



Real Time College

**Course: Full Stack Web
Development
Duration: 650 Hours
Hands-On-Training**



www.rt-ed.co.il



info@rt-ed.co.il



077-7067057

רח' מרדכי רוזנסקי 14, ראשון לציון



מי אנחנו ?

קבוצת **Real Time Group** הוקמה בשנת 2007, החברה מספקת פתרונות עסקיים וטכנולוגיים בתחומי התוכנה, החומרה והתקשורת, פיתוח ותמיכה בפרויקטים מולטי-דיסציפלינאריים במערכות: **Real-Time \ Embedded \ Multi-Core \ Linux Kernel \ Embedded Linux** למגוון לקוחות בארץ ובחו"ל.

הקבוצה מורכבת משלוש חטיבות הבאות: בית תוכנה, מרכז הדרכה והשמה/מיקור חוץ.

Real Time College

חטיבת ההדרכה מובילה בהדרכה למסלולי הכשרה מקצועית לתחום ההייטק וטכנולוגיית המידע, מסלולי הכשרה והסבה לתחום ההיי-טק וטכנולוגיית המידע.

- מיקור חוץ והשמת בוגרים בחברות הייטק מתקדמות.
- קורסים ייעודיים בקבוצות קטנות בתחום התוכנה, התקשורת והמחשבים.
- מגוון רחב של קורסי מחשוב וטכנולוגיות מידע במתכונת אישית ו/או קבוצתית.

Real Time Development

חטיבת הפיתוח, אחראית על תכנון, פיתוח, בדיקות וייעוץ במערכות תוכנה, מערכות משובצות מחשב, מערכות תקשורת ובמתן פתרונות טכנולוגיים בנושאים הנ"ל:

- HW/SW Development
- Networking Applications
- Web Testing
- Network Testing
- QA and Automation Tests
- RT / Embedded Applications.
- BSP/ Low Level Programming / Board Bring Up.
- Linux User/Kernel/Embedded Applications
- Manufacturing Testing

Real Time H.R

חטיבת ההשמה אחראית על מיקור חוץ והשמה של הבוגרים בחברות ההייטק וטכנולוגיית המידע. בסיום המסלול החברה מבצעת מיקור חוץ או השמה לבוגרים במסלולים הבאים:

- מפתחי Full Stack ו-Linux & RT Embedded בחטיבת הפיתוח של החברה.
- מיקור חוץ לשותפות עסקיות של החברה.
- השמת הבוגרים בחברות היי טק.

אודות המסלול:

מסלול Full Stack מעניק את כלל הידע, הניסיון והכלים הנדרשים לסטודנט בכדי להצטרף לצוות פיתוח כמפתח צד שרת (Server) והן צד לקוח (Client).

כאמור הפיתוח ב WEB נחלק לשני תחומים:

צד שרת (Server Side)

צד השרת נותן מענה ומשגר נתונים (שאליות) שהתקבלו מהלקוח, בהינתן ובכל רגע נתון יכולות להיות עשרות אלפי בקשות ישנו צורך בחישוב תשובות מהירות, אחסון המידע וטעינתו, צד השרת הינה לב ליבה של התוכנה.

צד לקוח (Client Side)

צד הלקוח הינו ממשק המשתמש, לקוח קצה, במהלך השנים גדלה כמות המכשירים והדרכים שאנו צורכים את המידע והיום ישנם אינספור טכנולוגיות, כמו כן, הצרכים השתנו, אם בעבר הייתה דרישה לעמוד סטאטי והשאררת פרטים, היום הדרישות מורכבות, הכוללות בין השאר תקשורת אינטראקטיבית, הצגת מידע מורכב, מילוי טפסים, עיבוד בזמן אמת והצגתן של מפות אינטראקטיביות.

מהו תפקיד המפתח Full Stack ?

בשנים האחרונות חל מפנה בתחום הפיתוח, אם בעבר ציפו מהמפתח לכתוב צד שרת ולהכיר צד לקוח (Web) או לכתוב צד לקוח ולהכיר צד שרת.

