AUTHORITY PARTNERS

AP Lab Guide

DOCUMENT BY

Authority Partners

200 Spectrum Center Dr, Suite 300
Irvine, CA 92618

DATE
September 2025

Table of Contents

1. AP Lab Overview

Who is AP Lab for?

Motivated, passionate builders (students or early-career tech enthusiasts) who want real-world software development and QA experience.

Program Format and Duration

A free training program held in person in Sarajevo at the AP Lab Center for Talent & Innovation. It runs for 12 weeks, with a commitment of at least 20 hours per workweek.

Career Outcomes

Trainees are eligible for junior roles in full-stack development or QA. Historically, about 50% of alumni have been hired by Authority Partners.

Technology Stack and Mentorship

Common technologies include C#, .NET Core, Angular, HTML/CSS/JavaScript, Bootstrap, xUnit, Azure DevOps, Entity Framework Core, MSSQL and Git/TFS. You'll get a dedicated mentor, guest lectures and hands-on project work covering the full software development lifecycle.

2. Application & Selection

Target Candidate Profile

- Candidates who show strong motivation, growth mindset, curiosity and some exposure to software development (school/side projects or self-study)
- A formal CS degree is not required; excellent English is expected

Required Time Commitment

Minimum 20 hours/week. Many participants pair the program with studies, but consistent effort is required.

Selection Process

1. Application review and grading
2. Shortlist invited to interview with the AP Lab Mentor and Talent Acquisition team
3. Selection of about 20 trainees for the session

Competition & Outcomes

Applications are competitive. The program is well-established, and about half of accepted trainees are eventually hired by Authority Partners.

3. How to Prepare & Apply

A. Preparation Checklist

1. **Sharpen your fundamentals.**
   - Object-oriented programming, HTTP/REST APIs, version control (Git), relational databases (SQL), front-end basics and software testing principles.

2. **Build a showcase project.**
   - Example: a backend API in C#/.NET + a simple Angular front-end. Include basic unit tests and version control using Git. Host it in a public repository.

3. **Plan your schedule.**
   - Reserve at least 20 hours/week for training, project work, mentor sessions and self-study.

4. **Adopt a learning mindset.**
   - Be coachable, ask questions and document progress (e.g., README files, mini-demos, weekly updates).

5. **Ask thoughtful questions.**
   - Before or during the application process, show initiative by asking things such as:
     - "Is it possible to attend AP Lab alongside university courses or a part-time job?"
     - "Is a remote or hybrid attendance option available?"
     - "How much one-on-one mentor time can I expect? Will there be code review sessions or office hours?"
     - "If I'm self-taught and don't have a CS degree, but have done side coding projects, can I still apply?"

B. What to Include in Your Application

When you write your application (CV, portfolio section and any cover letter or motivation statement), make sure to include:

1. **Project descriptions, with rich detail.**
   - For each project, clearly describe:
     - Project title and objective: what problem you were solving, or what functionality you aimed to build.
     - Your role and contributions: what parts of the project you handled, whether you designed APIs, built UI components, wrote database schemas, implemented business logic or developed tests.
     - Technologies used: e.g. "C#, .NET Core, Entity Framework, SQL Server, Angular, Bootstrap, GitHub, xUnit for unit testing."
     - Challenges you faced and how you addressed them: performance bottlenecks, tricky bugs, asynchronous flows, deployment issues, testing edge cases, etc.
     - Results or outcomes: did your project run successfully? Did you deploy it somewhere? Did it serve users, solve your chosen problem or help you learn a new technology?

    - What you learned: highlight clear takeaways; e.g. “I improved my understanding of asynchronous programming in C#,” or “I learned how to structure end-to-end tests with xUnit,” or “I strengthened my Git workflow including branching and pull requests.”

2. **Courses, extra learning and side projects.**
    - List relevant online courses, bootcamps, MOOCs, workshops or certifications you've completed (or are currently doing). Briefly note the key skills or technologies you learned, and ideally one or two small outcomes or mini projects you built as part of that course.
    - If you're self-studying, call that out explicitly: say “Self-study path: completed tutorials in REST API design (Postman, Swagger), built demo app with C#/.NET, did a Udemy course on Angular and read documentation/examples on Azure DevOps pipelines.”

3. **Motivation and growth mindset.**
    - Explain why you want to join AP Lab specifically: what attracts you to hands-on software engineering, and what role AP Lab plays in your career plan.
    - Illustrate your learning approach: describe a time when you taught yourself a technology or resolved a programming challenge on your own (or with limited guidance), how you persisted, overcame obstacles and reflected on what you learned.
    - Show your long-term view: for example, “I want to become a full-stack engineer working on web platforms,” or “I want to build high-quality software and improve my testing/automation skills,” and explain how AP Lab will help you with that goal.

4. **Availability and commitment.**
    - State clearly that you can dedicate 20 hours/week; if you have other time commitments (studies, job, family), briefly explain how you will balance AP Lab’s workload.
    - If you need relocation or travel to Sarajevo, state that clearly and possibly how you plan to manage it.

5. **Questions and initiative.**
    - End your application with 1-2 smart questions for the AP Lab coordinators or Talent Acquisition team. This shows you’re proactive, curious and serious. Examples might include:
        - “Can I combine AP Lab with university classes or a part-time job, and if so, what would balance look like?”
        - “Is hybrid or partial remote participation a possibility under special circumstances?”
        - “What will the weekly rhythm of mentor-supported project work be, in practice — how many hours of dedicated instruction, office hours, coding sprints and code review can I expect?”
        - “If accepted, is there anything you recommend candidates start learning or preparing in advance to make the most of the first weeks of training?”