streams*

7.6. *Тестирование*
- 7.6.1. Jest
- 7.6.2. Mocha
- 7.6.3. TDD - *Cucumber*

7.7. Архитектура и передовой опыт
- 7.7.1. DRY
- 7.7.2. SOLID
- 7.7.3. CRUD
- 7.7.4. MVC
- 7.7.5. Монолиты
- 7.7.6. Микроуслуги
- 7.7.7. Гексагональные архитектуры

7.8. Typescript
- 7.8.1. Типы, интерфейсы и классы
- 7.8.2. Функции и модули
- 7.8.3. Дженерики
- 7.8.4. *Namespaces*
- 7.8.5. Декораторы

7.9. API REST
- 7.9.1. Get
- 7.9.2. Post
- 7.9.3. Put
- 7.9.4. Delete
- 7.9.5. *Swagger*
- 7.9.6. Построение API REST с Express

7.10. Создание и контейнеризация приложения с помощью NestJS
- 7.10.1. Nest CLI
- 7.10.2. Docker
- 7.10.3. Создание приложения

Модуль 8. Базы данных для *fullstack*-разработчиков

8.1. Базы данных для *fullstack*-разработчиков
- 8.1.1. База данных в рамках разработки приложений
- 8.1.2. Способности баз данных
- 8.1.3. SQL (*Structured Query Language*)

8.2. Выборы баз данных
- 8.2.1. Приложение или услуга к рассмотрению
- 8.2.2. Категории баз данных
- 8.2.3. Панорамы баз данных

8.3. Разработка с использованием MySQL
- 8.3.1. Разработка с использованием MySQL
- 8.3.2. Развертывание реляционной модели с помощью MySQL
- 8.3.3. Подключение к MySQL

8.4. Разработка с использованием Oracle Database
- 8.4.1. Разработка с использованием Oracle DB
- 8.4.2. Развертывание модели
- 8.4.3. Подключение к Oracle Database

8.5. Разработка с использованием Oracle SQL Server
- 8.5.1. Oracle SQL Server
- 8.5.2. Развертывание модели
- 8.5.3. Подключение к SQL Server

8.6. Разработка с NoSQL
- 8.6.1. Сравнение с базами данных SQL
- 8.6.2. Создание базы данных в MongoDB
- 8.6.3. Подключение к MongoDB

8.7. Развитие с помощью сетей
- 8.7.1. Развитие с помощью сетей
- 8.7.2. Создание базы данных с помощью Neo4j
- 8.7.3. Подключение с помощью Neo4j

8.8. Разработка баз данных на основе пар «ключ-значение» (k-v)
- 8.8.1. Разработка баз данных на основе пар k-v
- 8.8.2. Создание базы данных с помощью Redis
- 8.8.3. Подключение к Redis

8.9. Базы данных с другими типами данных
- 8.9.1. *Elastic Search*
- 8.9.2. *Inmemory Database*
- 8.9.3. Разработка с использованием пространственных данных

8.10. Базы данных. Расширенные аспекты
- 8.10.1. Базы данных в нативных облачных разработках
- 8.10.2. Базы данных в архитектуре микросервисов
- 8.10.3. CI/CD и базы данных

Модуль 9. UX CX. Клиентский опыт

9.1. *Пользовательский опыт*
- 9.1.1. *Клиентский опыт* (CX)
- 9.1.2. Новые потребности потребителей
- 9.1.3. *Обратная связь* в *клиентском опыте*

9.2. Инновационные технологии
- 9.2.1. Думающие машины
- 9.2.2. Новые способы обмена информацией
- 9.2.3. Измерение того, что невозможно измерить

9.3. Каналы взаимодействия с пользователями
- 9.3.1. Клиентская аналитика
- 9.3.2. Персонализация
- 9.3.3. Многосторонние каналы взаимодействия с пользователями

9.4. Пользовательская аналитика
- 9.4.1. Структура веб-сайта
- 9.4.2. Пользовательская аналитика
- 9.4.3. Расширенная пользовательская аналитика

9.5. Nielsen и его влияние на CX
- 9.5.1. Nielsen и его влияние на CX
- 9.5.2. Техники *User Testing*

9.6. Инструменты в *Customer Experience*
- 9.6.1. Расширенные инструменты
- 9.6.2. Мобильность
- 9.6.3. Доступность