כיום חברות הייטק דורשות מהעובדים ידע וניסיון במגוון רחב של שפות תכנות וטכנולוגיות ברמה גבוהה במטרה ליישם צד לקוח וכן צד שרת. התהליך התחיל מתוך הבנה של חברות ההייטק שבכדי לקדם את הפיתוח יש "לאחד" כוחות ולהקטין את משך ההמתנה, למנוע אי-הבנות בין המחלקות ולהגדיל את הוורסטיליות והגמישות של מוצר.

מהו מסלול Full Stack ?

זהו המסלול המלא כיום בתחום ה Full Stack Web Development,

המסלול כולל את כלל הקורסים אשר נועדו להעניק לך את הידע והניסיון המקצועי הנדרשים במגוון טכנולוגיות, לרבות שפות תכנות ופלטפורמות כגון: AngularJS, CSS3, Javascript, MySQL, NodeJS, ועוד.

הסילבוס נקבע בתיאום עם השותפות העסקיות – חברות הייטק המעוניינות לקלוט את בוגרינו.

מידע על המסלול

כחברת ההייטק המובילה באספקת שירותי מומחה ולמגוון לקוחות מעל כעשור, אנו מודעים לדרישת חברות ההייטק, הנושאים הרלוונטיים והידע הנדרש בכדי ליישם אותך במשרה רלוונטית.

אנו ב Real Time Group פיתחנו **מסלול ייחודי להכשרה והשמה לתחום Full Stack Web Development** -

המטרה במסלול זה היא להכשיר אותך כמומחה - Full Stack Web Development, להעניק לך את כלל הידע והניסיון הנדרשים בכדי להוות מקצוען בתחום. לקראת סוף ההכשרה, חטיבת ההשמה תבנה קו"ח עם המועמד ותחל עמו את תהליך ההשמה/מיקור-חוץ לחברות הייטק רלוונטיות לתחומן.

לבסוף מטרת המסלול - לשלב אותך בתעשייה כמפתח בחטיבת הפיתוח בחברתנו או באחת מהשותפות העסקיות.

תיאור המסלול:

- המסלול נערך במתכונת של קורסים מעשיים הכוללים הרצאות עיוניות, תרגול מעשי רב, עבודת שטח מעשית וכן סטאז' מעשי בחטיבת הפיתוח של החברה.
- בחירת התמחות AngularJS או ReactJS
- היקף התוכנית – 650 שעות
 - **500 שעות** פרונטליות ומעשיות במחלקת ההדרכה
 - **150 שעות** – סטאז' מעשי בחטיבת הפיתוח של החברה
- הלימודים מתקיימים:
 - מסלול ערב - 2 מפגשים בשבוע בין השעות 18:00-22:00 – משך מסלול ערב- כשנה.
 - מסלול בוקר - 3 מפגשים בשבוע בין השעות 9:00-16:00 - מסלול בוקר - 6 חודשים.

מטרות המסלול:

- המסלול נועד להעניק לסטודנטים את הידע הנדרש בכדי לשלבם בתעשייה Full Stack Web Development הסילבוס גובש בתיאום עם השותפות העסקיות - חברות הייטק אשר מעוניינים לקלוט את בוגרנו.
- המסלול כולל מבחן עבור כל מודול נלמד.
 - הגשת התרגילים על בסיס שבועי.
 - קיים מועד נוסף עבור נכשלים / נעדרים.
 - נדרשת נוכחות של לפחות 85% ועמידה במבחנים כתנאי הכרחי עבור קבלת תעודת המקצוע.

קהל היעד:

- כלל המעוניינים להתמקצע בתחום האינטרנט ופיתוח בעולם ה-Web.
- חסרי רקע קודם בתחום ה-Web או בעלי רקע בסיסי.
- בעלי תואר ראשון או הנדסאים המעוניינים בהסבה מקצועית.

תנאי קדם:

- אנגלית ברמה בינונית .
- בעלי רקע ו/או ניסיון בפיתוח שפות תכנות – יתרון
- ראיון ידע מקצועי + בחינה .

