

בוטקאמפ הסבת אקדמאים בפיתוח Full Stack בטכנולוגיות ואפליקציות WEB

מספר קורס 4575 - 590 שעות לימוד

רקע:

הפיתוח המסורתי בסביבת ה-WEB (אינטרנט) נחלק שניים:

1. **פיתוח צד לקוח** – אחראי, בין היתר, על ממשק המשתמש.
2. **פיתוח צד שרת** – מתרכז בחלק הלוגי, הנמצא "מאחורי" ממשק המשתמש.

הפיתוח המסורתי התאפיין בשני תפקידים מסורתיים – מפתחים המתמחים בצד-לקוח ומפתחים המתמחים בצד שרת. מפתח Front-End ומפתח Back-End. חלוקה זו אילצה לפזר את הידע על פני מפתחים שונים בארגון וגרמה לבעיות תקשורת ותגובות איטיות.

לאורך השנים, הכלים העומדים לרשות מפתחים השתפרו והטכנולוגיות מאפשרות לייצר פתרונות מורכבים במהירות יחסית. הדבר נכון הן לצד-שרת והן לצד-לקוח. אותם פתרונות וסביבות יעילים, הובילו לכך שבשנים האחרונות מסתמנת מגמה של "איחוי" הפיתוח של שני הצדדים – לקוח ושרת, לכדי תפקיד אחד: **Full Stack Developers**. מפתחי **Full Stack** שולטים במגוון אדיר של יכולות לצד הלקוח, ומאפשרים בניית מסכים אינטראקטיביים והצגה דינאמית וחכמה של מידע ותוכן מצד אחד, ומצד שני, הבנה עמוקה ויכולת פיתוח של תהליכים הנדרשים לעבד את הנתונים בצד השרת ולהנגישם בצורה יעילה ומסודרת לצד הלקוח.

עצם העובדה שניתן לרכז את כל הידע, מקצה (לקוח) לקצה (שרת), באדם אחד, אשר מבין את המורכבות שבשני הצדדים, אך גם מכיר את הפתרונות לכל צד, מובילה לשיפור בתהליך הפיתוח בארגון. לכן, הדרישה ל-Full Stack Developers גדלה מאוד בתקופה האחרונה וצפויה רק להמשיך ולגדול.

ביכולתו של צוות מפתחים, בעלי יכולות גבוהות למימוש והכלה של הטכנולוגיות העדכניות בעולם ההייטק, למצוא את הפתרון היעיל ביותר ולהתאימו לכל משתמשי הקצה הקיימים ואף לאלו העתידים לבוא. העובדה שכיום ניתן לקבל מענה שכזה, בעזרת מפתח Full Stack הופכת אותו לאטרקטיבי ביותר. ככל שעוד ועוד פלטפורמות, ארגונים, שירותים ובילויים נעים לכיוון האינטרנט, כך יוסיפו הטכנולוגיות להשתפר ולהתייעל, ואיתן גם הצורך במפתחים בעלי יכולת לראות ולחבר את התמונה כולה בזמן אמת.

בוטקאמפ (Bootcamp) בפיתוח Full Stack הינו תכנית הכשרה אינטנסיבית ומעמיקה עבור בעלי תארים אקדמיים אשר מטרתה להכשיר את הבוגרים להיות מפתחי תוכנה מקצה לקצה בסביבת Web ולהפוך ל-Full Stack Developer מן השורה הראשונה. הקורס מועבר בדגש על למידה יישומית Hands-on המשלב בסיס תיאורטי חזק עם ביצוע משימות ופרויקטים המדמים עבודה בחברות.

הקורס מאפשר לבוגריו לפתח Rich Internet Applications ולהפוך ל-Web Developer תוך שימוש במילה האחרונה של טכנולוגיות צד-לקוח כגון Angular ו-React. בנוסף לכך אופן הלימוד מספק ראייה רוחבית בצד השרת וזאת ע"י התמקדות בטכנולוגית צד שרת משמעותית ופופולארית Node.js.