9.7. Новые методологии
- 9.7.1. Задача пользователя
- 9.7.2. UX-процесс
- 9.7.3. Изучение пользователей

9.8. Передача информации о дизайне
- 9.8.1. *Вайрфрейминг*
- 9.8.2. Инструменты коммуникации при дизайне
- 9.8.3. Расширенные инструменты коммуникации при дизайне

9.9. Дизайн анализа ковариации (UI)
- 9.9.1. Дизайн анализа ковариации (UI)
- 9.9.2. Веб- и мобильные интерфейсы
- 9.9.3. Веб- и мобильные компоненты

9.10. Разработка CX
- 9.10.1. Разработка CX
- 9.10.2. Разработка новых впечатлений
- 9.10.3. Интерфейсы

Модуль 10. Непрерывная интеграция и развертывание приложений

10.1. Непрерывная интеграция и непрерывное развертывание: CI/CD
- 10.1.1. Использование непрерывной интеграции и непрерывного развертывания (CI/CD)
- 10.1.2. Различия между непрерывной интеграцией и непрерывным развертыванием (CI/CD)
- 10.1.3. Непрерывная интеграция и непрерывное развертывание. Преимущества CI/CD

10.2. Новые парадигмы разработки
- 10.2.1. От Waterfall к DevOps
- 10.2.2. Руководство по стилю: 12 факторов
- 10.2.3. Cloud Native, микроуслуги и *Serverless*

10.3. DevOps, за пределами CI/CD
- 10.3.1. DevOps
- 10.3.2. DevOps. *Continuous Everything*
- 10.3.3. DevOps vs SRE

10.4. Контейнерные технологии I - Docker
- 10.4.1. Контейнеры. Взнос
- 10.4.2. Docker. Архитектура
- 10.4.3. Процесс развертывания с помощью Docker

10.5. Контейнерные технологии II - Kubernetes
- 10.5.1. Оркестровка
- 10.5.2. Kubernetes
- 10.5.3. Экосистема Kubernetes

10.6. Конфигурирование инфраструктуры с помощью GitOps
- 10.6.1. Неизменяемая инфраструктура
- 10.6.2. GitOps
- 10.6.3. Инструменты GitOps

10.7. Конвейеры и автоматизация. Примеры использования CI/CD
- 10.7.1. Непрерывная интеграция
- 10.7.2. Развертывание и непрерывная доставка
- 10.7.3. Автоматическая валидация
- 10.7.4. Передовая практика в области CI/CD

10.8. CI/CD с Jenkins. Примеры
- 10.8.1. CI/CD с Jenkins
- 10.8.2. Конвейеры Jenkins
- 10.8.3. Передовые методы работы с Jenkins

10.9. Экосистема CI/CD
- 10.9.1. Организация экосистемы
- 10.9.2. Расширенные инструменты
- 10.9.3. Dagger. Будущее

10.10. Заключительные фазы CI/CD-ориентированного программного цикла
- 10.10.1. Применение IA в процессе CI/CD
- 10.10.2. DevSecOps
- 10.10.3. *Chaos Engineering*

