Web Development - Learning Plan

🎯 Key Differentiators of This Program:

- **Realistic expectations** - No false promises about immediate employment
- **Broad IT awareness** - Not just coding, but understanding the ecosystem
- **Career guidance** - Real talk about Hungarian market, EPAM path, networking
- **Structured practice** - W3Schools + LeetCode, not just homework
- **Individual attention** - Round-robin approach ensures everyone gets spotlight
- **Community building** - Cohort supports each other beyond course
- **Holistic skills** - Tools (Git, Slack, Figma), soft skills (communication), technical skills
- **Continued support** - Relationship doesn't end after Session 20

🎓 Program Philosophy:

Core Understanding:

- This course is **foundation**, not job-ready training
- Goal: Give them **tools and direction** to continue learning
- Prepare them for **next-level programs** (EPAM, other mentorships)
- Build **broad IT awareness**, not just web dev tunnel vision
- **Realistic expectations** about EU junior market

🕐 Session Structure

Standard 90-minute Session Structure (Sandwich Method):

- 🕐 **0-10 min:** Welcome & Homework Review Overview
  - Quick wins: Show 2-3 excellent homework examples
  - Common mistakes: Address typical issues everyone had
  - Q&A on homework
- 🕐 **10-35 min:** NEW CONTENT - Theory + Live Coding Demo (25 min)
  - Introduce new concept (5-10 min theory)
  - Live coding demo (15-20 min)
  - "Follow along" or just watch (depends on complexity)
- 🕐 **35-40 min:** QUICK BREAK (5 min)
  - Biological necessity
  - Chat, questions, stretch
- 🕐 **40-75 min:** HANDS-ON PRACTICE (35 min)
  - Guided exercise applying what they just learned
  - I circulate, help individually
  - Pair programming encouraged
  - Mini-challenge for fast finishers

- 🕐 **75-85 min:** CAREER/PROFESSIONAL SKILLS (10 min)
  - LinkedIn tip of the week
  - Job search strategy
  - Industry insight
  - Professional tools demo
  - Women in tech resources
- 🕐 **85-90 min:** WRAP-UP (5 min)
  - Homework assignment explanation
  - What's next session
  - Motivation boost

Why this order:

- Start with homework when minds are fresh
- New content when energy is high
- Break before intense practice
- Practice while concept is still fresh
- Career stuff at end (easier content, nice conclusion)

📚 20-Session Structure:

- **Phase 1:** HTML Foundations (Sessions 1-3)
- **Phase 2:** CSS Mastery (Sessions 4-8)
- **Phase 3:** JavaScript Fundamentals (Sessions 9-13)
- **Phase 4:** React & Modern Development (Sessions 14-20)

🎓 Complete Learning Plan

Phase 1: HTML Foundations

SESSION 1: Welcome & HTML Basics

Main Topics:

- Program introduction & realistic expectations
- Web development overview & IT career landscape
- HTML basics: structure, semantic elements
- Setting up development environment

Learning Objectives:

- Understand course goals and what comes AFTER this course
- Create basic HTML page structure
- Know semantic HTML5 elements

- Environment ready (VS Code, Git, GitHub)

Resources:

- **Pre-session**:
  - Install VS Code, Git
  - Create GitHub account
  - Watch: "HTML Crash Course" (Traversy Media, 1hr)
- **In-session**: MDN HTML Basics guide
- **Post-session**:
  - W3Schools Git Tutorial
  - W3Schools HTML Tutorial (sections 1-10)

Hands-on Practice:

- Individual tasks on W3Schools HTML exercises
- Round-robin: Each participant completes one exercise, explains their approach
- I guide their thinking process publicly
- Start portfolio: basic HTML structure

Career/Professional (10 min):

- **IT Landscape Overview**:
  - Various IT roles: Frontend, Backend, Full-stack, DevOps, QA, Data, Security
  - How HTML/CSS/JS fits into each
  - "You're learning universal skills, not just one career path"
- **Reality Check**: Junior market in Hungary/EU
  - Why it's harder for non-EU graduates
  - Local students have advantage (company graduate programs)
  - This course = Step 1, not final destination

Homework:

- Create "About Me" HTML page with proper structure
- Complete 10 W3Schools HTML exercises (specific list provided)
- GitHub setup and first commit
- **Tracking**: Checkbox in shared spreadsheet when completed

