

Учебная программа курса “Frontend FullStack Developer. Frontend разработчик”

Учебный центр MYEDU, принадлежащий Myedu OÜ Утверждено 19.09.2024	
Название учебной программы	Frontend FullStack Developer. Frontend разработчик
Группа учебных программ	153135 Разработка и анализ программного обеспечения и приложений / Tarkvara ja rakenduste arendus ning analüüs
Цели обучения	К концу обучения учащийся соответствует уровню Middle Frontend разработчика, знает полный цикл веб-разработки и может создавать простые веб-приложения
Знания и навыки, приобретенные при успешном завершении учебной программы. Достижимые результаты обучения	Компетенции учащегося к концу обучения: HTML/CSS: освоит создание адаптивных веб-страниц с использованием HTML, Flexbox и Grid Layout. JavaScript/TypeScript: получит навыки работы с JavaScript и TypeScript для динамического контента, ООП и управления DOM. SQL/PostgreSQL/MongoDB: научится работать с реляционными и нереляционными базами данных, используя SQL, PostgreSQL и MongoDB, а также управлять транзакциями и CRUD-операциями. React/Redux/Angular: освоит создание динамичных интерфейсов с использованием React, Redux и Angular для эффективного управления состоянием приложений. Node.js/Express: получит опыт разработки серверной части приложений и настройки API. Docker: научится использовать Docker для контейнеризации приложений. GitHub: освоит управление версиями кода и совместную работу над проектами с помощью Git. Linux/Bash: изучит основы работы в Linux и Bash для управления окружением. Rest API/Postman: научится создавать и тестировать Rest API с использованием Postman. Heroku/Firebase/AWS/Azure/Google Cloud: получит опыт развертывания приложений в облачных платформах.
Для кого этот курс. Целевая группа	Специалисты, которые работают или планируют работать в сфере ИТ, и хотят получить базовые знания и навыки, необходимые для работы в качестве Frontend FullStack-разработчика в Эстонии и за ее пределами
Требования на начало обучения, предъявляемые к обучаемому лицу, если они являются предпосылкой достижения результатов обучения	Для участия в обучении требуются базовые знания работы с компьютером. Учащийся должен иметь компьютер или ноутбук с оперативной памятью не менее 4 Гб с доступом в Интернет
Язык обучения	Русский язык
Общий объем обучения, в том числе доля аудиторной, практической и самостоятельной работы	400 академических часов
Самостоятельная работа	240 академических часов
Работа в аудитории	160 академических часов
Продолжительность обучения	12 месяцев

<p>Содержание обучения: - Структура и объем обучения - Содержание обучения</p>	<p>Программа обучения создана на основе профессионального стандарта Noorem tarkvaraarendaja, tase 4 (Junior Software Developer, EstQF Level 4, Младший разработчик программного обеспечения, уровень 4) Теоретический материал и практическая работа под руководством преподавателя</p>
	<p>Модуль 1. Web Core <i>16 вебинаров, 32 академических часа аудиторной работы и 48 академических часов самостоятельной работы</i> Студент освоит основы создания веб-страниц с использованием HTML и CSS, включая семантику, стилизацию элементов, Flexbox, отзывчивую верстку и анимации. Также изучит JavaScript: переменные, циклы, функции, объекты, работу с DOM и событиями. Важное внимание уделяется асинхронности, работе с API, а также CRUD-операциям для взаимодействия с бэкендом. В конце модуля студент выполнит итоговый проект, демонстрируя свои знания</p>
	<p>Модуль 2. FULLSTACK <i>16 вебинаров, 32 академических часа аудиторной работы и 48 академических часов самостоятельной работы</i> В модуле изучаются основные принципы работы с сетью и протоколом HTTP. Студенты познакомятся с TypeScript, включая основы, функции, интерфейсы, классы, дженерики и интеграцию в проекты. Основное внимание уделяется Docker: установка, работа с образами, Docker Compose и оркестрация. В части SQL охватываются основы работы с реляционными базами данных, включая SQL-запросы, JOIN, подзапросы и практические примеры. В конце модуля студенты выполняют итоговый проект</p>
	<p>Модуль 3. React <i>16 вебинаров, 32 академических часа аудиторной работы и 48 академических часов самостоятельной работы</i> В модуле изучаются основные аспекты работы с React: JSX, Props, стилизация, работа с событиями и состоянием, формы, жизненный цикл компонента и взаимодействие с API. Студенты освоят React-хуки, навигацию в приложении, разделение кода (code splitting), работу с Redux (основы, middleware, асинхронный Redux, селекторы), а также настройку маршрутизации (приватные и публичные маршруты). Также будут рассмотрены вопросы деплоя веб-приложений, CI/CD, работа с Docker, PostgreSQL, и облачными платформами (AWS, Azure, Google Cloud). Завершается модуль выполнением итогового проекта</p>
	<p>Модуль 4. Node.js & React Native <i>16 вебинаров, 32 академических часа аудиторной работы и 48 академических часов самостоятельной работы</i> Модуль охватывает основы Node.js и Express, включая создание консольных приложений, CRUD-операции, работу с MongoDB и Mongoose, аутентификацию, обработку изображений и тестирование. Также изучаются Docker, вебсокеты и деплой на Heroku. В React Native фокус на подготовке к работе, стилизации, работе с формами, навигации, нативных компонентах, Redux и Firebase. Включает деплой приложений на Expo и выполнение итогового проекта</p>

