**Python - Разработчик**

**Python Basic**

1. Введение.
2. Основы работы с Python.
3. Операторы, выражения.
4. Условный оператор if, ветвления.
5. Условный оператор if: продолжение.
6. Цикл while.
7. Цикл for: циклы со счётчиком.
8. Цикл for: особенности работы с range.
9. Цикл for: работа со строками.
10. Вложенные циклы.
11. Числа. Типы int и float.
12. Функции: начало.
13. Особенности float.
14. Установка и настройка IDE.
15. Базовые коллекции. Списки.
16. List: методы работы со списками.
17. Представления списков.
18. Базовые коллекции. Строки.
19. Базовые коллекции. Словари.
20. Базовые коллекции. Кортежи.
21. Функции: продолжение.
22. Работа с файлами.
23. Исключения: работа с ошибками.
24. Введение в ООП.
25. Основные принципы ООП.
26. Итераторы и генераторы.
27. Декораторы: базовый уровень.
28. Углублённая работа с классами и встроенные декораторы.
29. Декораторы: продвинутый уровень.
30. Функции: конец.
31. Библиотеки для работы с данными.
32. Бонусный модуль. Основы работы с сетью.
33. Дипломный проект. Telegram-бот для поиска отелей.

**Python Django**

1. Введение в веб-фреймворки.
2. Введение в Django.
3. Обработка запросов в Django.
4. База данных и модели.
5. Формы.
6. Работа с файлами.
7. Тестирование.
8. Аутентификация и авторизация.
9. Админка.
10. Локализация и интернационализация.
11. Права доступа.
12. Документирование.
13. Оптимизация.
14. Экспорт данных.
15. Стажировка. Разработка интернет-магазина в команде с тимлидом.

**Python Advanced**

1. Flask: начало.
2. Основы работы в Linux. Часть 1.
3. Основы культуры CI. Часть 1.
4. Flask.
5. Основы работы в Linux. Часть 2.
6. Основы дебаггинга и профилирования. Часть 1.
7. Logger.
8. Деплой приложения на linux.
9. Docker.
10. Базы данных. Часть 1.
11. Многозадачность. Часть 1.
12. Многозадачность. Часть 2.
13. Базы данных. Часть 2.
14. Вводный модуль по MVC-фреймворку. Часть 1.
15. Вводный модуль по MVC-фреймворку. Часть 2.
16. Базы данных. Часть 3.
17. RESTful API.
18. MVC, язык шаблонов. Часть 3.
19. Базы данных. Часть 4.
20. Знакомство с ORM SQLAlchemy. Часть 1.
21. ORM SQLAlchemy. Часть 2.
22. CGI server.
23. Основы дебаггинга и профилирования. Часть 2.
24. Основы дебаггинга и профилирования. Часть 3.
25. Многозадачность: asyncio.
26. Многозадачность: fastAPI.
27. Postgres + Миграции.
28. Pytest, Mock.
29. Безопасность.
30. Основы культуры CI: linters.
31. CD, автоматический деплой.
32. Дипломный проект. Разработка задачника по Python.

Веб-вёрстка. Базовый уровень

1. Введение.
2. Базовый HTML.
3. Базовый CSS.
4. Работа с макетом.
5. Layout. HTML-разметка.
6. Layout. Flexbox.
7. Layout. Стилизация.
8. Продвинутый HTML.
9. Продвинутый CSS.
10. Сетки.
11. Адаптивность. Десктоп и планшет.
12. Адаптивность. Мобильные устройства.
13. JavaScript для верстальщика.
14. Доступность.
15. Кросс-браузерность.
16. База знаний.
17. Работа с хостингом.
18. Дипломный проект — вёрстка лендинга.

**SQL**

1. SQL. Введение.
2. Инструкции по установке MySQL.
3. Структура БД, запрос DESCRIBE.
4. Выбор и фильтрация данных, запрос SELECT.
5. Функции и выражения, агрегация данных.
6. Отношения и соединение таблиц: JOIN.
7. Группировка.
8. Модификация данных.
9. Подзапросы.
10. Объявление и изменение структуры данных.

