Junior Developer

პროგრამის ოფიციალური დანართი

კოდი	JD
ხანგრძლივობა	4–6 თვე (შესაძლებელია მოქნილი გრაფიკის ინდივიდუალურად შეთანხმება ბენეფიციარებისთვის).
სასწავლო დრო	სულ: 520 აკადემიური საათი (თეორია 200, პრაქტიკა 260, დამოუკიდებელი სწავლა 30, საბოლოო შეფასება 30).

მიმოხილვა

პროგრამა «Junior Developer» წარმოადგენს სტრუქტურირებულ პროფესიულ მომზადებას, რომელიც შეესაბამება დარგობრივ მოთხოვნებსა და რეალური სამუშაო გარემოს პრაქტიკას. პროგრამა ორიენტირებულია პრაქტიკული პროფესიული უნარების განვითარებაზე, სამუშაო პასუხისმგებლობის სწორად შესრულებასა და უსაფრთხოების მოთხოვნების დაცვაზე. სასწავლო გეგმა ორგანიზებულია თანმიმდევრული სასწავლო ბლოკების სახით, სწავლების შედეგების ეტაპობრივი მიღწევით, ზედამხედველობით პრაქტიკული დავალებებით და ფორმალური შეფასებით. ამ გვერდზე წარმოდგენილი ინფორმაცია წარმოადგენს სერტიფიკატის ოფიციალურ დანართს და გამოიყენება გარე და საერთაშორისო გადამოწმებისთვის.

ვისთვის არის პროგრამა

- პირები, რომლებიც ისწრაფვიან ოფიციალური პროფესიული კვალიფიკაციისა და დოკუმენტურად დადასტურებული კომპეტენციების მიღებისკენ — პროგრამული უზრუნველყოფის განვითარების საფუძვლები: პროგრამირება, ვებ-ტექნოლოგიები, მონაცემთა ბაზები და გუნდური მუშაობა-ის მიმართულებით.
- პრაქტიკული გამოცდილების მქონე პირები, რომლებიც სურთ უნარების სისტემატიზაცია, ცოდნის ხარვეზების შევსება და კომპეტენციების დადასტურება სტრუქტურირებული სწავლებისა და შეფასების გზით.
- კომპანიებისა და მომსახურების მიმწოდებლების თანამშრომლები, რომლებსაც სჭირდებათ კვალიფიკაციის დოკუმენტური დადასტურება კარიერული განვითარების, ტენდერებში მონაწილეობის ან შესაბამისობის მოთხოვნებისათვის.

წინაპირობები

წერა-კითხვისა და დათვლის საბაზისო უნარები. წინასწარი გამოცდილება სავალდებულო არ არის; მნიშვნელოვანია პრაქტიკულ სწავლაზე მზადყოფნა და

უსაფრთხოების წესების დაცვა.

სწავლების ფორმატი

კომპეტენციებზე დაფუძნებული სწავლება: თეორია და ზედამხედველობით პრაქტიკა (დაახლ. 50%).

სწავლის შედეგები

- შრომის უსაფრთხოების მოთხოვნების დაცვა და PPE-ის სწორად გამოყენება
- ტექნიკური დოკუმენტაციის (ნახაზები/სპეციფიკაციები) გაგება და პროცედურების დაცვა
- ხელსაწყოების/მასალების/აღჭურვილობის სწორად შერჩევა და უსაფრთხო მდგომარეობაში შენარჩუნება
- ძირითადი სამუშაო ოპერაციების შესრულება ხარისხის სტანდარტებისა და ტოლერანსების დაცვით
- გავრცელებული გაუმართაობებისა/დეფექტების ამოცნობა და კორექტირების შესრულება
- სამუშაო დოკუმენტაციის წარმოება და ეფექტური კომუნიკაცია ხელმძღვანელთან/კლიენტთან
- მცირე აპლიკაციების შექმნა, ტესტირება და დოკუმენტირება თანამედროვე ინსტრუმენტებითა და ვერსიების კონტროლით

შეფასება

შეფასება მოიცავს ქვიზებს, პრაქტიკულ დავალებებს, კოდის მიმოხილვას და საბოლოო პროექტს პრეზენტაციით.

