



განათ
ლება



FULL STACK დეველოპმენტი ხელოვნური ინტელექტის დახმარებით

კურსის სილაბუსი

სასწავლო კურსის
სტატუსი

საბაზისო სასერტიფიკატო კურსი BSCTBC110

შეხვედრებისა
და საათების
რაოდენობა

ხანგრძლივობა: 3 თვე
შეხვედრების რაოდენობა: 20 ლექცია
საღამოს საათები: 40

ლექტორი

ნოდარ კიზირია
ელფოსტა: nodar.kiziria@geolab.edu.ge

გიორგი ასანიძე
ელფოსტა: giorgi.asanidze@geolab.edu.ge

ანი ჩარექიშვილი
ელფოსტა: ani.charekishvili@geolab.edu.ge

სასწავლო კურსის
ფორმატი

თეორიული და პრაქტიკული ლექციები მოიცავს ინტერაქციულ
ლექციებს, ინდივიდუალურ და ჯგუფურ მუშაობას, დისკუსიებსა და
პრაქტიკულ სამუშაოებს.

კურსი დაყოფილია 4 ბლოკად. თითოეული ბლოკი მოიცავს ერთ
შუალედურ პროექტს. კურსის ბოლოს იქნება დამოუკიდებელი სამუშაო
პერიოდი ფინალური პროექტისთვის მოსამზადებლად

#	სასწავლო კურსის შინაარსი	ძირითადი მიზნები
ბლოკი 1	FRONT END-ის საფუძვლები და AI-POWERED დიზაინი	
ლექცია 1 (2 საათი)	ვების არქიტექტურა და სემანტიკური სტრუქტურა • თემები: როგორ მუშაობს ინტერნეტი, ბრაუზერი და სერვერი. HTML5-ის სემანტიკური თეგები. VS Code და პირველი პროექტის სტრუქტურა. • კონსტრუქტივისტული აქტივობა: Live Coding -ყველამ ერთად ავანცოთ მარტივი პორტფოლიოს ტიპის ვებგვერდის ჩონჩხი. • AI სტრატეგია: მოსწავლეები იყენებენ Google Gemini-ს სემანტიკური თეგების (მაგ: <main>, <article>, <aside>) მნიშვნელობის ასახსნელად და საკუთარი კონსპექტების შესადარებლად.	ესმის ვების არქიტექტურა და HTML5 სემანტიკური თეგები; შეუძლია AI-ის (Gemini) გამოყენებით სტრუქტურის გაანალიზება და ვებგვერდის მყარი, გამართული ჩონჩხის აწყობა.
ლექცია 2 (2 საათი)	სტილები და თანამედროვე CSS ჩარჩოები (Tailwind CSS) • თემები: CSS-ის მუშაობის პრინციპი. Tailwind CSS-ის ინტეგრაცია. ფერები, ფონტები (Google Fonts), ზომები, ბალები (Flexbox). DevTools-ის (Inspect element) გამოყენება დებაგინგისთვის. • კონსტრუქტივისტული აქტივობა: CSS Inspector Challenge - ლექტორი უჩვენებს „გაფუჭებულ“ ვებგვერდს, მოსწავლეებმა DevTools-ით უნდა იპოვონ პრობლემა და გამოასწორონ. • AI სტრატეგია: Google Stitch - დიზაინის იდეების გენერირება და Tailwind-ის კლასების სტრუქტურის ახსნა.	ფლობს CSS-ის მუშაობის პრინციპებსა და Tailwind CSS-ს; შეუძლია DevTools-ით ხარვეზების აღმოჩენა (დებაგინგი) და სტილების მოქნილად/სწრაფად აწყობა კოდში.
ლექცია 3 (2 საათი)	რესპონსიული დიზაინი და მობილურზე მორგება (Mobile-First) • თემები: რესპონსიული დიზაინის პრინციპები, Breakpoints, მედია ქუერი (Media Queries) Tailwind-ში. SVG და ვებგამოყენებადობის (UX/UI) საფუძვლები. Git და GitHub-ის საფუძვლები. • კონსტრუქტივისტული აქტივობა: Adaptive Challenge - მოსწავლეები გარდაქმნიან თავიანთ პორტფოლიოს ისე, რომ იდეალურად ჩანდეს როგორც მობილურ ტელეფონზე, ისე დიდ მონიტორზე. შესრულებული სამუშაოს Git-ზე ატვირთვა. • AI სტრატეგია: AI ასისტენტს ვთხოვთ შემოგვთავაზოს მობილური ნავიგაციის (Mobile Hamburger Menu) საუკეთესო პრაქტიკა და დაგვიჩვენოს მისი სუფთა სტრუქტურა.	იყენებს Mobile-First მიდგომასა და Breakpoints-ს; შეუძლია ვებგვერდის რესპონსიული ადაპტაცია სხვადასხვა ეკრანისთვის და კოდის Git/GitHub-ზე ატვირთვა.