**GIT**

1. Версии программного кода.
2. Установка Git.
3. Индекс и частичные коммиты.
4. Сравнение версий.
5. Отмена изменений и откат версий.
6. Репозитории и коллективная работа.
7. Ветки — создание и управление.
8. Слияние и разрешение конфликтов.
9. Полезные инструменты.
10. Правила работы с Git.

#### Английский для IT-специалистов

1. IT Resume and CV.
2. Job interview: questions and answers.
3. Teamwork.
4. Workplace communication.
5. Business letter.
6. Software development.
7. System concept development and SRS.
8. Design.
9. Development and Testing.
10. Deployment and Maintenance.

**Fullstack-Python**

**Веб-вёрстка Базовый уровень**

1. Знакомство с курсом.
2. Базовый HTML.
3. Базовый CSS.
4. Работа с макетом.
5. Layout. HTML-разметка.
6. Layout. Flexbox.
7. Layout. Стилизация.
8. Продвинутый HTML.
9. Продвинутый CSS.
10. Сетки.
11. Адаптивность. Десктоп и планшет.
12. Адаптивность. Мобильные устройства.
13. JavaScript для верстальщика.
14. Доступность.
15. Кросс-браузерность.
16. База знаний.
17. Работа с хостингом.
18. Анимация.

**Продвинутый уровень**

1. Вёрстка HTML-писем.
2. SVG.
3. Анимация на JS.
4. Оптимизация и процесс загрузки.
5. Сборщики.
6. Препроцессоры и Постпроцессоры.
7. CSS Grid.
8. CMS.
9. База знаний: часть 2.
10. Будущее вёрстки.

**JavaScript. Базовый уровень**

1. Введение.
2. Переменные и работа с числами.
3. Строки, boolean и условные операторы.
4. Массивы и циклы.
5. Функции.
6. Объекты.
7. Введение в DOM.
8. Константы, области видимости и замыкания.
9. Преобразование типов и нестрогие сравнения.
10. Обработка событий и пользовательский ввод.
11. Разработчик и soft skills.

**Продвинутый уровень**

1. Модули.
2. npm, package.json.
3. Event loop и асинхронная разработка.
4. Методы объекта и прототипы.
5. Тестирование.
6. Обработка ошибок.
7. Общение с сервером, async/await.
8. События.
9. ООП и его реализация в JavaScript.
10. ФП, работа с массивами и строками.

**Python Basic**

1. Введение
2. Основы работы с Python
3. Операторы, выражения
4. Условный оператор If, ветвления
5. Условный оператор if: продвинутые условия
6. Цикл while
7. Цикл for. Циклы со счётчиком
8. Цикл for. Параметры функции range()
9. Числа
10. Базовые коллекции: list
11. Цикл for: работа со списками
12. List comprehensions, slice и сортировка списков
13. Двумерные массивы
14. Базовые коллекции: кортежи и строки
15. Базовые коллекции: словари
16. Функции. Основы
17. Функции. Продвинутые темы
18. Базовые коллекции: множества
19. Алгоритмы и структуры данных
20. Функции. Оформление и документация
21. Работа с файлами
22. Поиск в подстроке с помощью деревьев и regex
23. Функции. Рекурсия
24. Введение в ООП
25. Работы с модулями
26. Работа с файловой системой
27. Элементы функционального программирования
28. Алгоритмы
29. Структуры данных
30. Принципы ООП
31. ООП. Продвинутые темы
32. Исключения
33. Протоколы, генераторы, декораторы и менеджеры контекста
34. Основы работы с сетью
35. Продвинутые темы

**Python Advanced**

1. Основы работы в Linux
2. Основы культуры CI
3. Основы дебаггинга и профилирования
4. Основы CI/CD
5. Тестирование приложения
6. HTML, DOM etc
7. Networking
8. Многозадачность
9. Базы данных
10. Индексы
11. Протоколы Python
12. Flask
13. Async web
14. Основы CI/CD N
15. Основы CI/CD N+1
16. Docker. Что это такое?
17. RESTful API
18. Знакомство с ORM
19. CGI server
20. Web development
21. Безопасность
22. Командный дипломный проект

**Vue.js**

1. Основы Vue.
2. Создание проекта.
3. Каталог товаров.
4. Пагинация и фильтрация.
5. Страница товара.
6. Корзина.
7. Работа с API. Список товаров.
8. Работа с API. Оформление заказа.
9. Деплой.

