כותבים: אלמוג וגמן ציפרוט, יונתן טל תאריך: 27.10.2017

חפיפת פיתוח אתרים

הקדמה

במסמך זה יש את כל המידע הנחוץ על מנת להגיע לתפקיד פיתוח בענף.
המסמך מקיף ומכיל חומרים אשר ידאגו שהנחפף יהיה מוכן לעבודה בצוות הן בפן הטכנולוגי והן בפן המקצועי.
החפיפה נבנתה מנקודת הנחה שלנחפף אין רקע בפיתוח אתרים.
לנחפף בעל ידע קודם מומלץ לעשות גרסה מקוצרת של חפיפה זו על לפי החלטת החופף והמפקדים.

החפיפה תתמקד בנושאים הבאים:

- הבנה בסיסית של מה זה אתר ואיך הוא עובד.
- בניית אתר בסיסי בעזרת HTML ו- CSS.
- לימוד Javascript באופן יסודי ומקיף.
- בניית צד שרת ב- NodeJS.
- עבודה נכונה בצוות.
- ניהול קוד בצורה נכונה עם Git.
- לימוד דרך העבודה של הענף.
- הבנת הצורך בבדיקות.

בכל נושא יינתן לנחפף מקור מידע עיקרי ממנו מומלץ ללמוד את החומר. בנוסף למקור המידע העיקרי, יינתנו מקורות מידע נוספים מהם ניתן ללמוד חלקים מסוימים. תוך כדי המעבר על החומר יינתנו משימות תרגול בסיסיות שעל הנחפף לעשות. משימות אלו יבדקו על ידי החופפים ויהוו שלב הכרחי על מנת לעבור לחלק הבא.
במהלך החפיפה הנחפף יפתח פרויקט מרכזי בשלבים. הפרויקט יכלול בתוכו את כל נושאי החפיפה ויהווה מעין פרויקט גמר.

מבוא

הענף שלנו נוקט בגישה של קוד פתוח ושימוש בטכנולוגיות חדשות ומתקדמות, לכן, אנו מתעסקים במגוון רחב מאוד של טכנולוגיות, דבר אשר לא מאפשר כתיבת חפיפה אחת מקיפה.
מסיבה זו, חילקנו את החפיפה למספר חלקים:

1. **חלק ראשון - חפיפה בסיסית.**
   חלק זה יתמקד בבסיס של בניית אתרים, הוא יסביר מה זה אתר ואיך בונים אותו.
   את חלק זה כל הנחפפים יעשו.
2. **חלק שני - חפיפה מתקדמת.**
   חלק זה יתמקד בצד השרת בכללי וב- NodeJS בפרט.
   לכן, לא כל הנחפפים יצטרכו לגשת לחלק זה, דבר אשר תלוי בצוות אליו הם יגיעו.
3. **חלק שלישי - Frameworks.**
   חלק זה יתמקד ב- frameworks בהם נעבוד בצד הלקוח, הוא יתחלק ל- Angular ול- React, כאשר כל נחפף ילמד רק את אחד מהם או, במקרה והגיע לצוות שלא מתעסק בהם הוא יעבור לחפיפה הצוותית.

חפיפה זו לא מיועדת להיות מקור הידע שלכם, היא רק נועדה להדריך אותכם ולתת לכם קו פעולה כללי לדרך הלמידה.
על מנת שתדעו שאתם לומדים כראוי, ניתן לכם משימות אותם תצטרכו לבצע.
משימות אלו יעזור לכם להבין שאתם לומדים במסלול הנכון ולא הולכים לאיבוד בחומר.

בהצלחה!

חלק ראשון - חפיפה בסיסית

חלק זה בחפיפה יעזור לכם להבין איך אתרי אינטרנט עובדים, איך בונים אותם, איך מעצבים אותם ואיך גורמים להם לעשות בדיוק מה שתרצו שהם יעשו.

מקור הידע הראשי בחלק זה יהיה מסמך לימוד בניית האתרים של MDN.
אינכם חייבים לעבור על כל מקור זה, אם יש דברים אותם אתם כבר בטוח יודעים, אתם יכולים לדלג עליהם.

מבוא

לפני שתתחילו ללמוד איך לבנות אתר, עליכם להבין מה זה אתר ואיך הוא פועל.

במודול Getting started with the Web קראו את:

- Installing basic software
- What will your website look like
- Dealing with files
- How the web works

וענו על השאלות הבאות:
(בנוסף למקור זה, ניתן גם לקרוא ממקורות אחרים על מנת לענות על השאלות).

1. הסבר במילים שלך מהם המושגים הבאים:
   a. HTML
   b. CSS
   c. JavaScript
2. ענה על השאלות הבאות:
   a. בעזרת איזה פרוטוקול ניתן להעביר HTML, CSS ו- Javascript ?
   b. מה עוד ניתן להעביר בפרוטוקול זה ?
   c. מהי הגרסה המאובטחת של אותו הפרוטוקול ?
3.
   a. מי זה טים-ברנס לי ?
   b. מה זה w3c ?