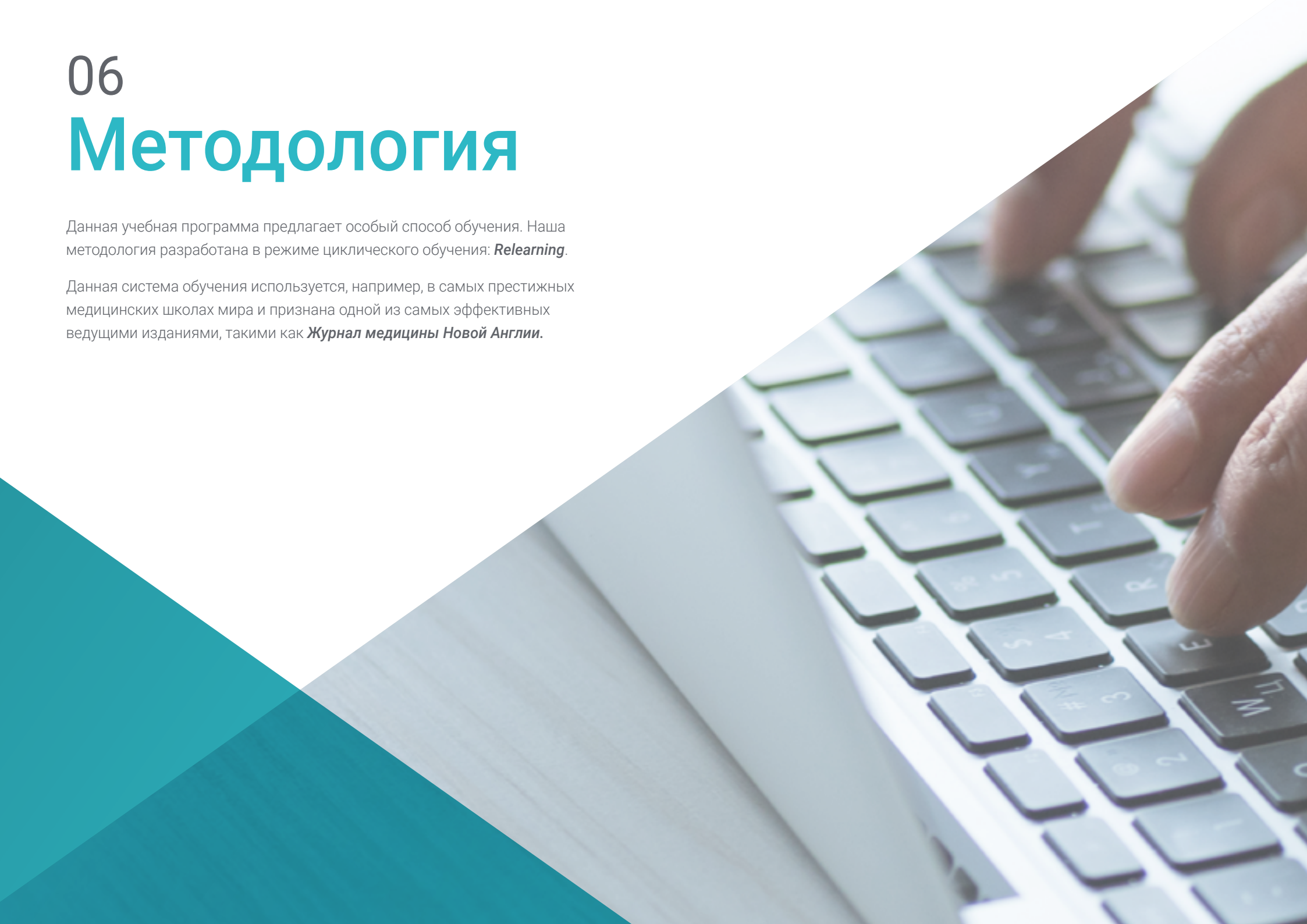
В вашем распоряжении будут самые современные образовательные ресурсы, свободный доступ к виртуальному классу 24 часа в сутки"

06

Методология

Данная учебная программа предлагает особый способ обучения. Наша методология разработана в режиме циклического обучения: ***Relearning***.

Данная система обучения используется, например, в самых престижных медицинских школах мира и признана одной из самых эффективных ведущими изданиями, такими как ***Журнал медицины Новой Англии.***



Откройте для себя методику Relearning, которая отвергает традиционное линейное обучение, чтобы показать вам циклические системы обучения: способ, который доказал свою огромную эффективность, особенно в предметах, требующих запоминания"

Исследование кейсов для контекстуализации всего содержания

Наша программа предлагает революционный метод развития навыков и знаний. Наша цель - укрепить компетенции в условиях меняющейся среды, конкуренции и высоких требований.

С TECH вы сможете познакомиться со способом обучения, который опровергает основы традиционных методов образования в университетах по всему миру”

Вы получите доступ к системе обучения, основанной на повторении, с естественным и прогрессивным обучением по всему учебному плану.

В ходе совместной деятельности и рассмотрения реальных кейсов студент научится разрешать сложные ситуации в реальной бизнес-среде.

Инновационный и отличный от других метод обучения

Эта программа TECH - интенсивная программа обучения, созданная с нуля, которая предлагает самые сложные задачи и решения в этой области на международном уровне. Благодаря этой методологии ускоряется личностный и профессиональный рост, делая решающий шаг на пути к успеху. Метод кейсов, составляющий основу данного содержания, обеспечивает следование самым современным экономическим, социальным и профессиональным реалиям.