**React**

1. Введение в React.
2. Создание проекта.
3. Компоненты на примере новостной карточки Reddit.
4. Больше о компонентах и введение в тестирование компонентов.
5. Работа с публичным API на примере Reddit.
6. Введение в Redux на примере новостной ленты.
7. Использование Redux-Thunk на примере бесконечной ленты.
8. Сайдбар + роутинг.
9. Формы на примере комментариев и карточки поста.
10. Оптимизируем приложение.
11. SSR + Next.JS.
12. Бонусный модуль: MobX.

**Angular**

1. Введение в Angular.
2. Вывод списка товаров.
3. RxJS: поиск товаров.
4. Работа с http. Улучшенный список товаров.
5. RxJS & Components Advanced: корзина товаров.
6. Роутинг: страница товара.
7. Формы: checkout.
8. Авторизация и защищённые зоны.
9. Тестирование.
10. Способы организации обмена данными.
11. Advanced Practice.
12. Advanced Practice: куда расти.

# Автоматизированное тестирование на Python

**Unity**

#### Введение в DOTS и ECS

Узнаете, что такое DOTS и ECS, поймёте основные принципы их работы. Изучите теорию композиции игровых процессов и разберёте Pure- и Hybrid- подходы в паттерне ECS. Научитесь создавать и конвертировать префабы в сущности и компоненты.

#### Обработка пользовательского ввода в Unity

Познакомитесь с принципами работы в системе ввода Unity. Научитесь перемещать персонажа, стрелять из оружия при нажатии на кнопку «Огонь» и перезаряжать его.

#### Взаимодействие объектов в Unity

Изучите основные законы физики, которые воспроизводятся в компьютерных играх. Научитесь строить системы взаимодействий объектов: нанесения урона, столкновения, реакции на попадание снаряда.

#### Хранение и использование статичных данных в Unity

Поймёте, как сохранять настройки игры в Scriptable Objects. Познакомитесь с форматами данных XML и JSON. Научитесь работать с библиотеками Newtonsoft и хранилищами данных AWS, Amazon, Dedicated. Узнаете, как разворачивать собственные сетевые решения на PHP и сохранять данные на сервере.

#### Создание искусственного интеллекта в играх

Узнаете роль искусственного интеллекта в играх и познакомитесь с его основными алгоритмами. Научитесь создавать искусственный интеллект на основе Utility System и использовать сторонние библиотеки.

#### Продвинутый подход к программированию: паттерны

Познакомитесь с теорией паттернов в программировании. Узнаете, какие паттерны существуют и научитесь выбирать подходящие под ваши задачи. Разберётесь, как внедрять и использовать паттерн DI на примере фреймворка Zenject.

#### Основы разработки графики в Unity

#### Поймёте, как устроена 3D-графика в компьютерных играх: познакомитесь с методами хранения, обработки и вывода 3D-данных. Научитесь создавать шейдеры и спецэффекты.

#### Освещение в игре

#### Разберётесь, как настраивать пайплайны и освещение в играх. Узнаете, как «запекать» свет.

#### Работа с анимацией в Unity

#### Поймёте, как анимировать персонажей и окружение в интерфейсе редактора Unity или с помощью кода. Научитесь создавать 2D и процедурную анимацию. Освоите продвинутые техники анимации.

#### Асинхронные процессы: от простого к сложному

#### Разберётесь, что такое асинхронные вычисления. Научитесь использовать функции Async/Await и Jobs. Познакомитесь с концепцией реактивного программирования и узнаете, как создавать и запускать код в параллельных потоках.

#### Расширение функционала редактора Unity

#### Поймёте, как расширить функционал редактора. Научитесь создавать собственные окна и инспекторы и работать в пакете Odin Inspector.