יתרונות קורס

- מאפשר למפתחים להיקלט במגוון חברות החל ביזמות פרטית וחברות הזנק וכלה במוסדיים.
- מכין את הבוגרים למקצועות המבוקשים והחמים בעולם פיתוח ה-WEB כיום.
- מספק הבנה מעמיקה של הארכיטקטורות והתהליכים עליהם נשענות הטכנולוגיות בקורס.
- מתרכז ביישום ובפרקטיקה על מנת להקנות ניסיון מעשי תוך כדי הלמידה.
- מאפשר לבוגריו לתת מענה מקצה לקצה ולשמש כמפתחי Full Stack המבוקשים כיום מאוד בקרב חברות גדולות וקטנות.

קהל יעד

- כל המעוניינים להתמקצע בתחום האינטרנט ותכנות בעולם ה-Web.
- מועמדים ללא רקע או עם רקע בסיסי בתכנות ל-Web.
- בעלי תואר ראשון או הנדסאים מכל תחום.
- למעוניינים בהבנה מעמיקה של Front End Development לבניית צד לקוח מתקדם ומשוכלל.
- למעוניינים לספק פתרון End to End בסביבת Web ולהפוך ל- Full Stack Developer.

מטרות

- פתיחת שער לעולם הפיתוח ולעולם האינטרנט.
- הכשרת מפתחי אינטרנט מומחים ועצמאיים, המסוגלים להקים ולנהל אתרי אינטרנט מ-א' ועד ת'.
- הכשרת מפתחים מקצועיים המסוגלים ללמוד לבד ולהרחיב את עולמם המקצועי.
- הכשרת Web Developers לניהול והצגת תכנים באתרי אינטרנט בסיסיים ומתקדמים.
- הכשרת Full Stack למתן מענה מקצועי וטכנולוגי הן בצד השרת והן בצד הלקוח, תוך הבנה מעמיקה של התהליכים והארכיטקטורה.

John Bryce Online Academy - מציעה לתלמידים מגוון רחב של קורסי Online בלמידה עצמית כהשלמה ממוקדת לתחום הנלמד. כמו כן במהלך הקורס, בנוסף לשעות הלימוד בכיתה, יתקיימו מפגשי תמיכה וליווי המתבצעים און ליין. המפגשים מאפשרים לתלמידים לקבל מענה וסיוע ממרצה/מתרגל בנוגע לחומר הנלמד בצורה יעילה וממוקדת, ללא צורך בהגעה למכללה.

תנאי קדם

- ראיון אישי / ייעוץ להכוונה מקצועית
- תואר ראשון
- מבדק התאמה פנימי

מתכונת הקורס

- 590 שעות אקדמיות (כולל 118 שעות הנחיית תרגולים)
- תל אביב: לימודי בוקר בין השעות 9.00-18.30 (8 שעות לימוד + שעותיים הנחיית תרגול)
- ארבע פעמים בשבוע

דרישות מהסטודנט

- לצורך ההצלחה במסלול יש לעמוד בדרישות הבאות:
- נוכחות בכל השיעורים של הקורס.
- מעבר בהצלחה של מבחנים ופרויקטים.

זכאות לתעודת גמר

- על מנת להיות זכאי לקבל תעודות גמר של המסלול יש לעמוד בדרישות הבאות:
- נוכחות ב- 85% מן המפגשים לפחות.
- הצלחה בפרויקטים.
- הצלחה במבחנים פנימיים.

תכני הקורס

שעות	נושא
4	מבוא טכנולוגי לעולם האינטרנט
12	HTML + HTML5
16	CSS + Advanced CSS + CSS3 + sass
4	Bootstrap
16	מבוא לתכנות
32	Basic Java Script
4	GIT
28	Advanced JS
8	סיבוכיות וביצועים
16	אלגוריתמיקה ומבנה נתונים
16	OOP WITH TypeScript
8	Jquery
16	Web-Services + REST+AJAX
4	HTML5 APIs
52	React.js
48	Node.JS + Advanced
16	Docker
16	MySQL
40	Angular
16	MongoDB
8	RabbitMQ
8	Hackathon
8	סדנת קו"ח והכנה מקצועית
76	סך כל שיעורי הנחיית ובדיקת פרויקטים
118	הנחיית תרגולים
590	סה"כ שעות לימוד