---

SESSION 2: HTML Structure & Semantics

Main Topics:

- Semantic HTML deep dive (header, nav, main, section, article, aside, footer)
- Links, images, lists
- HTML validation and accessibility basics

- File organization

Learning Objectives:

- Write semantic, accessible HTML
- Organize project files properly
- Understand HTML document structure fully

Resources:

- **Pre-session**: Read MDN "How browsers load websites"
- **In-session**: W3C Validator demo, WebAIM accessibility guide
- **Post-session**: "Semantic HTML" (Kevin Powell, YouTube)

Hands-on Practice:

- W3Schools exercises: Semantic elements, links, images
- Round-robin problem solving (10 exercises, rotate participants)
- Build portfolio structure: header, nav, main sections, footer

Career/Professional (10 min):

- **QA/Testing Roles**: How HTML knowledge helps testers
- **Web Designers**: HTML as design implementation
- **Tools Introduction**: Browser DevTools basics
- **EPAM Mentoring Preview**: What programs exist, entry requirements

Homework:

- Complete portfolio HTML structure (all sections)
- 15 W3Schools exercises
- Run HTML through validator, fix errors
- Add to tracking spreadsheet

SESSION 3: HTML Forms & Advanced Elements

Main Topics:

- Form elements (we'll add handlers in JS phase)
- Input types and attributes
- Tables for data
- Embedded content (video, iframe basics)

Learning Objectives:

- Create complete forms (functionality comes later)
- Use appropriate input types
- Structure tabular data
- Prepare for JavaScript integration

Resources:

- **Pre-session**: W3Schools Forms Tutorial
- **In-session**: MDN “Images, media, and form elements”
- **Post-session**: "HTML Forms" tutorial (freeCodeCamp)

Hands-on Practice:

- W3Schools form exercises (round-robin)
- Create contact form (no validation yet)
- Build skills/experience table
- Everyone completes 3 different input-type demonstrations

Career/Professional (10 min):

- **Backend Development**: How backend devs work with HTML forms
- **Cloud Services Introduction**: What is cloud? AWS/Azure/GCP basics
- **Why you'll hear about these**: Modern apps live in the cloud

Homework:

- Add contact form and tables to portfolio
- Complete 10 form-related W3Schools exercises
- **Preparation**: Think about color scheme and design for CSS phase
- Update tracking spreadsheet

PHASE 2: CSS Mastery (Sessions 4-8)

SESSION 4: CSS Fundamentals & Color Theory

Main Topics:

- CSS syntax, selectors, linking stylesheets
- Color theory basics and color palettes
- Typography fundamentals
- CSS cascade and specificity

Learning Objectives:

- Write clean CSS
- Choose cohesive color schemes
- Apply professional typography
- Understand how CSS rules apply

Resources:

- **Pre-session**:
  - Explore coolors.co, choose 3-5 colors
  - Read: "CSS Basics" (MDN)
- **In-session**:
  - Google Fonts demo
  - Coolors.co walkthrough
- **Post-session**: "CSS Tutorial for Beginners" (Net Ninja)

Hands-on Practice:

- W3Schools CSS exercises (selectors, colors, fonts) - round-robin
- Each participant shares their chosen color palette
- Apply basic styling to portfolio: colors and fonts

Career/Professional (10 min):

- **Design Tools**: Figma, Canva, Adobe XD overview
- **Why designers matter**: Collaboration between designers and developers
- **Theming concept**: How companies maintain brand consistency

Homework:

- Create complete color palette (primary, secondary, accent, neutral, success/error)
- Document in CSS variables
- Apply typography system to portfolio
- 15 CSS exercises on W3Schools
- Update tracking

SESSION 5: CSS Box Model & Layout Basics

Main Topics:

- Box model (margin, padding, border)
- Display properties
- Positioning (static, relative, absolute, fixed)
- Box-sizing

Learning Objectives:

- Master spacing and positioning
- Control element layout
- Debug spacing issues with DevTools

Resources:

- **Pre-session**: "CSS Box Model" (W3Schools)
- **In-session**: DevTools spacing visualization
- **Post-session**: "Learn CSS Box Model in 8 Minutes" (Web Dev Simplified)