סטאז' מעשי:

Hands On Training

במהלך המסלול הסטודנט יבצע סטאז' מעשי במשך 150 שעות מלווה במהנדס מחטיבת הפיתוח של החברה, פיתוח הפרויקט יהיה בחפיפה מלאה עם המודול הנלמד במטרה ליצור תיק עבודות וניסיון מעשי אשר יהווה הוכחה לידע וניסיון של המועמד במטרה למציאת עבודה כמפתח Full Stack **(תהליך הפרויקט)**

בסוף הסילבוס

מציאת עבודה

- לקראת סיום הלימודים יועבר הסטודנט לחטיבת ההשמה לצורך תהליך השמה בחברות הייטק.
- כחלק מתהליך ההשמה תדאג החברה ל
 - בניית קו"ח אופטימאליים לכל סטודנט.
 - שיקוף ציוני הקורסים ופרויקטים של הבוגרים לשותפות העסקיות של החברה.
 - עבור כל ראיון עבודה- הסטודנט יעבור הכנה אישית לראיון המקצועי, אחד על אחד.

קבלת תעודה:

- בסיום המסלול הסטודנט יקבל תעודה מטעם מכללת Real Time College.
- באישור משרד הכלכלה / תמ"ת.

תכני המסלול

Foundations	10
Basic Web services	10
HTML5	20
CSS3	20
Bootstrap	20
Javascript	60
Jquery	40
MongoDB	30
MySQL	30
GIT	10
Security	10
Development to Production	20
NodeJS	100
בחירת התמחות	
התמחות בכלים של Google	
Typescript	40
AngularJS	80
התמחות בכלים של Facebook	
ReactJS	60
Native React	60
Total	500

פירוט תכני המסלול

Foundation	<ul style="list-style-type: none"> • Web Basics • HTTP protocol • HTTPS protocol • Client and Server side • Web servers • Domains
Basic Web services	<ul style="list-style-type: none"> • Servers role and tasks • IP structure • DNS • Request & Response • Headers • REST
HTML5	<ul style="list-style-type: none"> • HTML introduction • Structure • Main tags and attributes • Links and images • Tables and frames • Html imports • New in HTML5
CSS3	<ul style="list-style-type: none"> • CSS Introduction • CSS in HTML • Selectors • Best practices • Animations • Transitions • Media queries and responsive design • Page templating and layouts • Colors and gradients • Elements manipulation • Font families • Inline / Outsource
Bootstrap	<ul style="list-style-type: none"> • Obtaining and using bootstrap • Tables • Forms • Lists • Grid system • Typography • Bootstrap best practice • Re-designing bootstrap style

Javascript	<ul style="list-style-type: none"> • Introduction to programming • What is Javascript? • Html and Javascript • Internal javascript • Selecting elements • Working with events (forms, click...) • Changing content • Styles manipulation • Variables • Arrays • Objects • If statement • Loops • Functions • External Javascript • JSON • Object Oriented Javascript • Async Javascript • Browser object model • Event Handling
Jquery	<ul style="list-style-type: none"> • Jquery Introduction • Jquery in HTML • selectors • Working with Jquery events • callback functions • Changing content • Styling with Jquery • Animating content • AJAX • Regular expressions • Jquery events • Effects and animations • Validation • Jquery plugins • Jquery UI

MongoDB	<ul style="list-style-type: none"> • Introduction • Installation • MongoDB API • Core concept • Environment • JSON • Indexing • Query optimizer • Schema Design • CRUD and MongoDB Shell • MongoDB and NodeJS
MySQL	<ul style="list-style-type: none"> • Introduction • SQL differences • MySQL data types • create & delete database • create, delete, duplicate tables • column definition: Index, Unique, Auto Increment.. • Query syntax CRUD • Advanced queries : Join, Union • Connecting to database • Retrieving data from a database • Inserting and updating data • Looping through data • Storing passwords securely • Secure Habits and Best practice
GIT	<ul style="list-style-type: none"> • Git Setup • Repositories • Commit • Branches • Forking • Working with Git
Security	<ul style="list-style-type: none"> • DDos • XSS • SQL injection • CSRF • Man in the middle • cross-site scripting attacks and other cross-site injections • X-Powered-By header • Strict-Transport-Security • clickjacking protection • MIME-sniffing • client-side caching