> *Наша программа готовит вас к решению новых задач в условиях неопределенности и достижению успеха в карьере"*

Кейс-метод является наиболее широко используемой системой обучения лучшими преподавателями в мире. Разработанный в 1912 году для того, чтобы студенты-юристы могли изучать право не только на основе теоретического содержания, метод кейсов заключается в том, что им представляются реальные сложные ситуации для принятия обоснованных решений и ценностных суждений о том, как их разрешить. В 1924 году он был установлен в качестве стандартного метода обучения в Гарвардском университете.

Что должен делать профессионал в определенной ситуации? Именно с этим вопросом мы сталкиваемся при использовании кейс-метода - метода обучения, ориентированного на действие. На протяжении всей курса студенты будут сталкиваться с многочисленными реальными случаями из жизни. Им придется интегрировать все свои знания, исследовать, аргументировать и защищать свои идеи и решения.

Методология *Relearning*

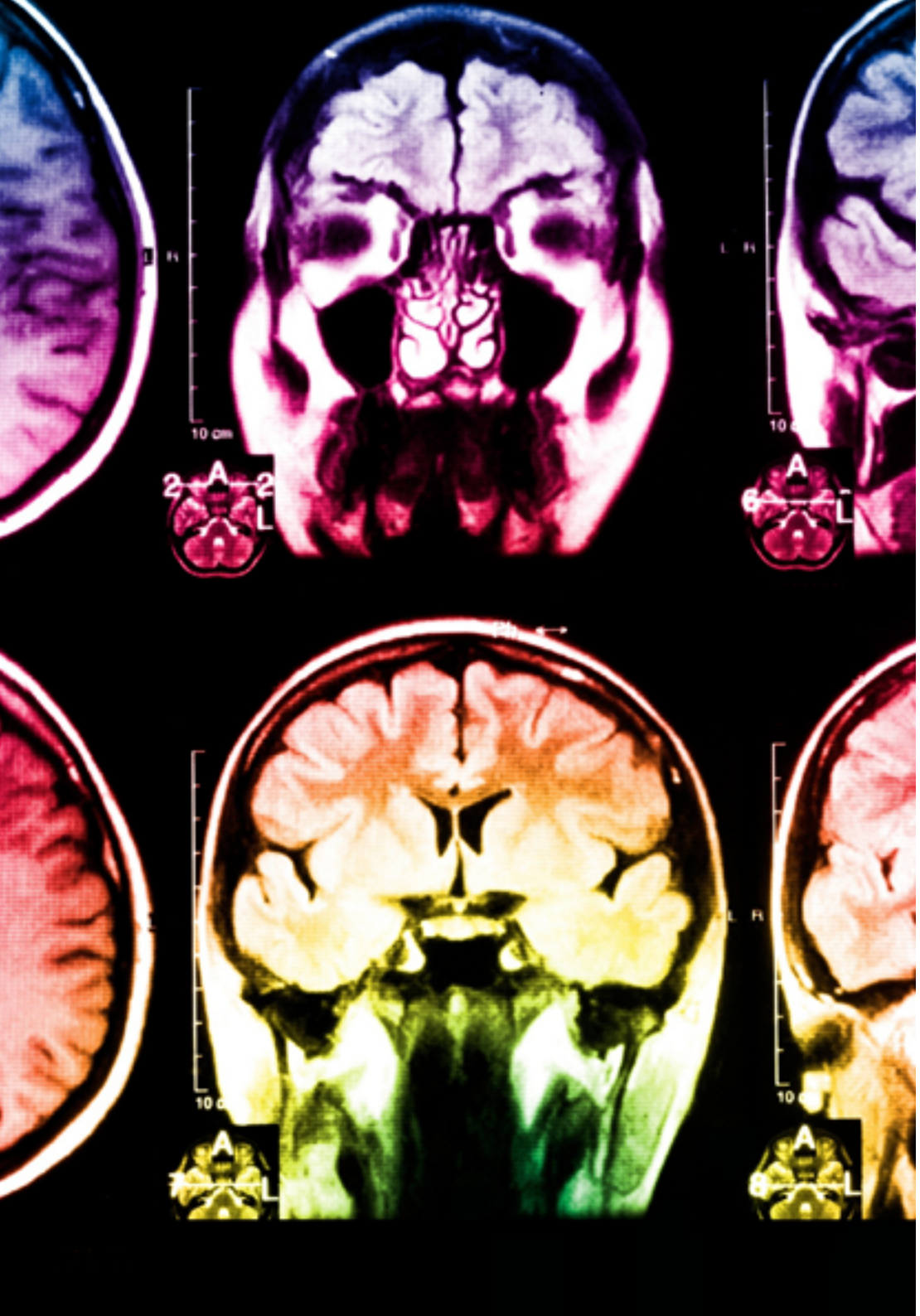
TECH эффективно объединяет метод кейсов с системой 100% онлайн-обучения, основанной на повторении, которая сочетает различные дидактические элементы в каждом уроке.

