

AUTOMATION COURSE

While students in large colleges study mass amounts of theory, we offer up to date, fresh and relevant QA and Testing Automation classes **focused on practical work methods, adapted to industry needs** so you can penetrate the job market with enough confidence and the right experience to do your job right.

Our classes are taught by industry experts, those who work simultaneously as interviewers and recruiters in high-tech companies and know exactly what it takes to succeed. Each student learns **only** what they need to know for their future jobs – for this reason, all candidates are screened and evaluated before admission in order to guarantee the highest level of learning and ensure future career opportunities.

What does this mean for you? You gain the best hands-on experience and pay less money - two birds, one stone.

Our knowledge, your future



Individuals

Our QA and Automation courses focus on practical knowledge; in class exercises, homework assignments and learning in small groups which allows for personal attention and better understanding of the material.



Companies

We offer customized QA and Automation courses and workshops according to your company needs. Course materials are suited to your everyday tasks and training requirements.



“Find Work”

Our goal is helping you achieve a total career upgrade and land that dream job. That's why we built a Professional Branding Intensive Course that gives our students a competitive edge over other candidates.

COMPANY DETAILS

HEADQUARTERS

Ramat Gan, Tel Aviv District

YEAR FOUNDED

2015

COMPANY TYPE

Educational Institution

COMPANY SIZE

40-50 employees

Who is this course for?

This course will teach you the most practical things required for getting an automation testing job. you will learn the most advanced automation concepts, how to design automation framework and how to solve the various complex issues that you may come across during your career.

This course is perfectly suitable for those who are:

- Software testers wanting to add automation testing to their toolbox.
- Software testers wishing to get an automation testing job.
- Automation testers seeking to upgrade their skills.
- QA engineers.
- Test Leads/Managers.
- Those who wish to learn automation testing process and frameworks.



Python development foundation

Developing using Python requires a bottom-up understanding of the language, its core constructs, and the solid foundation of the entire Python platform and development stack. In this module, you will gain the foundational knowledge of the Python platform.



Object-oriented concepts

The foundation of any software is the architecture. In this module, you will be immersed in an essential approach to software engineering: Object Oriented Analysis & Design. Gain skills in object-oriented concepts to model software that enables effective communication and useful abstraction.



Design patterns

Moving beyond merely writing valid code, it is crucial to develop insight into WHY, WHEN, and HOW to utilize design strategies most effectively. In this module, you will learn how to build solutions using Python for design patterns. After this model you will have the skills to deal successfully with any Python for automation related job interview question.



Frontend automation - selenium

We will review HTML, CSS, and JavaScript to level-set knowledge of frontend Automation development technologies. With that foundation in place, you will move on to learn the essential frontend automation capabilities using Selenium Framework necessary to build dynamic Automation tests for any web site.



Backend automation

In this module, you will build Backend services to test exposed data sent to the Frontend applications. Additionally, you will learn how to build data services, connect services to data stores, and perform query, mutation, and subscription operations.



Continuous development

Continuous development is the process of executing automated tests as part of the software delivery pipeline to obtain feedback on the business risks associated with a software release candidate as rapidly as possible. software release candidate as rapidly as possible.

THE INSTRUCTORS



Alex Kuznetsov
DevOps Team Lead
Ceva



Yuval Wilf
Senior Director of R&D
Cognyte



Daniel Gotlieb
DevOps Team Lead
Trigo



OUR ALUMNI WORK WITH THE BEST



COURSE CONTENT



45 Hours

Automation Professional
Course



90 Hours

Independent
Practice



24 Hours

Professional Branding
Booster-Course



Available for 6 Months

Job Search
Support



12 Hours

High-Tech English
Booster-Course

AUTOMATION DEVELOPMENT

Course description

Students will learn **Python development, automating full stack (Front and Backend) and how to establish automation infrastructure**. This course, which was designed by working professionals, will begin from the basics of automation testing using Selenium and continue to the most advanced automation testing topics. Students will be taught Python development, as well as advanced techniques that will ease their automation testing learning process.