#### Менеджеры игровых состояний в Unity. Взаимодействие с UI

#### Научитесь создавать системы инвентаря, крафта и прокачки персонажа. Узнаете, как связать данные игры с UI.

#### Основы разработки сетевых игр на основе Photon

#### Поймёте, как создавать сетевые игры и выстраивать логику сетевых взаимодействий с помощью Photon Cloud.

#### Оптимизация производительности игр

#### Научитесь анализировать производительность игр с помощью Profiler. Сможете находить и исправлять недочёты в коде и структуре игры — это поможет оптимизировать её производительность.

#### Монетизация компьютерных игр

#### Научитесь анализировать поведение пользователей, внедрять внутриигровые покупки и читать игровые метрики. Поймёте, как монетизировать игры.

#### Работа со звуком

#### Поймёте, как озвучивать игры. Научитесь интегрировать в проект звуковой движок Wwise и работать в нём.

#### Сборка и публикация игр в мобильные сторы

#### Научитесь публиковать игры в сторах. Разберёте инструменты монетизации и аналитики: Applovin, App Metrica, Google Play Resolver. Узнаете об особенностях сборки проекта под Android и iOS.

#### Тестирование игровых проектов

#### Разберётесь, как тестировать программы и научитесь пользоваться фреймворком NUnit в Unity. Научитесь создавать скрипты для автоматизации тестирования.

#### Работа в команде и дальнейшее развитие

#### Научитесь работать в команде и отслеживать изменения в игре с помощью git. Сможете подготовиться к собеседованиям и выработать стратегию профессионального развития.

#### Финальная работа

#### Разработаете MVP-версию игры в жанре twin-stick shooter.

**Unity Разработчик игр**

Изучите основы разработки на Unity. Научитесь программировать на C#, работать с ассетами, создавать игровой ландшафт и персонажей. В рамках курса вы выполните 5 проектов.

1. Введение в Unity.
2. Работа с ассетами на сцене.
3. Знакомство с интерфейсами.
4. Переменные типы данных, операторы.
5. Игра-викторина.
6. Работа со временем в Unity.
7. 1 проект: игра Save the village.
8. Перемещение и вращение.
9. Знакомство с физическим движком. 2 модуля.
10. 2 проект: игра Wild Ball. 4 модуля.
11. Нарративный дизайн видеоигр.
12. 3 проект: платформер. 4 модуля.
13. Практикум C#.
14. Знакомство с Blender.
15. Props Modeling.
16. Blender: работа с материалами.
17. Low Poly character.
18. Создание игрового ландшафта.
19. TimeLines.
20. CineMachine.
21. Постпроцессинг.
22. Командная разработка.
23. 4 проект: игра Crazy Run. 2 модуля.
24. Геймдизайн.
25. AI и поиск пути.
26. Raycasts.
27. Финальная работа: игра в жанре shoot‘em up.

**Middle-Разработчик игр на Unity**

Углубитесь в разработку на Unity. Научитесь создавать AI, разрабатывать графику, работать с анимацией, собирать и публиковать игры. В рамках курса вы выполните 1 проект.

* 1. Введение в DOTS и ECS.
  2. Обработка пользовательского ввода в Unity.
  3. Взаимодействие объектов в Unity.
  4. Хранение и использование статичных данных в Unity.
  5. Создание искусственного интеллекта в играх.
  6. Продвинутый подход к программированию: паттерны.
  7. Основы разработки графики в Unity.
  8. Освещение в игре.
  9. Работа с анимацией в Unity.
  10. Асинхронные вычисления в Unity.
  11. Расширение функционала редактора Unity.
  12. Менеджеры игровых состояний в Unity. Взаимодействие с UI.
  13. Основы разработки сетевых игр на основе Photon.
  14. Оптимизация производительности игр.
  15. Монетизация игр.
  16. Работа со звуком.
  17. Сборка и публикация игр в мобильные сторы.
  18. Тестирование игровых проектов.
  19. Работа в команде и дальнейшее развитие.
  20. Финальная работа: игра в жанре twin-stick shooter.

**Blender**

Научитесь создавать детализированные сцены и визуализации. Сможете работать со сложными эффектами. В рамках курса вы выполните 5 проектов.

