



React

The Complete React Course
[85 Hrs]

Instructor: Pondit Team

E-Mail:

Phone:

Overview

আইটি ইন্ডাস্ট্রির জন্য দক্ষ জনশক্তি তৈরিতে পন্ডিট স্পন্সরশীপ প্রদান করছে। যার পরিপ্রেক্ষিতে ইউনিভার্সিটির ছাত্রছাত্রীরা একটি পরীক্ষায় ভালো রেজাল্টের মাধ্যমে ১০০% পর্যন্ত ফ্রি কোর্স করার সুযোগ পাচ্ছেন।

পরীক্ষার মাধ্যমে আপনি ১০০% পর্যন্ত স্পন্সরশীপ পাবেন তবে আপনাকে অবশ্যই পরীক্ষায় ৮০% মার্ক্স পেতে হবে। ১০০% স্পন্সরশীপ না পেলে পরীক্ষার রেজাল্টের উপর ভিত্তি করে ৭৫% অথবা ৫০% ডিসকাউন্ট পাবেন।

বর্তমান বিশ্বে জাভাস্ক্রিপ্ট একটা জনপ্রিয় স্ক্রিপ্টিং ল্যাঙ্গুয়েজ। এর চাহিদা এবং জনপ্রিয়তা এতটাই যে ২০১৭ ডেভেলপার জরিপে এই ল্যাঙ্গুয়েজটি শীর্ষস্থান অর্জন করে। যদি আপনি ক্যারিয়ার হিসেবে জাভাস্ক্রিপ্টকে বেছে নেন তাহলে আপনার কোনদিন কাজের অভাব হবে না। ভালো সম্মানী ছাড়াও আপনি যদি কোন ফার্ম এ জব না করতে চান, আপনার জন্য রয়েছে অনলাইন মার্কেটপ্লেস।

আর React হচ্ছে ফেইসবুক দ্বারা develop করা frontend এ ব্যবহৃত একটি জাভাস্ক্রিপ্ট লাইব্রেরী। বর্তমানে React ডেভেলপার এর চাহিদা প্রচুর। Single Page Application কিংবা সংক্ষেপে SPA তৈরী করতে React ব্যবহার করা হয়। আর React জানলে, React Native -এ কাজ করা সহজ। React Native দিয়ে Android এবং iOS এর জন্য মোবাইল এপ্লিকেশন তৈরী করা যায়।

এটি দিয়ে তৈরী কিছু এপ্লিকেশনঃ

□ সোশ্যাল নেটওয়ার্কিং অ্যাপ্লিকেশন

- ফেসবুক
- ইনস্টাগ্রাম
- প্রিন্টারেস্ট
- টুইটার
- রেডিট

□ শেয়ারিং প্ল্যাটফর্ম

- এয়ারবিএনবি (Airbnb)
- লিফ্ট (Lyft)
- উবার

Prerequisites

প্রোগ্রামিং নলেজ থাকলে ভালো

Course Information

- Duration : 85 Hours
- Day : রবিবার এবং বৃহস্পতিবার
- Time : রাত ৯.৩০টা থেকে ১১.৩০টা পর্যন্ত
- Reg. Start : Jun 22, 2022
- Reg. End : Aug 3, 2022
- Class Start : Aug 4, 2022

□ মিডিয়া সাইটস

- ইয়াহ! (ভাদের মেইল ক্লায়েন্ট রিয়েক্ট দিয়ে ডেভেলপ করা হয়েছে)
- নিউইয়র্ক টাইমস

□ ভিডিও প্ল্যাটফর্ম

- নেটফ্লিক্স

□ SaaS (সফটওয়্যার এজ এ সার্ভিস)

- ড্রপবক্স
- সেন্ডগ্রিড
- আসানা
- ইনভিশনঅ্যাপ
- জ্যাপিয়ার

□ অন্যান্য বিশেষ

- মাইক্রোসফট
- ইবে (Ebay)