Мы улучшаем метод кейсов с помощью лучшего метода 100% онлайн-обучения: *Relearning*.

В 2019 году мы достигли лучших результатов обучения среди всех онлайн-университетов в мире.

В TECH вы будете учиться по передовой методике, разработанной для подготовки руководителей будущего. Этот метод, играющий ведущую роль в мировой педагогике, называется *Relearning*.

Наш университет - единственный вуз, имеющий лицензию на использование этого успешного метода. В 2019 году нам удалось повысить общий уровень удовлетворенности наших студентов (качество преподавания, качество материалов, структура курса, цели...) по отношению к показателям лучшего онлайн-университета.



В нашей программе обучение не является линейным процессом, а происходит по спирали (мы учимся, разучиваемся, забываем и заново учимся). Поэтому мы дополняем каждый из этих элементов по концентрическому принципу. Благодаря этой методике более 650 000 выпускников университетов добились беспрецедентного успеха в таких разных областях, как биохимия, генетика, хирургия, международное право, управленческие навыки, спортивная наука, философия, право, инженерное дело, журналистика, история, финансовые рынки и инструменты. Наша методология преподавания разработана в среде с высокими требованиями к уровню подготовки, с университетским контингентом студентов с высоким социально-экономическим уровнем и средним возрастом 43,5 года.

Методика Relearning позволит вам учиться с меньшими усилиями и большей эффективностью, все больше вовлекая вас в процесс обучения, развивая критическое мышление, отстаивая аргументы и противопоставляя мнения, что непосредственно приведет к успеху.

Согласно последним научным данным в области нейронауки, мы не только знаем, как организовать информацию, идеи, образы и воспоминания, но и знаем, что место и контекст, в котором мы что-то узнали, имеют фундаментальное значение для нашей способности запомнить это и сохранить в гиппокампе, чтобы удержать в долгосрочной памяти.

Таким образом, в рамках так называемого нейрокогнитивного контекстно-зависимого электронного обучения, различные элементы нашей программы связаны с контекстом, в котором участник развивает свою профессиональную практику.

В рамках этой программы вы получаете доступ к лучшим учебным материалам, подготовленным специально для вас:

Учебный материал

Все дидактические материалы создаются преподавателями специально для студентов этого курса, чтобы они были действительно четко сформулированными и полезными.

Затем вся информация переводится в аудиовизуальный формат, создавая дистанционный рабочий метод TECH. Все это осуществляется с применением новейших технологий, обеспечивающих высокое качество каждого из представленных материалов.

Мастер-классы

Существуют научные данные о пользе экспертного наблюдения третьей стороны.