ლექცია 4 (2 საათი)	შუალედური პროექტი 1 - „Responsive Portfolio Launch“ (შეტრიალებული საკლასო ოთახი) • დავალება: მოსწავლეებმა GitHub Pages-ზე უნდა განათავსონ საკუთარი, სრულად რესპონსიული პორტფოლიო, სადაც წარმოდგენილი იქნება მათი იდეები და მომავალი პროექტების გეგმა. • ფორმატი: მოსწავლეები თავად გამოდიან პრეზენტაციით, ხსნიან თავიანთი დიზაინის გადაწყვეტილებებს. • Peer Review: სხვა მოსწავლეები შუალედური პროექტის წარმდგენს გაუზიარებენ უკუკავშირს. • AI როლი: პრეზენტაციის სტრუქტურის დაგეგმვა ხელოვნური ინტელექტის დახმარებით.	წარადგენს GitHub Pages-ზე განთავსებულ სრულად რესპონსიულ პორტფოლიოს, ასახულებს დიზაინერულ გადაწყვეტილებებს და იღებს/ამუშავებს Peer Review უკუკავშირს.
ბლოკი 2	ინტერაქტიული ინტერფეისი და REACT-ის ჯაფოსნობა	
ლექცია 5 (2 საათი)	JavaScript-ის საფუძვლები და დინამიკური ვები • თემები: ცვლადები, ფუნქციები, მასივები და ობიექტები JS-ში. DOM მანიპულაციები, ივენტები (EventListeners). ასინქრონული JavaScript (fetch, async/await) და API-დან მონაცემების წამოღება. • კონსტრუქტივისტული აქტივობა: API Hunt — უფასო საჭარო API-დან (https://jsonplaceholder.typicode.com/) მონაცემების წამოღება და გვერდზე დინამიურად გამოსახვა. • AI სტრატეგია: JS ლოგიკის შეცდომების დებაგინგი ხელოვნური ინტელექტის დახმარებით. კოდის „ადამიანურ ენაზე“ თარგმნა (Code Explainer).	JavaScript-ის საბაზისო კონცეფციებს (DOM, Events, Async/Fetch); შეუძლია API-დან მონაცემების წამოღება, გვერდის გაცოცხლება და AI-ით კოდის დებაგინგი.
ლექცია 6 (2 საათი)	შესავალი React დეველოპმენტში • თემები: რატომ React? Virtual DOM-ის კონცეფცია. JSX სინტაქსი. კომპონენტების შექმნა და ერთმანეთში ჩაშენება. Props - მონაცემების გადაცემა კომპონენტებს შორის. • კონსტრუქტივისტული აქტივობა: React Lego - აპლიკაციის ინტერფეისის დაშლა პატარა, დამოუკიდებელ კომპონენტებად (ბარათები, ლილაკები, ნავიგაცია) და მათი ხელახალი აწყობა. • AI სტრატეგია: Antigravity - კომპონენტების ავტომატური გენერირება და სუფთა კოდის სტრუქტურის შექმნა.	ესმის Virtual DOM-ისა და JSX-ის ლოგიკა; შეუძლია ინტერფეისის დაშლა ხელახლა გამოყენებად React კომპონენტებად და Props-ის მეშვეობით მონაცემთა გადაცემა.

