Boston Coding Bootcamp

Full-Stack Web Developer

Michael Klishevich info@busation.com www.codingbootcamp.ru

Basic Course Information

Bootcamp starts: October 23, 2017
Bootcamp ends: April 23, 2018
Form of study: online webinars using Zoom
Schedule of study: two times a week (Tuesday, Thursday) from 7pm to 9pm lecture, one time a week (Saturday) from 10am to 12am seminary (Moscow time).
Home work: after each lecture home task is given, so plan than you will need at least 4 hours of home work each week.
Cost of study: $200 a month.
Conditions of enrolment: interview
Ideal student: able to read English, should have some understanding of programming, good if have some experience with HTML/CSS and any programming language, although possible for students without any prior experience, but with technical mentality. Age above 16 years.
Purpose of course: to give enough knowledge to get job as a junior web-developer. Help to find a job will be given, though no guarantee to get a job is given.
Language of course: English and Russian
Notice: those with poor academic performance could be expelled without compensation.

Enrolment

Drop me email info@busation.com
or Telegram me @klishevich
or Twitter me @busation
to schedule interview

Program

Welcome to Programming (1 week)

1. Introduction to Programming **(2 hours)**
   - Review of Programming Specialisations: **web developer (frontend, backend, full-stack)**, mobile (iOS, Android), SQL (MS SQL, PL SQL, Postgres), Data Science (Machine Learning, Data Mining, Cluster Analysis), iOS, VR development, Games development
   - Review of Programming Languages and Frameworks: Java, C#, Python, **Ruby on Rails**, Swift, Objective C, **JavaScript**, Angular, **React**, Vue, jQuery
   - **Home task:** install ruby, play around with IRB
2. How to find first programming job **(0,5 hour)**
3. Version Control: **Git (GitHub)**, SVN, TFS **(1 hour)**
   - **Home task:** Setting GitHub account, add SSH keys and creating first repository
4. Text Editors and IDEs: **Sublime Text**, Visual Studio Code, Atom, WebStorm, RubyMine
   - **Home task**: Set up Sublime Text editor, add JavaScript and Ruby autocomplete plugins.
5. Linux or MacBook basics **(1 hour)**
   - Terminal basic commands
   - VI basic commands
   - **Home task:** Create basic HTML index page using Terminal and VI

6. Tools **(0,5 hour)**
   - Slack
   - Jira
   - Git
   - **Home task:** Register Slack Account and join the Bootcamp team

CSS / HTML / JavaScript (2 - 3 week)

7. Basics of CSS / HTML **(2 hours**)
   - **Home task:** Create your personal page with your bio, photo, email, phone. Using as much css and html tags as you can (https://fvsch.com/code/base-stylesheet/full.css)
8. Basics of JavaScript + jQuery **(2 hours)**
   - **Home task:** Add some JavaScript to you page: alerts, form to calculate your precise age, manipulations with DOM
9. Responsive sites with **Twitter Bootstrap** / Material UI / Foundation **(2 hours)**
   - **Home task:** Create you Bio page using bootstrap template. Page should be responsive, use grid system, tables, navigation, at least two pages, carousel
10. Setting up Linux server **(2 hours)**
    - **Home task:** Register Linode account, acquire domain name, set up nginx and add site to hosting.

Ruby on Rails (4 - 12 week)

11. About Ruby, about Rails, about object oriented programming **(1 hour)**.
    - IRB
    - Basic OOP concepts
    - **Home task**: interactive console ruby application. And application using OOP approach.
12. Ruby on Rails (RoR) project structure **(2 hours)**.
    - MVC programming pattern. MVC parts: Models, Views, Controllers, Routing.
    - Gems, Bundler.
    - Static Ruby on Rails site.
    - **Home task:** make personal two page site with header and footer.
13. Deployment application tool - Capistrano **(1 hour)**
    - **Home task:** Set up Capistrano for your project
14. SQL Basics **(2 hours)**
    - Install and configurate Postgres (Linux, OSX)
    - Set up Rails database config, create model, seed data, basic SQL commands.
    - Theory or relational databases
    - **Home task:** Create two tables, with index and foreign key. Scripts to fill data in tables, to select data, to group data, to update data, to delete.
15. Rails Assets **(2 hours)**
    - CSS, JavaScript, Images, Fonts
    - Add Bootstrap to project
    - **Home task**: add custom template to project
16. Authentication with Devise gem **(1 hour)**
    - **Home task:** Create site with registration and user page
17. Authorization with CanCan **(1 hour)**
    - **Home task:** Admin user to list all users.
18. Work with images and files **(2 hours)**
    - Set up image magic
    - Add Images and files to your project.
19. Rails console **(1 hour)**
    - **Home task**: create data through rails console, work with models, select data
20. Mailers интеграция с Gmail **(2 hours)**
    - **Home task**: Set up mailers and templates to your project
21. Background jobs: Redis, Resque **(2 hours).**
    - Resque-web

- God
- **Home task**: Create model Messages, create UI for admin to create message. Set up weekly news email for all users send with resque.