Hands-on Practice:

- W3Schools box model exercises (round-robin)
- DevTools exploration: inspect and modify live sites
- Apply proper spacing to portfolio

Career/Professional (10 min):

- **DevOps Overview**: What is DevOps? CI/CD basics
- **Deployment concept**: How websites go from code to live
- **GitHub Pages**: We'll deploy here soon

Homework:

- Perfect portfolio spacing (consistent margins/padding)
- Style buttons and form elements
- 12 positioning exercises
- Update tracking

SESSION 6: Flexbox & Modern Layouts

Main Topics:

- Flexbox concepts and mental model
- Flex container vs items
- Common patterns: navigation, cards, centering
- Responsive layout foundations

Learning Objectives:

- Build flexible layouts
- Create navigation bars and card grids
- Center elements efficiently

Resources:

- **Pre-session**: "Flexbox Froggy" (game, complete all levels)
- **In-session**: CSS Tricks "Complete Guide to Flexbox"
- **Post-session**: "Flexbox Tutorial" (Traversy Media)

Hands-on Practice:

- Flexbox Froggy together (project on screen, participants guide)
- W3Schools Flexbox exercises (round-robin)
- Rebuild portfolio layout with Flexbox

Career/Professional (10 min):

- **Project Management Tools**: Jira, Trello, Asana deep dive
- **Agile/Scrum basics**: How tech teams work
- **Sprints, standups, retrospectives**

Homework:

- Convert portfolio to Flexbox layout
- Complete remaining Flexbox Froggy levels
- 10 Flexbox exercises
- Update tracking

SESSION 7: Responsive Design & Grid

Main Topics:

- Mobile-first philosophy
- Media queries and breakpoints
- CSS Grid basics
- Responsive images and units

Learning Objectives:

- Make portfolio fully responsive
- Use Grid for complex layouts
- Understand viewport and relative units

Resources:

- **Pre-session**:
  - "Grid Garden" (game)
  - Read: "Responsive Web Design Basics" (web.dev)
- **In-session**: DevTools device mode demo
- **Post-session**: "CSS Grid Tutorial" (Wes Bos, free)

Hands-on Practice:

- Grid Garden together (screen sharing)
- Test portfolio on different devices (DevTools)
- W3Schools responsive exercises (round-robin)
- Make portfolio mobile-friendly

Career/Professional (10 min):

- **Databases Overview**: What are databases? SQL vs NoSQL
- **How frontend connects to backend**: APIs concept
- **Data Engineer/Data Scientist roles**: Why data matters

Homework:

- Fully responsive portfolio (mobile, tablet, desktop)
- Screenshot at 3 sizes for documentation
- Complete Grid Garden
- 10 responsive design exercises
- Deploy to GitHub Pages
- Update tracking

SESSION 8: CSS Advanced & Pre-processors

Main Topics:

- Transitions and animations
- Transforms and shadows
- CSS variables (custom properties)
- Introduction to SASS/LESS
- CSS architecture basics (BEM methodology)

Learning Objectives:

- Add polish with animations
- Understand CSS pre-processors
- Write maintainable CSS
- Prepare for production-level styling

Resources:

- **Pre-session**:
  - "CSS Animations" (Kevin Powell, YouTube)
  - SASS documentation quickstart
- **In-session**:

    - SASS compiler demo (Live Sass Compiler in VS Code)
    - BEM naming examples
- **Post-session**: "SASS Tutorial" (freeCodeCamp)

Hands-on Practice:

- Add animations to portfolio (hover effects, transitions)
- W3Schools animation exercises (round-robin)
- Convert CSS to SASS (optional, depends on time)
- Refactor with BEM naming

Career/Professional (10 min):

- **Security Roles Overview**: What is cybersecurity in tech?
- **Why security matters**: HTTPS, data protection, authentication
- **Common security concepts**: Passwords, tokens, encryption basics

Homework:

- Polish portfolio with animations and effects
- Use CSS variables throughout
- Optional: Convert to SASS
- 10 advanced CSS exercises
- **Milestone**: Portfolio should look professional now
- Update tracking

PHASE 3: JavaScript Fundamentals (Sessions 9-13)

SESSION 9: JavaScript Basics & DOM Introduction

Main Topics:

- JavaScript role in web development
- Variables, data types, operators
- Console and debugging
- Introduction to DOM
- Selecting and manipulating elements

Learning Objectives:

- Write basic JavaScript
- Select HTML elements
- Change content and styles dynamically
- Debug with console

Resources:

- **Pre-session**:
  - "JavaScript for Beginners" (freeCodeCamp, first 30 min - or article)
  - W3Schools JS Tutorial (Variables, Data Types)
- **In-session**: DevTools Console exploration
- **Post-session**: "JavaScript DOM Crash Course" (Traversy Media)

Hands-on Practice:

- W3Schools JS exercises: variables, operators (round-robin)
- LeetCode Easy problems: simple algorithms (group solving)
- DOM manipulation: change portfolio heading, colors on button click

Career/Professional (10 min):

- **AI in Development**: ChatGPT, Claude, GitHub Copilot demo
- **How I use AI daily**: Code suggestions, debugging, learning
- **AI as assistant, not replacement**: You must understand the code

Homework:

- Add JavaScript file to portfolio
- Dynamic date in footer
- Simple greeting based on time of day
- 15 W3Schools JS exercises
- 2 LeetCode Easy problems
- Update tracking

SESSION 10: Functions & Events

Main Topics:

- Functions: declaration, parameters, return
- Arrow functions
- Event listeners and handlers
- Form events
- Preventing default behavior

Learning Objectives:

- Write reusable functions
- Handle user interactions
- Control form behavior
- Create interactive features

Resources:

- **Pre-session**:
  - "JavaScript Functions" (W3Schools)
  - "JavaScript Guide" (MDN)
- **In-session**: Event debugging in DevTools
- **Post-session**: "JavaScript Events Tutorial" (Net Ninja)

Hands-on Practice:

- W3Schools functions and events exercises (round-robin)
- LeetCode problems requiring functions (group solving)
- Add interactivity to portfolio:
  - Button clicks
  - Form submission handling
  - Dark mode toggle

Career/Professional (10 min):

- **Collaboration Tools Deep Dive**:
  - Slack: channels, threads, integrations, bots
  - Teams: meetings, file sharing
  - Why async communication matters in remote work

Homework:

- Add interactive features to portfolio:
  - Dark mode toggle (with localStorage)
  - Form validation
  - Smooth scroll navigation
  - "Scroll to top" button
- 12 W3Schools JS exercises
- 3 LeetCode Easy problems
- Update tracking

SESSION 11: Arrays, Loops & Data Manipulation

Main Topics:

- Arrays and array methods (map, filter, forEach, reduce)
- Loops (for, while)
- Iterating over data
- Generating HTML from arrays
- Template literals

Learning Objectives:

- Work with collections efficiently
- Use modern array methods
- Generate dynamic content
- Transform data

Resources:

- **Pre-session**:
  - "JavaScript Arrays" (W3Schools)
  - "Array Methods" (JavaScript.info)
- **In-session**: Array methods visualization
- **Post-session**: "8 Must Know JavaScript Array Methods" (Web Dev Simplified)

Hands-on Practice:

- W3Schools array exercises (round-robin)
- LeetCode array problems (group solving)
- Refactor portfolio:
  - Skills from array
  - Projects from array
  - Dynamic generation

Career/Professional (10 min):

- **Version Control Deep Dive**:
  - Git workflows: feature branches, PR process
  - GitHub: Issues, Projects, Wikis
  - CI/CD basics: GitHub Actions introduction

Homework:

- Convert portfolio data to arrays of objects
- Generate sections dynamically
- Add filter/sort functionality
- 10 array-focused exercises
- 3 LeetCode array problems
- Update tracking

SESSION 12: Objects, JSON & localStorage

Main Topics:

- JavaScript objects

- JSON format
- localStorage API
- Data persistence
- Structuring application data

Learning Objectives:

- Work with structured data
- Store and retrieve data
- Persist user preferences
- Prepare for API integration

Resources:

- **Pre-session**:
  - "JavaScript Objects" (W3Schools)
  - "Working with JSON" (MDN)
- **In-session**: localStorage DevTools inspection
- **Post-session**: "JavaScript localStorage" (JavaScript.info)

Hands-on Practice:

- W3Schools objects exercises (round-robin)
- LeetCode object manipulation problems
- Portfolio data restructuring:
  - Central data object
  - localStorage for preferences
  - Theme persistence