ლექცია 7 (2 საათი)	State-ის მართვა და Hooks (useState, useEffect) • თემები: კომპონენტის შიდა მდგომარეობა (useState). ივენთების მართვა React-ში. გვერდის სიცოცხლის ციკლი და ეფექტები (useEffect). ფორმებთან მუშაობა React-ში. • კონსტრუქტივისტული აქტივობა: Interactive Dashboard - „ჭკვიანი“ ტასქ-მენეჯერის (To-Do App) აწყობა, რომელშიც შესაძლებელია დავალებების დამატება. • AI სტრატეგია: ხელოვნურ ინტელექტთან დისკუსია თემაზე: „როდის გამოვიყენოთ Local State და როდის Props?“. შეცდომების პრევენცია state-ის განახლებისას.	მართავს აპლიკაციის შიდა მდგომარეობას useState და useEffect Hooks-ით; შეუძლია ინტერაქტიული, რეალურ დროში მოქმედი React აპლიკაციის (მაგ. Task Manager) აწყობა.
ლექცია 8 (2 საათი)	შუალედური პროექტი 2 - „Interactive React App“ (შეტრიალებული საკლასო ოთახი) • დავალება: ინტერაქტიული, მრავალკომპონენტიანი React აპლიკაცია, რომელიც იყენებს საჯარო API-ს და მართავს სხვადასხვა მდგომარეობას (States). • ფორმატი: მოსწავლეები აზიარებენ ეკრანს, ახორციელებენ თავიანთი კოდის დემონსტრირებას და პასუხობენ თანაკურსელების შეკითხვებს. • AI როლი: კოდის რეფაქტორინგი და ოპტიმიზაცია ხელოვნური ინტელექტის მიერ პროექტის ჩაბარებამდე.	წარადგენს ინტერაქტიულ, მრავალკომპონენტიან React აპლიკაციას საჯარო API-ის ინტეგრაციითა და გამართული State Management-ით.
ბლოკი 3	BACK END და სერვერული ლოგიკა (NODE.JS & EXPRESS)	
ლექცია 9 (2 საათი)	შესავალი Node.js-სა და Express-ში • თემები: რა არის Back End? Node.js გარემო და npm (პაკეტების მენეჯერი). Express.js ფრეიმვორკი. სერვერის გაშვება ლოკალურად. მარშრუტიზაციის (Routing) საფუძვლები. • კონსტრუქტივისტული აქტივობა: Our First Server - საკუთარი ლოკალური სერვერის შექმნა, რომელიც პასუხობს სხვადასხვა მისამართზე (მაგ. /hello, /api/data). • AI სტრატეგია: API არქიტექტურის დაგეგმვა და Express სერვერის საწყისი შაბლონის (Boilerplate) გენერირება ხელოვნური ინტელექტის დახმარებით.	ესმის Back End-ის არქიტექტურა; შეუძლია Node.js/Express-ის გარემოში ლოკალური სერვერის გაშვება, მარშრუტიზაციის (Routing) გამართვა და AI-ით Boilerplate-ის დაგეგმვა.

ლექცია 10 (2 საათი)	REST API და CRUD ოპერაციები • თემები: RESTful API-ს პრინციპები. HTTP მეთოდები: GET, POST, PUT, DELETE. Request Body, Query Parameters და Path Variables. Middleware-ის კონცეფცია. • კონსტრუქტივისტული აქტივობა: API Builder - მოსწავლეები აწყობენ იმიტირებულ მონაცემთა ბაზას (In-Memory Array) და წერენ სრულ CRUD ლოგიკას დავალებების სამართავად. • AI სტრატეგია: API „ენდპოინტების“ ტესტირება Thunder Client-ით (Antigravity IDE Extension) და ხელოვნური ინტელექტის მიერ დაწერილი ავტომატური ტესტებით.	ფლობს RESTful API-ის პრინციპებსა და HTTP მეთოდებს; ანყოფნა სრულ CRUD ლოგიკას მონაცემთა სამართავად და ამონებს ენდპოინტებს Thunder Client-ით.
ლექცია 11 (2 საათი)	მონაცემთა ბაზები და Mongoose • თემები: რელაციური vs არარელაციური ბაზები. MongoDB-ს გაცნობა (Atlas Cloud). Mongoose ბიბლიოთეკა Node.js-ში. სქემების (Schemas) და მოდელების (Models) შექმნა. მონაცემების შენახვა, წაკითხვა, განახლება და წაშლა ბაზაში. • კონსტრუქტივისტული აქტივობა: Database Mapping - რეალური სქემის შექმნა იუზერებისა და პროდუქტებისთვის. ბაზასთან დაკავშირება და პირველი ჩანაწერების გაკეთება. • AI სტრატეგია: MongoDB სქემის შექმნა და მონაცემთა ვალიდაციის წესების შემუშავება AI-ს გამოყენებით.	ერკვევა NoSQL ბაზების ლოგიკაში (MongoDB); შეუძლია Mongoose-ის სქემებისა და მოდელების შექმნა მონაცემთა სტრუქტურირებული შენახვისა და ვალიდაციისთვის
ლექცია 12 (2 საათი)	შუალედური პროექტი 3 - „Own REST API & DB“ (შეტრიალებული საკლასო ოთახი) • დავალება: დამოუკიდებელი, ფუნქციონალური REST API, რომელიც დაკავშირებულია MongoDB-სთან და აწარმოებს სრულფასოვან CRUD ოპერაციებს. • ფორმატი: მოსწავლეები წარმოადგენენ API-ს დოკუმენტაციას და აჩვენებენ, თუ როგორ მუშაობს მონაცემთა ბაზა რეალურ დროში. • AI როლი: შეცდომების დამუშავების (Error Handling) ლოგიკის გაუმჯობესება ხელოვნურ ინტელექტზე დაფუძნებული ასისტენტის დახმარებით.	ქმნის და წარადგენს MongoDB-სთან დაკავშირებულ დამოუკიდებელ, ფუნქციონალურ REST API-ს გამართული Error Handling ლოგიკით.
ბლოკი 4	სრული ინტეგრაცია (FULL STACK) და ხელოვნური ინტელექტზე დაფუძნებული API-ს ჩაშენება	