Audience

- শিক্ষার্থী (যারা সফটওয়্যার ডেভেলপমেন্টে ক্যারিয়ার গড়তে চান)
- চাকুরীজীবী

Evaluation

Curriculum

1. Basic JavaScript

- 1.1. 1. Introduction: Javascript introduction
- 1.2. a. Job field
- 1.3. b. Tools and technologies in javascript
- 1.4. c. Installing necessary tools
- 1.5. d. Variables
- 1.6. e. Constants
- 1.7. f. Primitive types
- 1.8. g. Objects
- 1.9. h. Arrays
- 1.10. i. Functions
- 1.11. j. Loop
- 1.12. k. Operators
- 1.13. l. Control flow
- 1.14. m. Factory functions
- 1.15. n. Constructor functions
- 1.16. o. Value vs reference types
- 1.17. p. Cloning an object
- 1.18. q. Garbage collections
- 1.19. r. Math

- 1.20. s. String
- 1.21. t. Template literals
- 1.22. u. Map
- 1.23. v. Reduce
- 1.24. w. forEach
- 1.25. x. Filter
- 1.26. y. Elements or objects finding
- 1.27. z. Prototype chaining
- 1.28. 2. Advanced Topics: a. Execution context
- 1.29. i. Memory component
- 1.30. ii. Code component
- 1.31. b. Execution context phases
- 1.32. i. Creation phase
- 1.33. ii. Execution phase
- 1.34. c. Call stack
- 1.35. d. Hoisting
- 1.36. e. Scope
- 1.37. f. Lexical environment
- 1.38. g. Scope chain
- 1.39. h. Temporal dead zone
- 1.40. i. Block
- 1.41. j. Shadowing
- 1.42. k. Closure
- 1.43. l. Browser
- 1.44. m. Callback queue or task queue
- 1.45. n. Event loop
- 1.46. o. Microtask queue
- 1.47. p. Microtasks
- 1.48. q. Starvation of callback functions
- 1.49. 3. ES6 features: a. Let vs const
- 1.50. b. Objects
- 1.51. c. This keyword
- 1.52. d. Binding this
- 1.53. e. Arrow function
- 1.54. f. Arrow function and this
- 1.55. g. Spread operator
- 1.56. h. Classes
- 1.57. i. Inheritance
- 1.58. j. Modules
- 1.59. k. Named and defaults exports
- 1.60. 4. Extra topics: a. Promise
- 1.61. b. Promise chaining
- 1.62. c. Async/await
- 1.63. d. Currying
- 1.64. e. Modules export import
- 1.65. c. First react app d. Hello World

2. React.js

- 2.1. 1. Introduction:
- 2.2. a. What is react?
- 2.3. b. Setting up the development environment

2.4. c. First react app
2.5. d. Hello World
2.6. e. Custom configs
2.7. f. Full stack architecture
2.8. 2. Components:
2.9. a. Setting up the project
2.10. b. First react component
2.11. c. Specifying children
2.12. d. Rendering classes dynamically
2.13. e. Rendering Lists
2.14. f. Conditional rendering
2.15. g. Handling events
2.16. h. Binding event handlers
2.17. i. Updating the state
2.18. j. What happens when state changes
2.19. k. Passing event arguments
2.20. l. Project – set up a project
2.21. m. Building a products component
2.22. n. Deleting a product
2.23. 3. Composing components:
2.24. a. Composing components
2.25. b. Passing data to components
2.26. c. Passing children
2.27. d. Debugging the react apps
2.28. e. Props vs state
2.29. f. Raising and handling Events
2.30. g. Updating the state
2.31. h. Single source of truth
2.32. i. Removing the local state
2.33. j. Multiple components in sync
2.34. k. Lifting the state up
2.35. l. Stateless functional component
2.36. m. Destructuring Arguments
2.37. n. Lifecycle Hooks
2.38. o. Mounting phase
2.39. p. Updating phase
2.40. q. Unmounting phase
2.41. r. Exercise – Decrement button
2.42. s. Exercise – facebook like button
2.43. 4. Pagination, Filtering, Sorting:
2.44. a. Pagination - Pagination component
2.45. b. Pagination – Component Interface
2.46. c. Pagination – Displaying pages
2.47. d. Pagination – Handling page changes
2.48. e. Pagination – paginating data
2.49. f. Pagination – Type checking with prototypes
2.50. g. Exercise – List group component
2.51. h. Filtering - Component Interface
2.52. i. Filtering – Displaying items
2.53. j. Filtering – Default props
2.54. k. Filtering – Handling sections