נושא	תתי נושאים
מבוא טכנולוגי לעולם האינטרנט ומבוא ל- Web Protocols	<ul style="list-style-type: none"> מושגי יסוד בתקשורת פרוטוקול HTTP אתרי Web ו- FTP סוגי דפדפנים שרתי אינטרנט ארגוני תקינה טכנולוגיות צד שרת וצד לקוח ארכיטקטורת אינטרנט העברת מידע מובנה <ul style="list-style-type: none"> What is XML: ○ Well-formed XML ○ JSON ○
HTML5 & HTML	<ul style="list-style-type: none"> עקרונות HTML: <ul style="list-style-type: none"> מבוא לשפת התגיות HTML ○ מבנה ועקרונות התחביר ○ תגיות נפוצות ומאפיינים ○ יצירת לינקים ○ הוספת תמונות ○ יצירת טבלאות על ידי שימוש ב < DIV > ○ HTML5: <ul style="list-style-type: none"> מה חדש ב - HTML5 ○
CSS	<ul style="list-style-type: none"> מבוא ל- CSS מבנה ותחביר שילוב CSS ב- HTML סלקטורים מאפייני עיצוב נפוצים מבנה תצוגה Display structure and positioning Floating elements
Advanced CSS & CSS 3	<ul style="list-style-type: none"> New selectors Pseudo elements/classes: Hover: Before: after more... Box-sizing Shadow Border Column layout Element resize Gradient Font Face Media Queries, Responsive design Transformations Transitions
Bootstrap	<ul style="list-style-type: none"> Overview Implementation and Usages Combining CSS techniques UI component
מבוא לתכנות	<ul style="list-style-type: none"> Flow chart

	<ul style="list-style-type: none"> Logical solution Variables and Types Input/Output Conditions and nested conditions Loops index, while, do-while, for-each - לולאות ולולאות מקוננות
Basic Java Script	<ul style="list-style-type: none"> Introduction to Java Script Types Flow of Control Arrays Functions and Scope Simple nonfunctional Objects Events Forms Timers and Asynchronous programming JSON and XML Local Storage
GIT	<ul style="list-style-type: none"> Set up Git GitHub Service Create a repository Fork a repository
Advanced Java Script	<ul style="list-style-type: none"> Extending objects Extending functions JavaScript Object Oriented ES6/7/8 The Browser Object Model (if time permits) Event Handling (if time permits)
סיבוכיות	<p>מודול זה מתמקד בהבנת סיבוכיות ריצה של קוד. מטרת המודול היא להקנות יכולת להתחשב בביצועים ובכתיבת קוד אפקטיבי.</p>
אלגוריתמיקה ומבנה נתונים	<ul style="list-style-type: none"> linked list queue stack dictionary array
Web Services , REST & AJAX	<ul style="list-style-type: none"> Server Definitions IP structure, DNS Request / Response mechanism Headers REST Using jQuery's AJAX Features
jQuery	<ul style="list-style-type: none"> jQuery Library Introduction jQuery basics – document ready, callback functions jQuery structure and components jQuery Selectors

	<ul style="list-style-type: none"> ▪ Traversing document elements ▪ Modifying CSS attributes ▪ Binding and unbinding events ▪ JQuery Extensions (Plugins)
HTML5 APIs	<ul style="list-style-type: none"> ▪ Messaging ▪ Web worker ▪ Sockets ▪ Canvas ▪ Geolocation ▪ Video/Audio
Type Script & OOP	<ul style="list-style-type: none"> ▪ Introduction ▪ Advantages ▪ Class And Object ▪ interfaces ▪ Functions And Constructors ▪ OOP Principles <ul style="list-style-type: none"> ○ Inheritance ○ Encapsulation ○ Polymorphism ▪ Types ▪ Generics ▪ Modules And Namespaces
React	<ul style="list-style-type: none"> ▪ Virtual DOM ▪ Components ▪ props ▪ Components Lifecycle ▪ Routing ▪ Redux ▪ Stateless vs state full
Node.js	<ul style="list-style-type: none"> ▪ Introduction to Node.js <ul style="list-style-type: none"> ○ What is Node and what is it not ○ Node.js Features ○ Our first Node.js script: Hello World ○ Building a web server in Node.js ○ Debugging node applications ▪ Building your Stack <ul style="list-style-type: none"> ○ Pulling in other libraries ○ Building custom libraries ○ A-synchronicity and callbacks ○ Blocking vs. non-blocking I/O ○ Working within the event loop ▪ Modular JavaScript with Node.js <ul style="list-style-type: none"> ○ Writing Modular JavaScript with Node.js ○ Core Modules ○ Installing Packages ○ Publishing packages ▪ Avoiding common pitfalls with Async.js