ლექცია 13
(2 საათი)

Front End-ისა და Back End-ის კავშირი

- თემები: CORS პოლიტიკა და მისი მოგვარება. Axios ბიბლიოთეკა React-ში. სერვერიდან მონაცემების წამოღება და React State-ში ასახვა. გარემოს ცვლადები (.env).
- კონსტრუქტივისტული აქტივობა: Bridge the Gap - მოსწავლეები აერთიანებენ თავიანთი React UI-ს და Node.js სერვერს. ვებგვერდზე გამოჩნდება ბაზიდან რეალურ დროში წამოღებული მონაცემები.
- AI სტრატეგია: ქსელური შეცდომების (CORS, 500 Internal Server Error) დიაგნოსტიკა და გადაჭრა ხელოვნურ ინტელექტთან ერთად.

აკავშირებს React UI-ს Node.js სერვერთან Axios-ით;

მართავს CORS პოლიტიკასა და გარემოს ცვლადებს (.env) მონაცემთა უწყვეტი მიმოცვლისთვის.

ლექცია 14
(2 საათი)

უსაფრთხოება და ავტორიზაცია (JWT)

- თემები: როგორ მუშაობს ავტორიზაცია ვებში. პაროლების ჰეშირება (bcrypt). JSON Web Tokens (JWT) კონცეფცია. დაცული მარშრუტები (Protected Routes) React-სა და Express-ში.
- კონსტრუქტივისტული აქტივობა: Secret Gate - რეგისტრაციისა და ლოგინის გვერდის აწყობა. მხოლოდ ავტორიზებული მომხმარებლებისთვის განკუთვნილი პანელის შექმნა.
- AI სტრატეგია: ავტორიზაციის უსაფრთხოების აუდიტი. ხელოვნურ ინტელექტს ვთხოვთ, შეამოწმოს ჩვენი JWT კოდი პოტენციურ ხარვეზებზე.

იყენებს პაროლების ჰეშირებას (bcrypt) და JSON Web Tokens (JWT);

ქმნის ავტორიზაციის სისტემასა და დაცულ მარშრუტებს (Protected Routes).

ლექცია 15
(2 საათი)

ხელოვნური ინტელექტის (Gemini API) ინტეგრაცია კოდში

- თემები: როგორ ჩავრთოთ ხელოვნური ინტელექტი ჩვენს აპლიკაციაში. API გასაღებების მართვა. Gemini API SDK-ს გამოყენება Node.js-ში. Prompt Engineering კოდის დონეზე (System Instructions-ის გაწერა).
- კონსტრუქტივისტული აქტივობა: AI Assistant inside my App - აპლიკაციაში ჩაშენებული მინი-ჩატის ან ჭკვიანი გენერატორის შექმნა (მაგალითად: მომხმარებლის მიერ შეყვანილი პროდუქტების მიხედვით რეცეპტების გენერატორი).
- AI სტრატეგია: Prompt-ების ოპტიმიზაცია სასურველი ფორმატის (მაგ. სუფთა JSON პასუხის) მისაღებად.

ახდენს Gemini API-ის ინტეგრაციას Node.js აპლიკაციაში;

მართავს API გასაღებებსა და Prompt Engineering-ს კოდის დონეზე პერსონალიზებული ფუნქციონალის შესაქმნელად.