TOPIC	DESCRIPTION
PYTHON CORE	
Overview and setup	<ul style="list-style-type: none"> • Introduction • Case studies • Market share • Python Interpreters • Compiling Vs. Interpreting • Installation • IDE setup • PyEnv • Venv
Python basics	<ul style="list-style-type: none"> • Data types • Variables • Statically Vs. dynamically typed • Local Vs. global variables • Casting & parsing • String Literals

	<ul style="list-style-type: none"> • String format • Constants • Syntax • Operators • Statements • Conditions • Loops • Data input • Comments • Reserved words
OOP	<ul style="list-style-type: none"> • Classes • Objects • Methods and functions • Function parameters • Function arguments • Function overload • Function overriding • Constructors • Packages • Inheritance • Programming conventions • Garbage collection • Regular expressions
Data structures	<ul style="list-style-type: none"> • Array • List • Tuple • Dictionary • Mutable vs immutable
Files (I/O)	<ul style="list-style-type: none"> • Files and paths • Reading • Writing • Modes • Text and binaries • File manipulations • Directories • Paths • Buffers • Streams
Error handling	<ul style="list-style-type: none"> • Errors Vs. Exceptions • Try • Except • Finally • Raise exceptions
Libraries and tools	<ul style="list-style-type: none"> • Debug • Libraries

	<ul style="list-style-type: none"> • Wheel files • Pip and PyPi • Requirements file • Poetry
Unit test	<ul style="list-style-type: none"> • What is a testing framework? • PyTest • Decorators • Assertions • Run order • Soft & hard fails • Test run and suites
Design patterns	<ul style="list-style-type: none"> • What are Design patterns? • Anti-design patterns • Programming conventions • Coupling and de-coupling • Page object • Conventions • Refactoring
Source control management	<ul style="list-style-type: none"> • What is Git? • Understanding Git • Terminology • Setup & configuration • Configuring Git • Repository overview • Building your repository • Branching and Merging • Commits • Resolving conflicts • Pull Requests • Remote repository
Python project	<ul style="list-style-type: none"> • Creating a Python program
FRONTEND AUTOMATION	
Selenium intro	<ul style="list-style-type: none"> • Overview • Selenium modules • Selenium components
HTML, CSS, JavaScript	<ul style="list-style-type: none"> • Introduction • Elements • Attributes • CSS • JavaScript • Ajax • Angular JS
Selenium WebDriver	<ul style="list-style-type: none"> • WebDriver Architecture

	<ul style="list-style-type: none"> • Drivers • WebDriver object methods • Driver options • Asserts and verifications • Switch and navigation • Working with iFrames • Synchronization • Reporting system • Page object model (POM) • Actions • Selenium keys • Alerts • Browser actions • Authorization
WebDriver elements	<ul style="list-style-type: none"> • Web elements overview • Locating Web elements • Web elements interaction • Locators overview • Locators types • Choosing the right locators • Relative locators • Controllers • Web elements data • File uploads • Elements screenshot
Headless Testing	<ul style="list-style-type: none"> • Headless testing usage • Advantages and disadvantages • Show cases • Headless types • Headless test execution
Final web project	<ul style="list-style-type: none"> • Selenium WebDriver project - "Automating Web Project"
BACKEND AUTOMATION	
REST API overview	<ul style="list-style-type: none"> • HTTP intro • Layers • What is API? • What is REST? • Why REST? • REST vs. SOAP • API overview • Rest methods • Status codes
JSON	<ul style="list-style-type: none"> • Overview • Structure • JSON rules • JSON parsing

REST hands-on	<ul style="list-style-type: none"> • Requests • REST client • REST CRUD operations • REST methods • Creating REST requests • Response processing • Working with a real API • Running automatic API tests • Selenium & API testing integration • API as Data Driven tests
Backend server	<ul style="list-style-type: none"> • Flask introduction • Routes • Path params • Query params • Binding • Debug mode
Relational Database	<ul style="list-style-type: none"> • Overview • PyMySQL • SQLAlchemy • Setup and configuration • MySQL drivers • Running queries from our code • Creating and working with a remote database • Connections • Statements and prepared statements • Security • DB operations
DB CRUD operations	<ul style="list-style-type: none"> • Select • Update • Insert • Delete • Indexing • Query profiling • Automating DB tests
Final backend project	<ul style="list-style-type: none"> • Building and automating backend server
CONTINUOUS DEVELOPMENT	
CI Intro	<ul style="list-style-type: none"> • Overview • Why continuous integration is used for? • Jenkins introduction • Jenkins Architecture
Jenkins core	<ul style="list-style-type: none"> • Setup & configuration • Jenkins configurations • Plugins • Pipelines

	<ul style="list-style-type: none"> • Alerts and notifications • Authentication • Authorization • Creating users
Build jobs	<ul style="list-style-type: none"> • What are build Jobs • Building your build jobs • Build triggers • Scheduling • CRON expressions • Reporting • Disabling and deleting jobs • Jenkins - Maven Integration • Automating automations • Nightly runs
Cloud testing	<ul style="list-style-type: none"> • Cloud introduction • SauceLabs • RemoteWebDriver • Browser pools • Parallel testing