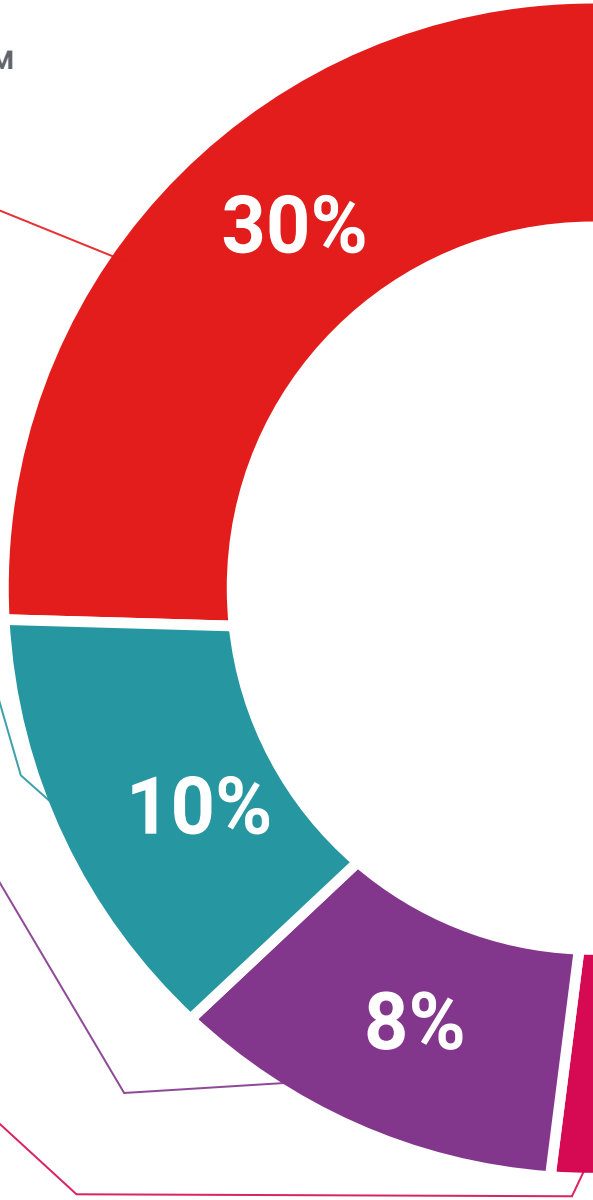
Так называемый метод обучения у эксперта укрепляет знания и память, а также формирует уверенность в наших будущих сложных решениях.

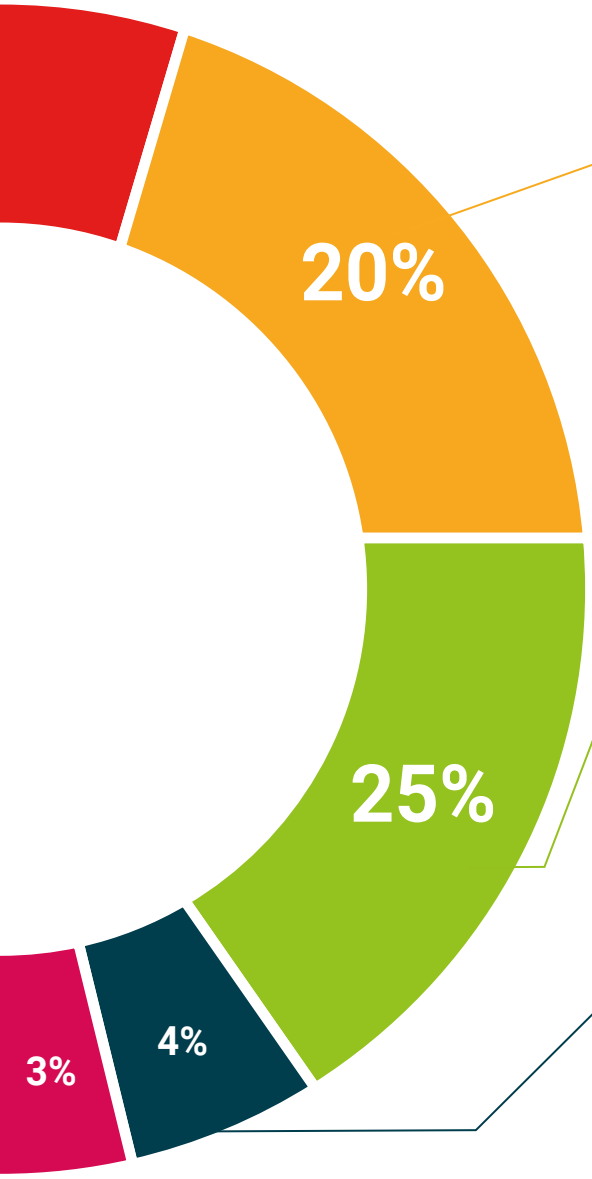
Практика навыков и компетенций

Студенты будут осуществлять деятельность по развитию конкретных компетенций и навыков в каждой предметной области. Практика и динамика приобретения и развития навыков и способностей, необходимых специалисту в рамках глобализации, в которой мы живем.