* 1. Вступление.
  2. Основы интерфейса.
  3. Основы моделирования. Полигональное моделирование.
  4. Полигональное моделирование под сглаживание.
  5. Основы моделирования. Процедурное моделирование.
  6. Основы моделирования. Скульптинг.
  7. Основы анимации и симуляции.
  8. UV-развёртка.
  9. Работа с текстурами.
  10. Работа с материалами.
  11. Визуализация и освещение.
  12. Визуализация в Octane render.
  13. Усложнённые симуляции.
  14. Композитинг в блендере**.**

**Профессия C#-разработчик**

Разберётесь в основах ООП, освоите базовые возможности C# и визуальное программирование.

* 1. Введение в профессию. NET Developer.
  2. Базовые возможности C#.
  3. Управляющие конструкции: ветвление и циклы.
  4. Массивы.
  5. Методы.
  6. Файлы.
  7. Структуры и введение в ООП.
  8. Коллекции.
  9. Работа с сетью.
  10. Визуальное программирование. Основы технологии Windows Presentation.
  11. Глубокое погружение в ООП. 3 модуля.
  12. Делегаты.
  13. Дополнительный материал.
  14. Многопоточность.
  15. ADO.
  16. Entity Framework.
  17. Паттерны.
  18. ASP.NET Core. Общие понятия и архитектура.
  19. ASP.NET Core. Работа с данными.
  20. Веб-API ASP.NET Core.
  21. Что ещё нужно знать desktop- и backend-разработчику.
  22. Финальная работа: создание собственной CRM.

**Git**

Выучите основные команды Git и узнаете, как в командах организуют работу по веткам.

* 1. Работаем с Git на своём компьютере.
  2. Работаем с удалённым репозиторием.
  3. Командная работа в Git.
  4. Сравнение версий и отмена изменений.
  5. Инструменты и правила работы с Git.

**Веб-вёрстка. Базовый уровень**

Познакомитесь со стандартами Web 2.0, научитесь работать с макетами и форматировать код, освоите адаптивную вёрстку. В рамках курса вы выполните 1 проект.

* 1. Введение.
  2. Базовый HTML.
  3. Базовый CSS. 2 модуля.
  4. Подготовка к вёрстке.
  5. Layout. HTML-разметка.
  6. Layout. Flexbox.
  7. Layout. Стилизация.
  8. Продвинутый HTML. Формы.
  9. Продвинутый CSS.
  10. Адаптивность. Десктоп.
  11. Адаптивность. Мобильные устройства.
  12. JavaScript для верстальщика.
  13. Доступность.
  14. Кросс-браузерность.
  15. База знаний.
  16. Сетки.
  17. Работа с хостингом.
  18. Анимация.
  19. Финальная работа: вёрстка лендинга по брифу.

**Python**

**Git**

* Введение в Git
* Установка и настройка Git
* Основные команды терминала
* Работа с репозиториями в Git
* Управление файлами репозитория
* История изменений
* Работа с ветками репозитория
* Публикация репозитория
* Слияния веток
* Управление версиями
* Создание pull request
* Сложные операции
* Работа с fork-репозиториями
* Введение в Python
* Встроенные типы и операции с ними
* Практикум. Игра «Угадай число»
* Функции
* Модули и библиотеки
* Работа с файлами. Кодировки
* Полезные инструменты. Обработка исключений
* Практикум. Консольный файловый менеджер

**Backend-developer**

##### Основы языка Python

* Знакомство с Python
* Встроенные типы и операции с ними
* Функции
* Работа с модулями и пакетами
* Генераторы и comprehensions. Множества
* Работа с файлами
* Работа с файловой системой. Исключения в Python
* Декораторы
* Объектно-ориентированное программирование
* ООП. Продвинутый уровень
* ООП. Полезные дополнения

##### Linux. Рабочая станция

* Введение. Установка ОС
* Настройка и знакомство с интерфейсом командной строки
* Пользователи. Управление пользователями и группами
* Загрузка ОС и процессы
* Устройство файловой системы Linux. Понятия файла и каталога
* Введение в скрипты Bash. Планировщики задач crontab и at
* Управление пакетами и репозиториями. Основы сетевой безопасности
* Введение в Docker