Development to Production	<ul style="list-style-type: none"> • Buying & setting domain • Buying server • Server Configurations • Setting DNS • Uploading and working with FTP client-side caching
NodeJS	<ul style="list-style-type: none"> • Introduction to NodeJS • Building NodeJS application • Core Modules • Installing packages • Using built in libraries • Building custom libraries • A-synchronicity & callbacks • blocking vs non-blocking I/O • Sync and Async operations • Async common errors • Working with file systems • Real time listen and emit • Server and client sockets • ExpressJS introduction • Express Installation and setup • Express routing • Express templating and views • Express cookies and sessions • Authentication with PassportJS • Social Auth with Passport

התמחות בכלים של Google	
Typescript	<ul style="list-style-type: none"> • Introduction • Basic data types • Destructuring • Classes • Interfaces • Generics • Modules • Namespaces • Ambients • Decorators • Advanced Types • Symbols
AngularJS	<ul style="list-style-type: none"> • Angular Introduction • Building single page applications • Template • Controller • Scope & Scope Inheritance • Directives • Modules • Dependency Injection • Services • Interpolation • Rendering HTML • DOM event handlers • Filters • Validation • Routing • Server communication • Custom directives

התמחות בכלים של Facebook	
ReactJS	<ul style="list-style-type: none"> • Introduction • Virtual DOM • ES6 • Classes • Modules • JSX • Naming conventions • Rendering • Parser • Components • Props • Validation Props • State • Life Cycle • Reusable components • Forms • Events • Debug • API Calls • Flux • Animations
Native React	<ul style="list-style-type: none"> • Introduction • Accessing the library • Built Components • Lifecycles • Styling • Handling Layout • Reusing Components • Handling user input • Form validation • Redux • Animations • Navigation

קורס	חלק בפרויקט
תיאור הפרויקט	<p>בפרויקט זה נפתח אפליקציית אינטרנט. מטרת הפרויקט היא לחוות ולהתנסות בפיתוח פרויקט מתחילת הדרך של אפיון ותכנון עד לשלב בו יש לנו אפליקציית אינטרנט על שרת ייעודי, מאובטחת, מוצפנת מוכנה לקבל ולשרת את הקהל יעד שלה.</p> <p>הפרויקט ידמה פעילות של חברת נדל"ן.</p> <p>בשלב הראשון אפיון של הפעולות, העיצוב, קהל יעד, זרימת מידע ואף נכין API שיאפשרו לאחרים להשתמש בשרת והנתונים שלנו.</p> <p>שלב שני נתחיל לתכנן את סביבת העבודה שלנו, תוכנות, ספריות ואולי התקנות נוספות לביצוע הפרויקט.</p>
HTML5	<p>שלב שלישי הינו הכנת השלד של הפרויקט. בעזרת קוד HTML נכין תשתית עבור האפליקציה, נפתח עמודים רלוונטיים ונחלק את הפרויקט לתיקיות נכונות.</p> <p>בשלב זה ניתן יהיה כבר לראות את סידור הפרויקט, מכיוון שחלק מהתיקיות עם תוכן תהינה אפשרות לפתוח את העמודים בדפדפן ולהרגיש את הפרויקט בפועל.</p>
CSS3 & Bootstrap	<p>שלב רביעי הוא שלב העיצוב. בשלב זה נשתמש בעמודים מהשלב הקודם ונלביש עליהם את העיצוב על פי האפיון שיצרנו.</p> <p>כעת ניתן להשלים את הנראות של העמודים שלנו הכוללים צבעים לרקע ולטקסט, נסדר את התוכן במיקום נכון על גבי העמוד, נשתמש באנימציות ונכין סקיצות של נתונים מדומים על גבי העמודים לצורך דגימה ראשונית של נראות הפרויקט.</p> <p>בשלב זה לעתים המפתח שולח את הסקיצה ללקוח ומקבל הערות ואישורים להמשך הדרך.</p>
Javascript & JQuery	<p>שלב חמישי הוא שלב התכנות של הפרויקט. בשלב זה אנו נוסף אינטראקציה עם הגולש, תגובות לאירועים, לחיצות והקלדות של הגולש.</p> <p>בשלב זה הגולש כבר יכול להתחיל להרגיש שהאתר פעיל לחלוטין ויכול להגיב לפעולות שהוא מבקש ממנו אך בפועל אנחנו עדיין לא עובדים עם שרת ולכן הכל עדיין לא יעבוד באמת.</p>
MongoDB / Mysql	<p>שלב שישי הינו הכנת מאגר נתונים. בשלב זה נכין טבלאות וקשרים על פי האפיון שיהיו מוכנים לקבל מידע ולספק מידע לאפליקציה שלנו.</p>
NodeJS	<p>שלב שביעי הינו פיתוח שרת עבור הפרויקט שלנו, לאחר קריאה והבנה של האפיון נכין שרת המסוגל לעבוד עם המידע מהמאגרים. השרת יקבל תזרים מידע וידע לאבד ולשמור אותו במאגר נתונים. וכמובן לספק אותו בכיוון ההפוך כשהגולש צריך.</p> <p>בשלב זה נתקין את הספריות שקבענו בתכנון הפרויקט ונשלב אותם בקוד.</p>