Дополнительная литература

Новейшие статьи, консенсусные документы и международные руководства включены в список литературы курса. В виртуальной библиотеке TECH студент будет иметь доступ ко всем материалам, необходимым для завершения обучения.





Метод кейсов

Метод дополнится подборкой лучших кейсов, выбранных специально для этой квалификации. Кейсы представляются, анализируются и преподаются лучшими специалистами на международной арене.

Интерактивные конспекты

Мы представляем содержание в привлекательной и динамичной мультимедийной форме, которая включает аудио, видео, изображения, диаграммы и концептуальные карты для закрепления знаний.

Эта уникальная обучающая система для представления мультимедийного содержания была отмечена компанией Microsoft как "Европейская история успеха".

Тестирование и повторное тестирование

На протяжении всей программы мы периодически оцениваем и переоцениваем ваши знания с помощью оценочных и самооценочных упражнений: так вы сможете убедиться, что достигаете поставленных целей.

07

Квалификация

Специализированная магистратура в области Программирование для frontend и fullstack-разработчиков гарантирует, помимо самого строгого и современного обучения, получение диплома об окончании Специализированной магистратуры, выдаваемого TECH Технологическим университетом.

Успешно пройдите эту программу и получите университетский диплом без хлопот, связанных с поездками и оформлением документов"

Данная **Специализированная магистратура в области Программирование для frontend и fullstack-разработчиков** содержит самую полную и современную программу на рынке.

После прохождения аттестации студент получит по почте* с подтверждением получения соответствующий диплом **Специализированной магистратуры,** выданный **ТЕСН Технологическим университетом.**

Диплом, выданный **ТЕСН Технологическим университетом**, подтверждает квалификацию, полученную в Специализированной магистратуре, и соответствует требованиям, обычно предъявляемым биржами труда, конкурсными экзаменами и комитетами по оценке карьеры.

Диплом: **Специализированная магистратура в области Программирование для frontend и fullstack-разработчиков**

Формат: **онлайн**

Продолжительность: **12 месяцев**


tech технологический университет

Настоящий

ДИПЛОМ

свидетельствует о том, что

Г-н/Г-жа _____, с документом, удостоверяющий личность_____

Освоил(а) и успешно прошел(ла) аккредитацию программы

СПЕЦИАЛИЗИРОВАННАЯ МАГИСТРАТУРА

в области

Программирование для frontend и fullstack-разработчиков

Данный диплом специализированной программы, присуждаемый Университетом, соответствует 1500 учебным часам, с датой начала дд/мм/гггг и датой окончания дд/мм/гггг.

ТЕСН является частным высшим учебным заведением, признанным Министерством народного образования Мексики с 28 июня 2018 года.

17 июня 2020 г.

Д-р Tere Guevara Navarro
Ректор

unique TECH code: AFWOR23S techtitute.com/certificates


Специализированная магистратура в области Программирование для frontend и fullstack-разработчиков

Общее распределение учебного плана

Тип предмета	Часы
Обязательный (Обязат.)	1500
Необязательный (Необязат.)	0
Внешние стажировки (ВС)	0
Магистерская диссертация (МД)	0
Всего	1500

Общее распределение учебного плана

Курс	Предмет	Часы	Характер
1°	Fullstack-разработка	150	Обязат
1°	Frontend в программировании	150	Обязат
1°	Язык JAVASCRIPT применительно к fullstack-разработке	150	Обязат
1°	Макет веб-сайтов применительно к fullstack-разработке	150	Обязат
1°	Инструменты JavaScript. Библиотека ReactJs	150	Обязат
1°	Фреймворк для JavaScript. Angular	150	Обязат
1°	Программирование на языке NodeJS	150	Обязат
1°	Базы данных для fullstack-разработчиков	150	Обязат
1°	UX CX. Клиентский опыт	150	Обязат
1°	Непрерывная интеграция и развертывание приложений	150	Обязат

Д-р Tere Guevara Navarro
Ректор

tech технологический университет

*Гаагский апостиль. В случае, если студент потребует, чтобы на его диплом в бумажном формате был проставлен Гаагский апостиль, TECH EDUCATION предпримет необходимые шаги для его получения за дополнительную плату.