22. AWS S3 file storage and about other AWS services **(2 hours)**
    - **Home task**: Set up AWS account. Set up AWS S3 storage for your project
23. Integration with REST APIs. With Twitter example **(2 hours)**
    - **Home task:** Integrate your project with Twitter, Facebook
24. Rspec-testing **(2 hours)**
    - Unit tests
    - Request tests
    - Testing JavaScript
    - **Home task:** write tests for your project
25. Developing your ruby library - Gem **(2 hours)**
    - **Home task:** Create your personal library
26. Developing REST API **(4 hours)**
    - API template, API version
    - CORS settings
    - Authentication (Gems, JWT for server)
    - Serializers
    - Tests
    - Services
    - Queries
    - **Home task:** create REST API for model singers
27. Personal project **(4 hours)**
    - **Project types:** E-Commerce site or Integration module or Social network
    - **Project should have:** custom UI design, at least one integration, at least one background job.
28. Team project 3 members in each team **(4 hours)**
    - **Project type:** REST API for education system. Models: Teacher, Student, Exam, ExamResult.
    - **Project should have:** custom UI design, at least two integrations, at least two background job.

Node.js and React (13 - 21 week)

29. What is Node.js **(2 hours)**
    - Install NPM, node
    - EcmaScript 6+
    - node REPL
    - Webpack
    - React alternatives: Angular 2, Vue, Meteor, Ember
    - **Home task:** create new node.js project play around with node.
30. Integrate React in RoR Project **(2 hours)**
    - Setting up Webpack for Rails project
    - ES6 modules, import, export
    - Dynamic page with React example.
    - **Home task:** Editable Grid page with React (integrated in Rails app): Webpack, ES6+
31. react-rails / react_on_rails projects **(2 hours)**
    - react-rails example
    - react_on_rails example
    - Home task: Make demo project with two models and server side rendering with one of projects.
32. Redux **(3 hours)**
    - Why to use Redux
    - Redux alternatives
    - Dynamic page with Redux example
    - Redux Thunk Middleware: Redux Thunk
    - **Home task:** Editable Grid page with React and Redux:
33. Single Page Application SPA **(2 hours)**

    - Create new JavaScript project add necessary dependancies Webpack, React, Redux, React Router.
    - **Home task:** Create personal project with Webpack, React, Redux, React Router and couple of pages.

34. Using create-react-app for new project **(2 hours)**
    - Project structure
    - Start up scripts
    - Adding Redux and React Router
    - Grid Page
    - **Home task:** Continue what was done at lecture - Create page and Edit page and routing
35. What it takes to be Webpack Engineer **(2 hours)**
    - Assets, Loaders, npm scripts
    - dev and productions configurations
    - start with mocks
    - **Home task:** Set up webpack from scratch
36. Productivity: ESLint, mocks with nock **(1 hour)**
    - **Home task:** Set up personal eslintrc.js, play around with different code styles
37. Testing Application: Jest, Karma **(3 hours)**
    - testing functions
    - testing components
    - integration tests
    - **Home task:** full coverage for your project
38. Authentication for SPA **(2 hours)**
    - Understanding of authentication and authorisation
    - JWT authentication
    - **Home task:** create new project and add JWT authentication with existing Rails backend
39. Node.js Express server **(4 hours)**
    - Purpose of node.js server
    - Add services and mocks to node.js server
    - Adding business logic to node.js server
    - Using authentication on server
    - **Home task:** create application with Express server connected to Postgres database and CRUD UI.
40. Google Firebase **(2 hours)**
    - What is Firebase and when to use it
    - Using Firebase as backend for SPA
    - **Home task:** replace Postgres with Firebase
41. Setting Up Production server **(2 hours)**:
    - Nginx, Express.
    - **Home task:** set up production environment for your node.js project with assets.
42. Continuous deployment of node.js application with Jenkins / Travis CI **(2 hours)**
    - **Home task:** deploy your node.js application with Jenkins
43. Personal project **(4 hours)**
44. Team project **(4 hours)**

Exams

After course knowledge will be tested, and education certificate will be given with assistance to get the first job.