ლექცია 16 (2 საათი)	შუალედური პროექტი 4 - „AI-Powered Full Stack App“ (შეტრიალებული საკლასო ოთახი) • დავალება: მუშა Full Stack აპლიკაცია, სადაც მომხმარებელს შეუძლია დარეგისტრირდეს, შეინახოს მონაცემები ბაზაში და მიიღოს პერსონალიზებული პასუხები ჩაშენებული ხელოვნური ინტელექტისაგან. • ფორმატი: ინტერაქტიული პრეზენტაცია, სადაც მოსწავლეები გვაჩვენებენ სრულ მომხმარებლის გზას (User Flow). • AI როლი: დიზაინის დახვეწა Framer Motion ანიმაციებით, რომელთა კოდსაც ხელოვნური ინტელექტის დახმარებით დანერგენ.	წარადგენს სრულყოფილ Full Stack აპლიკაციას ავტორიზაციით, მონაცემთა ბაზითა და ჩაშენებული AI ასისტენტით/ფუნქციონალით.
ბლოკი 5	ფინალური პროექტის ლაბორატორია და მომავლის ხედვა	
ლექცია 17 (2 საათი)	ხელოვნური ინტელექტის (Gemini API) ინტეგრაცია კოდში • თემები: პროექტის დაგეგმვა: სქემები, DB დიაგრამები, API endpoints-ის რუკა. Git branches, Pull Requests და გუნდური მუშაობის სიმულაცია. დეპლოიშენტი: React-ის გაშვება Vercel-ზე, Node.js სერვერის - Render-ზე. • კონსტრუქტივისტული აქტივობა: Push to Production - მოსწავლეები თავიანთ პროექტს ტვირთავენ ღრუბელში და აზიარებენ პირველ საჯარო ბმულებს. • AI სტრატეგია: დეპლოიშენტის შეცდომების (Build errors) ანალიზი და მოგვარება ხელოვნურ ინტელექტზე დაფუძნებული ასისტენტის დახმარებით.	გეგმავს პროექტის არქიტექტურას და Git-ის ტოტებს; ახდენს React Front End-ის (Vercel) და Node.js Back End-ის (Render) დეპლოიშენტს ღრუბელში.
ლექცია 18 (2 საათი)	პრეზენტაციის ხელოვნება (Pitching) და მომავლის ტექნოლოგიები • თემები: როგორ წარვადგინოთ პროექტი დამკვეთთან ან ინვესტორთან (Pitching). Future Literacy: დისკუსია ვებტექნოლოგიების მომავალზე (Web3, AI Agents, No-Code/Low-Code-ის გავლენა დეველოპერებზე). • კონსტრუქტივისტული აქტივობა: Elevator Pitch Simulation - თითოეულ მოსწავლეს აქვს 1 წუთი, რათა დაარწმუნოს კლასი საკუთარი აპლიკაციის სარგებლიანობაში. • AI როლი: საპრეზენტაციო სლაიდების სტრუქტურის შექმნა ხელოვნური ინტელექტის დახმარებით.	ატარებს UX/Bug Bounty ტესტირებას, პოულობს სისტემურ თუ ვიზუალურ ხარვეზებს და ახდენს კოდის რეფაქტორინგსა და ოპტიმიზაციას AI-ის დახმარებით.

ლექცია 19 (2 საათი)	პრეზენტაციის ხელოვნება (Pitching) და მომავლის ტექნოლოგიები • თემები: როგორ წარვადგინოთ პროექტი დამკვეთთან ან ინვესტორთან (Pitching). Future Literacy: დისკუსია ვებტექნოლოგიების მომავალზე (Web3, AI Agents, No-Code/Low-Code-ის გავლენა დეველოპერებზე). • კონსტრუქტივისტული აქტივობა: Elevator Pitch Simulation - თითოეულ მოსწავლეს აქვს 1 წუთი, რათა დაარწმუნოს კლასი საკუთარი აპლიკაციის სარგებლიანობაში. • AI სტრატეგია: საპრეზენტაციო სლაიდების სტრუქტურის შექმნა ხელოვნური ინტელექტის დახმარებით.	ფლობს 1-წუთიანი Elevator Pitch-ისა და პროექტის წარდგენის ტექნიკას; ესმის ვებტექნოლოგიების მომავლის ტენდენციები (Web3, AI Agents).
ლექცია 20 (2 საათი)	პროექტების განხილვა/პრეზენტაცია და ლექტორთან უკუკავშირი	