tech технологический университет

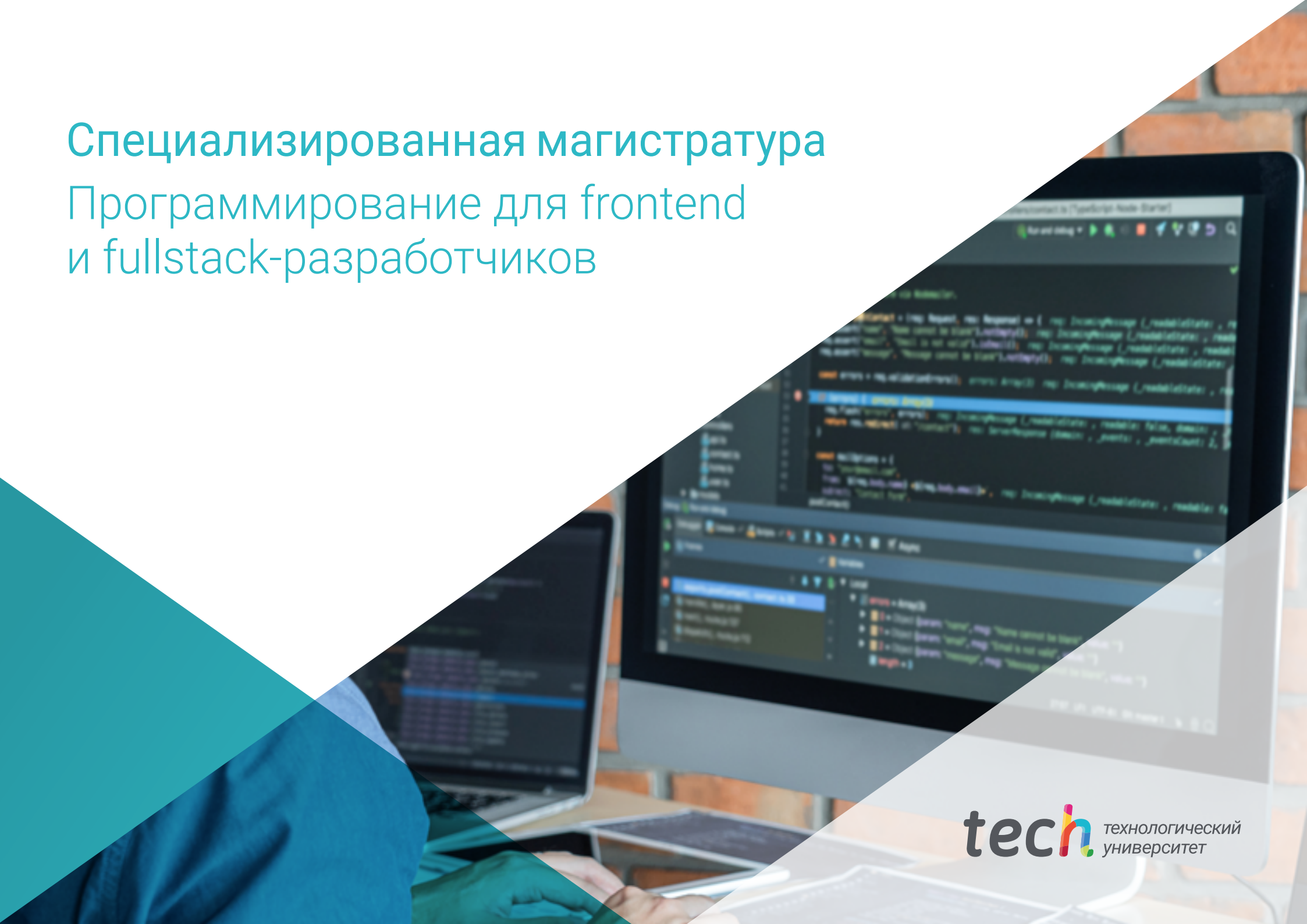
Специализированная магистратура

Программирование для frontend и fullstack-разработчиков

- Формат: **онлайн**
- Продолжительность: **12 месяцев**
- Учебное заведение: **TECH Технологический университет**
- Расписание: **по своему усмотрению**
- Экзамены: **онлайн**

Специализированная магистратура

Программирование для frontend и fullstack-разработчиков



tech технологический университет