##### Основы реляционных баз данных. MySQL

* Вебинар. Установка окружения. DDL-команды
* Видеоурок. Управление БД. Язык запросов SQL  
  Вебинар. Введение в проектирование БД
* Вебинар. CRUD-операции
* Видеоурок. Операторы, фильтрация, сортировка и ограничение. Агрегация данных
* Вебинар. Операторы, фильтрация, сортировка и ограничение. Агрегация данных
* Видеоурок. Сложные запросы
* Вебинар. Сложные запросы
* Видеоурок. Транзакции, переменные, представления. Администрирование. Хранимые процедуры и функции, триггеры
* Вебинар. Транзакции, переменные, представления. Администрирование. Хранимые процедуры и функции, триггеры
* Видеоурок. Оптимизация запросов. NoSQL
* Вебинар. Оптимизация запросов

##### Алгоритмы и структуры данных на Python. Базовый курс

* Введение в алгоритмизацию и реализация простых алгоритмов на Python
* Циклы. Рекурсия. Функции
* Массивы. Кортежи. Множества. Списки
* Эмпирическая оценка алгоритмов на Python
* Коллекции. Список. Очередь. Словарь
* Работа с динамической памятью
* Алгоритмы сортировки
* Деревья. Хеш-функция

##### Итоги четверти

Вы получите навыки работы с базами данных и ОС Linux. Реализуете около 30 алгоритмов с ветвлениями, циклами и рекурсиями от простых до сложных

**Fronten and Backend web-**

##### HTML/CSS. Интерактивный курс

* Основные понятия в веб-разработке
* Основы языка разметки документов HTML
* Основы языка оформления стилей документа CSS
* Основные теги для вёрстки, псевдоклассы и псевдоэлементы
* Основы позиционирования, работа с Flexbox
* Работа с макетом дизайна в формате Figma
* Основы создания адаптивного сайта
* Стандарты веб и вспомогательные инструменты

##### Базовый курс JavaScript

* Основы языка JavaScript
* Основные операторы JavaScript
* Циклы, массивы, структуры данных
* Объекты в JavaScript
* Введение в DOM
* Обработка событий в JavaScript
* Урок-практикум
* Анонимные функции, замыкания

##### Основы Django Framework

Вся работа на протяжении курса будет связана с созданием проекта, который позволит на основе практического опыта приобрести навыки, востребованные на современном рынке труда.

* Подготовительная работа
* Старт проекта
* Система шаблонов Django
* Взаимодействие с базой данных через Django ORM
* Пользователи в Django
* Административный раздел Django и работа с формами
* Логирование, кэш и отложенные задачи
* Пагинация, интернационализация и тестирование
* Развертывание проекта и профилирование

##### Итоги четверти

Вы научитесь создавать быстрые и безопасные сайты.

Создадите проект учебной платформы.

Изучите Django Framework: менеджеры моделей, отправка почтовых сообщений, создание и оптимизация сложных запросов к базе данных, работа с наборами форм, развёртывание Django-проекта на web-сервере и тестирование.

**Продвинутый Python**

##### Django REST framework

* Введение в REST и Django REST framework
* Введение в React. Компонентный подход
* Serializers. Renderers. Routers
* Views. Filtering. Pagination
* Routing. SPA
* Авторизация. Система прав
* Авторизация на стороне клиента
* Тестирование API. Фабрики данных
* Система версий API. Документация для API
* GraphQl
* Работа с формами. Сборка проекта для production
* Развертывание проекта с docker и docker-compose

##### Асинхронный чат

* Принципы проектирования комплексных систем. Проектирование чата
* Первый простейший протокол обмена сообщениями
* Работа сервера с множеством клиентов
* Конкурентность. Кооперативная многозадачность. Генераторы
* Конкурентность — часть 2. Многопоточность
* Конкурентность — часть 3. Параллелизм
* Конкурентность — часть 4. Итог
* Работа с высокоуровневым API Python asyncio
* Перевод чата на WebSocket-протокол
* Добавление функций для чата: комнаты, права, аутентификация и авторизация
* Работа с базами данных, асинхронные ORM
* Расширение клиента. Оформление кода в виде библиотеки
* Введение в Tkinter и основные принципы создания графических интерфейсов
* Функциональный интерфейс. События Tkinter, взаимодействие элементов интерфейса
* Рабочий клиент
* Подготовка сервера и клиента к распространению