Career/Professional (10 min):

- **AWS/Azure/GCP Services**:
  - S3/Blob Storage: where websites live
  - Databases: RDS, CosmosDB
  - Serverless: Lambda functions basics
- **Why this matters**: Modern apps are cloud-native

Homework:

- Create comprehensive portfolio data structure (JSON)
- Implement localStorage for all settings
- Generate entire portfolio from data object
- 10 object/JSON exercises
- 2 LeetCode problems
- Update tracking

SESSION 13: Async JavaScript & Fetch API Basics

Main Topics:

- Synchronous vs asynchronous
- Callbacks basics
- Promises introduction
- async/await basics
- Fetch API for data retrieval
- Working with APIs (very basic)

Learning Objectives:

- Understand asynchronous JavaScript
- Make HTTP requests
- Handle API responses
- Integrate external data (preparation for React)

Resources:

- **Pre-session**:
  - "Asynchronous JavaScript" (W3Schools)
  - "Fetch API" (MDN)
- **In-session**:
  - Network tab in DevTools
  - Public API demo (weather, quotes, etc.)
- **Post-session**: "Async JavaScript Crash Course" (Traversy Media)

Hands-on Practice:

- Fetch data from public API (JSONPlaceholder, GitHub API, etc.)
- Display fetched data in portfolio
- W3Schools async exercises (round-robin)
- Error handling practice

Career/Professional (10 min):

- **LinkedIn Strategy Part 1**:
  - Profile optimization: photo, headline, about
  - How recruiters search
  - Keywords that matter
- **Action**: Update LinkedIn profile today

Homework:

- Integrate one API into portfolio (e.g., GitHub repos, quotes)
- Handle loading and error states

- 8 async JavaScript exercises
- 2 API-related LeetCode problems
- **LinkedIn**: Update profile with new skills
- Update tracking

PHASE 4: React & Modern Development (Sessions 14-20)

SESSION 14: Introduction to React & Components

Main Topics:

- Why frameworks? What problem does React solve?
- Component-based architecture
- JSX syntax
- Setting up React (Vite)
- First component

Learning Objectives:

- Understand React philosophy
- Set up React project
- Create functional components
- Write JSX

Resources:

- **Pre-session**:
  - "React in 100 Seconds" (Fireship)
  - React official docs: Quick Start
- **In-session**: React setup walkthrough
- **Post-session**: "React Tutorial for Beginners" (Programming with Mosh)

Hands-on Practice:

- Set up React project together
- Create first components (Header, Footer)
- W3Schools React exercises (if available) or React docs tutorial
- Convert one portfolio section to React component

Career/Professional (10 min):

- **LinkedIn Strategy Part 2**:
  - Experience section: framing non-tech as tech-relevant
  - Projects section: showcase your portfolio

    - Skills endorsements
- **Action**: Post about starting React

Homework:

- Set up personal React portfolio project
- Convert 2-3 sections to components
- Complete React docs tutorial (Tic-Tac-Toe)
- **Reading**: React concepts (components, props, state)
- Update tracking

SESSION 15: Props, State & Event Handling in React

Main Topics:

- Props: passing data to components
- State with useState hook
- Event handling in React
- Conditional rendering
- Lists and keys

Learning Objectives:

- Pass data between components
- Manage component state
- Handle user interactions
- Render dynamic content

Resources:

- **Pre-session**: React docs: "Thinking in React"
- **In-session**: useState hook deep dive
- **Post-session**: "React Hooks Tutorial" (Codevolution)

Hands-on Practice:

- Create reusable components (Button, Card, Input)
- Build interactive features (dark mode in React)
- Map over data to generate components
- Exercises: Build mini-components (counter, toggle, form)

Career/Professional (10 min):

- **LinkedIn Strategy Part 3**:
    - Networking: who to connect with

    - Messaging templates
    - Engagement strategy (liking, commenting)
- **Communities**:
    - Django Girls, Women Who Code, PyLadies
    - Local Budapest tech meetups
    - How to find and join

Homework:

- Convert more portfolio sections to React components
- Implement state management (theme, filters, etc.)
- Build 2 practice components (e.g., todo list, counter)
- Join 2 women in tech communities
- Connect with 10 people on LinkedIn
- Update tracking