<p>שלב שמיני הוא פיתוח צד לקוח. בשלב זה אנו נשתמש בקוד שכתבנו עד עכשיו ונמיר אותו לקוד מתקדם יותר.</p> <p>בשלב זה הפרויקט קם לחיים ומצליח לנהל את כל הפעולות שרצינו באפיון. מצד הלקוח אשר מזין פרטים או פעולות ועד התזרים מידע ואיבודו בצד שרת ושמירתו במאגר.</p> <p>שלב זה הינו השלב האחרון בפיתוח היכולות של האפליקציה והשימוש בה, עתה היא מוכנה לשימוש.</p>	<p>ReactJS / AngularJS</p>
<p>שלב תשיעי הינו שלב האבטחה. לאחר שהאפליקציה מוכנה לשימוש ישנם כמה שלבים שנרצה לבצע על מנת להגן על הקוד והתוכן שלנו בפני פריצות וחשיפת מידע רגיש שלא בכוונה.</p> <p>נשתמש בספריות שהתקנו ונוסיף קוד נוסף לשמירת הפרויקט שלנו כך שיעשה את האמור באפיון ולא אף דבר נוסף שהפורצים ינסו להוסיף או להפך לקחת.</p>	<p>Security</p>
<p>שלב עשירי הינו השלב בו אנו נעלה את הפרויקט לשרת. כעת ניתן יהיה לגשת אליו בכתובת הרצויה, להשתמש בו מכל מקום בעולם.</p>	<p>Development to Production</p>
<p>שלב אחרון הינו להעלות אותו לשרת הגרסאות שלנו ולשמור אותו שם. כעת יש לנו גרסה ראשונה ומוכנה בשרת. כעת ברגע שנרצה לערוך אותה נוכל לעשות זאת באמצעות קטעי קוד ולשמור את השינויים בגרסאות נוספות כך שתמיד יהיה לנו גישה לגרסאות ישנות יותר אם נצטרך או אם נשתף את הקוד עם הצוות.</p>	<p>GIT</p>

הערות :

- פתיחת המסלול מותנה במספר נרשמים.
- דמי הרשמה אינם מוחזרים אלא במקרה של אי פתיחת מסלול ע"י המכללה.
- דמי הרשמה כלולים בשכר הלימוד.
- המכללה מביאה ידעתם של תלמידים שיתכנו שינויים בתוכן הקורסים ובמועדם.
- המכללה מתחייבת להודיע לתלמידים על כל שינוי.
- המכללה שומרת לעצמה את הזכות לשנות את תכני המסלול בהתאם לשיקול דעתה הבלעדית.