##### Фреймворк Flask

* Знакомство: werkzeug, Flask”
* Шаблоны Jinja2. Комплексные приложения на Flask. Blueprints
* Авторизация пользователя и начало работы с базой данных. SQLAlchemy
* Docker, docker-compose, Postgres. Миграции схем Flask-Migrate и alembic
* WTForms. Регистрация и авторизация пользователя
* Модели автора и статьи; связи one-to-one и one-to-many
* Модель статьи: категории, теги; связи в БД
* Flask Admin
* JSON REST API. CRUD, swagger, marshmallow, flask-combo-jsonapi, combojsonapi
* Продолжаем работу с flask-combo-jsonapi, знакомимся с combojsonapi, система плагинов
* Heroku. Публикация приложения

##### Итоги четверти

Разработаете сетевой чат с возможностью создавать пользователей, искать и добавлять друзей, отправлять сообщения выбранному пользователю.

Изучите востребованный на рынке фреймворк Flask и создадите с его помощью новостной портал.

Познакомитесь с Django REST Framework в связке с ReactJS и создадите ещё один проект — ToDo планировщик с разделением ролей, собственным API и документацией.

Прохождение курсов в данной четверти возможно в любом порядке.

**Проект разработка**

##### Архитектура и шаблоны проектирования на Python

* Паттерны веб-представления
* Архитектура Python-приложений
* Принципы проектирования
* Порождающие паттерны
* Структурные паттерны
* Поведенческие паттерны
* Архитектурные системные паттерны
* Антипаттерны
* Микросервисная архитектура

##### Командная разработка по методологии Agile/Scrum

**Блок 1. Теория**

* Введение в Agile
* Scrum, Kanban, Scrumban
* Работа в команде
* Подготовка к старту проекта
* Подготовка к проекту практика

**Блок 2. Практика**8 недельных спринтов. Каждый спринт — встреча с SCRUM мастером.  
  
Описание проекта.  
‍  
**Проблема клиента:**У компании есть собственная LMS, но в ней отсутствует общая платформа, где бы выпускники могли обмениваться полезной информацией и формировать свою базу знаний.

**Задача студентов:  
  
‍**Русскоязычный веб-сайт в формате системы тематических коллективных блогов с элементами новостного сайта, созданный для публикации новостей, аналитических статей, мыслей, связанных с направлениями обучения в образовательной компании N.

##### Итоги четверти

Готовый бизнес-проект, разработанный в команде на основе вашей идеи. Научитесь писать код на Python, организовывать взаимодействие сервера на базе Linux с сервером баз данных.

##### Компьютерные сети. Интерактивный курс

* Основы компьютерных сетей. Технология Ethernet. Часть 1.
* Физический и канальный уровень. Технология Ethernet. Часть 2
* Сетевой уровень. Часть 1
* Сетевой уровень. Часть 2
* Транспортный уровень
* Углублённое изучение сетевых технологий. Часть 1
* Углублённое изучение сетевых технологий. Часть 2
* Прикладной уровень

##### Продвинутый курс JavaScript

* Современный JavaScript
* ООП в JavaScript
* Асинхронные запросы
* Регулярные выражения
* Фреймворк Vue.js
* Компоненты Vue.js
* JavaScript на сервере
* Тестирование и сборка

##### Подготовка к собеседованию Python-разработчика

* Python: синтаксис языка, базовые структуры данных, функциональное программирование
* Python: парадигма ООП, особенности и отличия от других ЯП
* Python: стандартная библиотека Python
* Django: основные понятия ORM, структура и особенности проектирования
* Django: AJAX, JavaScript, jQuery
* Базы данных: работа с БД в Python и Django. Особенности и различия
* Фреймворки PyQt и Kivy
* Linux: базовые навыки работы в серверной операционной системе

##### Методы сбора и обработки данных из сети Интернет

* Основы клиент-серверного взаимодействия.
* Парсинг API
* Парсинг HTML. BeautifulSoup, MongoDB
* Системы управления базами данных MongoDB и SQLite в Python
* Парсинг HTML. XPath
* Scrapy
* Scrapy. Парсинг фото и файлов
* Selenium в Python
* Работа с данными