SESSION 16: React Hooks & Side Effects

Main Topics:

- useEffect hook
- Component lifecycle
- Fetching data in React
- Custom hooks (basic)
- Performance considerations

Learning Objectives:

- Manage side effects
- Fetch and display data
- Understand component lifecycle
- Write cleaner code with custom hooks

Resources:

- **Pre-session**: React docs: "useEffect"
- **In-session**: useEffect examples and common patterns
- **Post-session**: "useEffect Explained" (Fullstack Simplified)

Hands-on Practice:

- Fetch data with useEffect
- Implement search/filter functionality
- Create custom hook (e.g., useLocalStorage)
- Debug infinite loops (common useEffect mistake)

Career/Professional (10 min):

- **Job Search Reality Check**:
  - Why applying now is premature
  - Hungarian/EU junior market challenges
  - Advantage of local graduates
- **Better Path**:
  - EPAM mentoring programs (entry requirements, application process)
  - Other bootcamps and courses
  - Building portfolio projects
  - Contributing to open source

Homework:

- Integrate API data into React portfolio
- Implement custom hooks
- Complete advanced React tutorial
- **Research**: EPAM mentorship program, save application for later
- Update tracking

SESSION 17: React Router & Multi-Page Apps

Main Topics:

- React Router setup
- Navigation and links
- Route parameters
- Nested routes
- Programmatic navigation

Learning Objectives:

- Create multi-page React apps
- Handle navigation
- Pass data through routes
- Build professional app structure

Resources:

- **Pre-session**: React Router docs: Tutorial
- **In-session**: Router setup and patterns
- **Post-session**: "React Router Tutorial" (Net Ninja)

Hands-on Practice:

- Add routing to portfolio:
    - Home, About, Projects, Contact pages
    - Project detail pages with route params
    - Navigation component
- Build navigation menu with active states

Career/Professional (10 min):

- **Diagramming & Design Tools**:
    - Miro: collaborative whiteboarding
    - Figma: design and prototyping
    - Draw.io: flowcharts and diagrams
    - Mermaid: diagrams as code
- **Why these matter**: Communication and planning

Homework:

- Complete React Router integration
- Create project detail pages
- Add 404 page
- Practice: Build a small routed app (e.g., blog)
- Update tracking

SESSION 18: React Context API & State Management

Main Topics:

- Props drilling problem
- Context API
- useContext hook
- Global state management
- When to use Context vs props

Learning Objectives:

- Manage global application state
- Avoid props drilling
- Structure larger React apps
- Prepare for advanced state management (Redux mention)

Resources:

- **Pre-session**: React docs: "Context"
- **In-session**: Context setup and patterns
- **Post-session**: "React Context & Hooks Tutorial" (Net Ninja)

Hands-on Practice:

- Create theme context for dark mode
- User preferences context
- Auth context (simulated)
- Refactor portfolio to use Context where appropriate

Career/Professional (10 min):

- **Continuous Learning Paths**:
  - After this course: recommended next steps
  - Free resources: freeCodeCamp, Scrimba, Codecademy
  - YouTube channels: Traversy Media, Net Ninja, Web Dev Simplified
  - Books: "Eloquent JavaScript", "You Don't Know JS"
- **Building in Public**: Documenting learning on LinkedIn/Twitter

Homework:

- Implement Context API in portfolio
- Build practice app using Context (e.g., shopping cart)
- Create learning roadmap for next 3 months
- Start documenting learning journey on LinkedIn (weekly posts)
- Update tracking

SESSION 19: React Best Practices & Optimization

Main Topics:

- Component organization
- Code splitting and lazy loading
- useMemo and useCallback
- React DevTools
- Error boundaries
- Deployment preparation

Learning Objectives:

- Write production-ready React code
- Optimize performance
- Debug React applications
- Prepare for deployment

Resources:

- **Pre-session**: React docs: "Thinking in React", "Best Practices"

- **In-session**: React DevTools deep dive, Performance profiling
- **Post-session**: "React Performance Optimization"

Hands-on Practice:

- Refactor portfolio for optimization
- Code splitting implementation
- React DevTools exploration
- Performance audit and fixes
- Prepare for deployment (Vercel/Netlify)