	<p>Модуль 5. Angular 16 вебинаров, 32 академических часа аудиторной работы и 48 академических часов самостоятельной работы</p> <p>Основы Angular: настройка среды, работа с компонентами, шаблонами и директивами. Сервисы и внедрение зависимостей. Маршрутизация и навигация. Формы и взаимодействие с сервером через HTTP. Управление состоянием приложения. Тестирование и оптимизация.</p> <p>Также включает Architect Skills: проектирование баз данных, оптимизация, транзакции, и дополнительные темы, такие как NoSQL, кластеризация и Big Data. Завершается итоговым проектом</p> <p>Итоговый проект Групповой проект, требования к которому основаны на реальных потребностях компаний-работодателей. Сдача и защита проекта, добавление проекта в портфолио</p>
<p>Методы обучения</p>	<p>Аудиторная (онлайн) работа: теоретический материал (лекции и обсуждение практических примеров)</p> <p>Практическая работа (онлайн и самостоятельно): выполнение заданий, разработка программного обеспечения, оформление документации. Все практические задания выполняются в среде IntelliJ IDEA, github и проверяются преподавателем. Дается детальная обратная связь в письменном виде. Преподаватель объясняет новый теоретический материал, отвечает на вопросы студентов по новой теме и предыдущему домашнему заданию. Занятия проходят в Zoom и Discord. Каждый урок записывается, запись доступна закрытой группе в Slack. Между лекциями проходят квесты, задач, тесты, дополнительные видео уроки. Преподаватель доступен для связи каждый день. Если студент пропустил лекцию, посмотрел её в записи и появились вопросы по задачам/теории – он может написать преподавателю свои вопросы и получить ответ в любой день</p> <p>Практические упражнения: задания на разработку алгоритмов, применение типов данных и структур, освоение различных инструментов сборки и запуска приложений</p> <p>Самостоятельная работа: прочтение дополнительных материалов по теме и просмотр обучающих видео</p> <p>Общение с одноклассниками и преподавателями в закрытом чате в Slack</p>
<p>Описание учебной среды и учебных помещений, предметов обстановки и оборудования, необходимых для достижения целей, поставленных в учебном плане, а также приобретения навыков и их соответствия законодательным требованиям охраны здоровья, если таковые установлены</p>	<p>Занятия с преподавателем проходят в форме онлайн видео конференций на платформе Zoom. В учебной среде проводятся теоретические и практические занятия, рассылаются домашние задания. Учащийся задает свои вопросы и получает обратную связь от учителя. Количество студентов в одной группе: до 10 человек. Один академический час длится 45 минут. Каждая встреча длится 2 академических часа. Занятия проводятся два раза в неделю по будням вечером.</p>
<p>Перечень учебных материалов</p>	<p>Учебный материал предоставляется учащимся в электронном виде.</p> <ul style="list-style-type: none"> - конспекты лекций - шаблоны проектов и документации - учебные тренажеры

Требования к завершению обучения, включая методы оценки и критерии оценки	Посещение онлайн-занятий; выполнение домашних заданий Для оценки результатов обучения используется итоговое тестирование
Условия окончания обучения и выдаваемые документы	<p>Для успешного завершения обучения требуется принять участие по крайней мере в 80% академических и практических занятий и выполнить все домашние задания. Достижение результатов обучения оценивается через практическую работу и итоговое тестирование.</p> <p>Сертификат выдается учащемуся, который посетил не менее 80% уроков, выполнил все практические задания и успешно сдал итоговый тест.</p> <p>Свидетельство об участии в обучении или об окончании обучения выдается ученику в том случае, если результаты обучения не были достигнуты, но ученик принимал участие в обучении.</p> <p>Справка выдается в соответствии с количеством посещенных часов обучения, если учащийся участвовал по крайней мере в половине занятий.</p>
Описание квалификации, опыта учёбы или работы проводящего обучение лица, которые необходимы для прохождения обучения	Senior Developer с многолетним опытом в разработке и менторингу по Frontend и карьерным вопросам Team Lead