2.55. l. Filtering – implementing Filtering
2.56. m. Filtering – Adding all products
2.57. n. Sorting – Extracting products table
2.58. o. Sorting – Raising the sort event
2.59. p. Sorting – implementation sorting
2.60. q. Sorting – Extracting table header
2.61. r. Sorting – Extracting table body
2.62. s. Sorting – redering cell content
2.63. t. Sorting – Unique keys
2.64. u. Sorting – adding the sort icons
2.65. v. Sorting – Extracting the table
2.66. w. Sorting – Extracting a method
2.67. 5. Routing:
2.68. a. Setup
2.69. b. Adding routing
2.70. c. Using Switch
2.71. d. Link
2.72. e. Route props
2.73. f. Passing props
2.74. g. Route parameters
2.75. h. Optional parameters
2.76. i. Query string parameters
2.77. j. Redirects
2.78. k. Programmatic Navigation
2.79. l. Nested Routing
2.80. m. Exercise – Navbar and Routing
2.81. n. Adding react router
2.82. o. Adding routes
2.83. p. Adding the navbar
2.84. q. Linking to the ProductForm
2.85. 6. Forms:
2.86. a. Building a bootstrap form
2.87. b. Handling bootstrap form
2.88. c. Handling form submission
2.89. d. Refs
2.90. e. Controlled elements
2.91. f. Handling multiple elements
2.92. g. Common errors
2.93. h. Extracting a reusable inputs
2.94. i. Validations
2.95. j. Displaying validation error
2.96. k. Validation onChange
2.97. l. Joi or Yup
2.98. m. Validating a form using joi or yup
2.99. n. Disabling the submit button
2.100. o. Extracting a reusable form
2.101. p. Extracting helper rendering methods
2.102. q. Register form
2.103. r. Exercise – Product form
2.104. s. Exercise – search products
2.105. 7. Calling backend service:

- 2.106. a. Http clients or axios
- 2.107. b. Getting data
- 2.108. c. Lifecycle of a request
- 2.109. d. Updating data
- 2.110. e. Deleting data
- 2.111. f. Optimistic vs pessimistic updates
- 2.112. g. Expected vs unexpected errors
- 2.113. h. Extracting a reusable Http service
- 2.114. i. Extracting a config module
- 2.115. j. Displaying toast notifications
- 2.116. k. Saving the product
- 2.117. 8. Authentication and authorization:
- 2.118. a. Registering a new user
- 2.119. b. Submitting the registration form
- 2.120. c. Handling registration errors
- 2.121. d. Logging in a user
- 2.122. e. Submitting the login form
- 2.123. f. Handling login errors
- 2.124. g. Storing the jwt
- 2.125. h. Logging the user upon registration
- 2.126. i. JSON WEB Token(jwt)
- 2.127. j. Getting the current user
- 2.128. k. Displaying the current user on navbar
- 2.129. l. Logging out a user
- 2.130. m. Authorization
- 2.131. n. Showing or hiding elements based on the user
- 2.132. o. Protecting routes
- 2.133. p. Extracting protected routes
- 2.134. q. Redirecting after log out
- 2.135. 9. Deployment:
- 2.136. a. Environment variables
- 2.137. b. Production builds
- 2.138. c. Getting started with Heroku
- 2.139. d. Deploying to the Heroku
- 2.140. e. Setting environment variables on Heroku
- 2.141. f. Preparing the front-end for deployment
- 2.142. g. Deploying the front-end
- 2.143. 10. Advanced topics:
- 2.144. a. Setting up the development environment
- 2.145. b. Higher order components
- 2.146. c. Hooks
- 2.147. d. useState hook
- 2.148. e. useEffect hook
- 2.149. f. custom hook
- 2.150. g. Fetching data with hooks
- 2.151. h. Context
- 2.152. i. Formik basic with implementation
- 2.153. j. Redux idea (not implementation. Need another course for it)