	<ul style="list-style-type: none"> ○ Introducing the Asynchronous problem ○ Async.js Library to the rescue ○ Collections ○ Flow Controllers ▪ Working with the file system <ul style="list-style-type: none"> ○ Files manipulations ○ Folder manipulations ○ Putting the file-system module together Async.js ▪ Building Web applications with the Express Framework <ul style="list-style-type: none"> ○ Introduction to Express, installation and basic setup ○ Application configuration ○ Routing ○ Views and Templating options ○ Persistence with Cookies, In-Memory Sessions and session-stores ○ Social Authentication with Passport.js ▪ Connecting MySQL Server <ul style="list-style-type: none"> ○ Database connection ○ A-synchronicity Queries from node.js ▪ Sql-Node <ul style="list-style-type: none"> ○ Creating queries with NODE-SQL ○ SELECT- Fetching data ○ Join, where, group by, having ○ Create, Drop, Alter ○ Insert, Update, Delete
MongoDB	<ul style="list-style-type: none"> ▪ Introduction to MongoDB <ul style="list-style-type: none"> ○ What is MongoDB and Why? ○ Installation and RoboMongo as Environment ○ Core concepts ○ Documentation <ul style="list-style-type: none"> • Different deployment models • JSON ▪ CRUD and the MongoDB Shell <ul style="list-style-type: none"> ○ Introduction to the MongoDB API and the core concepts of documents and collections ▪ Indexing and Schema Design <ul style="list-style-type: none"> ○ Indexing, query profiling and the query optimizer ○ Some schema design case studies ○ Practice modeling various domains ○ Schemas for atomic operations ▪ MongoDB and Node.js <ul style="list-style-type: none"> ○ Tooling up – installing mongo, clients and drivers. ○ Mongoose Schemas ○ CRUD operations ○ Single Page Applications with Express, Mongoose and Angular.js
Angular	<ul style="list-style-type: none"> ▪ Introduction ▪ Modules ▪ Components ▪ Angular Lifecycle ▪ Data Binding

	<ul style="list-style-type: none"> ▪ Directives ▪ Dependency Injection ▪ Observables ▪ CRUD operations using http service ▪ Authentication & Routing ▪ DOM Manipulation ▪ Rxjs
Docker	<ul style="list-style-type: none"> ▪ Docker for windows ▪ Docker basic – command line ▪ CI/CD how it works? ▪ Dockerizing node.js application ▪ Docker compose – if time permits
RabbitMQ	<ul style="list-style-type: none"> ▪ A message broker ▪ Connecting node.js with RabbitMQ ▪ Pulling from the queue ▪ Inserting data to the queue
Hackathon	<p>A Great class exercise, building an interesting web application e2e with full collaboration under your mentor. Demonstrating the real industry challenges, facing and solving issues in real time development.</p>
סדנת קו"ח והכנה מקצועית	<p>בשלב זה ניתן להתחיל לעבוד בארגוני הייטק, בתחומים שנלמדו. הסדנא תאפשר הבנייה נכונה ומותאמת של קורות החיים, והכנה מקצועית לראיונות עבודה.</p>
Ongoing projects	

הריני מאשר שניתנה לי ההזדמנות לעיין במסמך זה

שם: _____ ת.ז: _____ חתימה: _____ תאריך: _____