გაცემული დოკუმენტები

- პროგრამის დასრულების სერტიფიკატი
- პროგრამის ციფრული დანართი
- საბოლოო პროექტის შეფასების ფურცელი

სასწავლო გეგმა

ბლოკი	თემები

კვირები 1–2 (40 სთ): საფუძვლები და დეველოპერის გარემო	• პროგრამის გაცნობა, სასწავლო პროცესი და დეველოპერული აზროვნება • დეველოპერის გარემოს დაყენება: Node.js, Git, IDE • ვების საფუძვლები: HTTP, ბრაუზერები, HTML სემანტიკა, ხელმისაწვდომობა • კომანდური ხაზი და Git სამუშაო პროცესი • CSS საფუძვლები და რესპონსიული დიზაინი (Tailwind CSS) • JavaScript საფუძვლები: ტიპები, ფუნქციები, მასივები, ობიექტები • მინი პროექტი: სტატიკური ლენდინგ გვერდი + GitHub რეპოზიტორია
კვირები 3–4 (40 სთ): ფრონტენდის არქიტექტურა	• TypeScript-ის საფუძვლები: ტიპები, ინტერფეისები, generics • Next.js App Router: როუტინგი, ლეიაუთები, მეტადატა • კომპონენტების სტრუქტურა და UI კომპოზიცია • Client და Server კომპონენტები • მონაცემების მიღება და შეცდომების დამუშავება • ფორმები და ვალიდაცია • მინი პროექტი: მრავალგვერდიანი ლოკალიზებული ვებსაიტი
კვირები 5–6 (40 სთ): სტეიტი, API და უსაფრთხოების საფუძვლები	• React-ის სტეიტ მენეჯმენტი და hooks • ხელახალი გამოყენებადი კომპონენტები და UI ბიბლიოთეკები (shadcn/ui) • ხელმისაწვდომობა და UX კონსისტენტურობა • REST API, JSON და HTTP სტატუს კოდები • ავთენტიკაციის საფუძვლები (cookies, სესიები) • მინი პროექტი: დაცული გვერდი (დემო)
კვირები 7–8 (40 სთ): ბექენდი და მონაცემთა ბაზები	• MongoDB-ის საფუძვლები და სქემების დიზაინი (Mongoose) • CRUD ოპერაციები Next.js API-ით • ინპუტ ვალიდაცია და შეცდომების პასუხები • ძიება, ფილტრაცია და პაგინაცია • ფაილების ატვირთვა და სურათების ოპტიმიზაცია • მინი პროექტი: CRUD ადმინ პანელი
კვირები 9–10 (40 სთ): უსაფრთხოება, წარმადობა და დებაგინგი	• ვების უსაფრთხოების საფუძვლები: OWASP, CSRF, XSS • როლებზე დაფუძნებული წვდომა • წარმადობა: ქეშირება, ინდექსები • ლოგირება და დებაგინგი • მინი პროექტის გამაგრება: უსაფრთხო API

კვირები 11–12 (40 სთ): დოკუმენტაცია და გუნდური მუშაობა	• README და API დოკუმენტაცია • კოდის სტილი, ლინტინგი და ფორმატირება • Git ბრენჩები, pull request და code review • დეპლოის საფუძვლები და env ცვლადები • მინი პროექტი: დეპლოი და რელიზ ნოუთები
კვირები 13–26 (280 სთ): პროფესიული პრაქტიკა და საბოლოო პროექტი	• პროფესიული პრაქტიკა რეალურ ამოცანებზე • ფუნქციონალის შექმნა user stories-ის მიხედვით • ბაგების გამოსწორება და რეფაქტორინგი • ყოველკვირეული code review და მენტორის უკუკავშირი • საბოლოო პროექტის განხორციელება • პრეზენტაცია და პორტფოლიოს მომზადება