Career/Professional (10 min):

- **Next Steps After This Course**:
  - Not ready for job applications yet (realistic expectation)
  - **Immediate**: Continue building projects (3-4 more)
  - **3 months**: Apply to EPAM or similar mentorship
  - **6 months**: Start selective job applications
  - **Ongoing**: Networking, community involvement, skill building
- **What employers expect from juniors**: Portfolio of 5+ projects, contributions, continuous learning

Homework:

- Optimize portfolio completely
- Deploy React portfolio (Vercel or Netlify)
- Build one more React project (idea: weather app, movie search, etc.)
- **Prepare**: Presentation for final session
- Update tracking

SESSION 20: Final Presentations & Career Roadmap

Structure:

🕐 0-45 min: Portfolio Presentations

- Each participant: 4-5 minutes
  - Live demo of React portfolio
  - Show one feature they're proud of
  - Explain one challenge overcome
  - Share their 3-month learning plan
- Peer feedback and encouragement

🕐 45-50 min: Break & Photos

🕐 **50-75 min: Your Career Roadmap (The Real Talk)**

Immediate Next Steps (Weeks 1-4):

- ✅ Deploy both portfolios (vanilla JS and React)
- ✅ Build 2 more projects (ideas provided)
- ✅ Daily: 1 hour coding practice
- ✅ Weekly: 2-3 LinkedIn posts about learning
- ✅ Join local meetups (1-2 per month)

3-Month Plan:

- ✅ Complete advanced course (recommendations provided)
- ✅ Apply to EPAM mentorship or similar programs
- ✅ Build 3-5 more projects showcasing variety
- ✅ Contribute to open source (beginner-friendly projects listed)
- ✅ Expand network: 50+ connections in tech
- ✅ Attend tech conferences/events in Budapest

6-Month Plan:

- ✅ Specialize: Choose frontend/backend/full-stack
- ✅ Learn advanced topics: Testing, TypeScript, Next.js, etc.
- ✅ Portfolio: 8-10 quality projects
- ✅ NOW consider selective job applications
- ✅ Or continue with structured mentorship until ready

The Reality of Job Hunting:

- 🎯 **Junior market is tough**, especially for non-local graduates
- 🎯 **Don't rush applications**: Premature applications = rejections = demotivation
- 🎯 **Build experience first**: Projects, contributions, community involvement
- 🎯 **Network matters**: Many jobs filled through connections, not applications
- 🎯 **Patience and consistency win**: 6-12 months of focused learning is normal

Alternative Paths to Employment:

- Internships (unpaid/paid)
- Freelance projects (Upwork, Fiverr for small tasks)
- Contributing to open source (builds resume)
- Volunteering tech skills for NGOs
- Teaching others (reinforces learning)

EPAM & Similar Programs:

- Entry requirements (basic HTML/CSS/JS - ✅ you have this now)
- Application process and timeline

- What they teach (advanced topics)
- Employment pathway after completion
- Higher success rate than cold applications

Resources Handout:

- Learning platforms (free & paid)
- YouTube channels
- Books and documentation
- Practice platforms (LeetCode, Codewars, HackerRank)
- Communities (Slack, Discord, Reddit)
- Local events and meetups in Budapest
- Job boards (when ready): LinkedIn, AngelList, local Hungarian sites

🕐 75-85 min: Q&A & Ongoing Support

- Any questions about career, tech, learning
- Post-course support:
    - Monthly office hours (if offered)
    - Slack/Email always open
    - Code review requests welcome
    - LinkedIn connections maintained

🕐 85-90 min: Closing & Celebration

- **What you've accomplished**:
    - HTML/CSS/JS foundations ✅
    - React basics ✅
    - Git & GitHub ✅
    - Two portfolio projects ✅
    - Understanding of IT landscape ✅
    - Roadmap for continued growth ✅
- **You are developers now** - not job-ready yet, but on the path
- **This is the beginning**, not the end
- **Stay consistent, stay connected, stay learning**
- Group photo, certificates (if provided)
- Exchange contacts
- Celebration! 🎉

Post-Course "Assignment" (No deadline, ongoing):

- Execute your 3-month plan
- Share wins in Slack (projects completed, courses finished, connections made)
- Help each other (peer support)
- When someone gets their first tech job: we all celebrate!