HTML

לאחר שהבנתם מה זה אתר אינטרנט אתם מוכנים להתחיל לבנות אותם.
קראו את:

- HTML basics
- What's in the head? Metadata in HTML
- HTML text fundamentals
- Creating hyperlinks
- Document and website structure
- Debugging HTML
- HTML table basics
- Your first HTML form
- The native form widgets
- Sending form data
- Form data validation
- Same Origin Policy

מומלץ להיעזר ב- w3schools על מנת ללמוד אילו אלמנטים קיימים ואיך להשתמש בהם.
אין צורך ללמוד לעומק כל אלמנט, אך מומלץ להכיר את כולם ולדעת על קיומם.

CSS

לאחר שסיימתם לקרוא על איך לבנות את הבסיס לאתר, כעת תלמדו איך לעצב אותו ולגרום לו להיראות בדיוק כפי שאתם רוצים.

קראו את החלק:

- CSS basics
- CSS syntax
- CSS Debugging
- המשיכו אל CSS Tutorial שב- w3schools ותקראו עד הפרק CSS Website Layout כולל.

לאחר סיום הקריאה, בצעו את המשימות הבאות.
ניתן להיעזר גם ב- CSS-Tricks או בכל מקור ידע אחר.

משימות

בחברות חיצוניות ישנו לרוב מעצב שאחראי על מראה האתר. על המפתחים בחברה לבנות אתר אינטראקטיבי על בסיס ה"תמונות" שהמעצב הכין.

לפניכם דוגמאות לעיצובים. בנו דפי HTML ובעזרת CSS עצבו אותם בכדי שיראו זהים לעיצובים הנתונים וכאשר הם מגיבים בהתאם.

1. עצבו דף Login לאתר: גשו לדף המשימה הנקרא "משימה 1 - עיצוב דף התחברות"
2. גשו אל החופפים אל קבלת המשימה הבאה.

בונוס

- Sass

JavaScript

לאחר שלמדתם איך אתרים עובדים ואיך לבנות ולעצב אותם, נשאר לכם ללמוד רק עוד דבר אחד בצד לקוח, איך לגרום לאתר לבצע פעולות מסוימות.

קראו את JavaScript basics.
ולאחר מכן, צפו בקורס JavaScript: Understanding the Weir d Parts שב- Udemy

שם המשתמש:
yesodot.anaf@gmail.com
סיסמא:
Crzhk2014!

מקורות נוספים בהם אתם יכולים לעיין:
- The Modern Javascript Tutorial
- You Don’t Know JS
- Eloquent Javascript

משימת JavaScript

לאחר שעשיתם את הקורס, ענו על השאלות הבאות ובצעו את התרגילים:
כנסו אל w3resource ופתרו תרגיל מכל נושא (ב- Basic ו- DOM בצעו תרגיל נוסף).

תרגיל מסכם - חפיפה בסיסית

כתבו משחק בעזרת HTML, CSS ו- Javascript ללא שימוש בחבילות נוספות.
משחקים מומלצים:
- Tetris
- Snake

ניתן לבחור גם משחק אחר לאחר התייעצות עם החופף.

בונוס

קראו על ה- frameworks הבאים:

- TypeScript
- JQuery

חלק שני - חפיפה מתקדמת

ברוכים הבאים לחפיפה המתקדמת, את החפיפה הזו, רק חלק מהנחפפים יעשו, לפי הצוות אליהם הם הגיעו, והמפקד שלהם.

NodeJS

צפו בקורס הבא במלואו: Learn and Understand NodeJS

בדיקות

כל פרויקט, בסופו של דבר, צריך לעבור שינויים במהלך דרכו, אם זה ל- refactoring או להוספת features.
כאשר אנו מוסיפים או משנים קוד בפרויקט, קיים סיכוי שנשבור את פעולתו התקינה או שנפגע בפעולתו של קוד אחר.
אנו עושים testing כדי שלאחר כל שינוי, נוכל לבדוק שהכל עובד כמצופה.

בכל פרויקט שתכתבו, כולל במשימת חפיפה, תצטרכו לעשות testing.
אותן הבדיקות שתעשו, לא רק יעזרו לכם, אלא גם למתכנתים האחרים שיגיעו לפרויקט אחריכם.

ישנם מספר סוגי בדיקות אשר רלוונטיות אליכם כרגע:

- Unit Testing
- Integration Testing

תקראו על נושאים אלו ותבינו את המשמעות שלהם.
לאחר מכן, עליכם ללמוד להשתמש בחבילות המתאימות ולהוסיף בדיקות לקוד שלכם.

אלו החבילות המקובלות לשימוש כיום ב- NodeJS:

- Mocha
- Chai

באפשרותכם לבחור חבילות אחרות ולהשתמש בהן, אך מומלץ להתייעץ תחילה עם החופפים או עם מקורות ידע אחרים בענף.

ניהול קוד

כאשר כותבים פרויקט, מומלץ לעבוד עם כלים שיעזרו לנו לעקוב אחר התקדמות הקוד שלנו, לשמור אותו במקום בטוח ולנהל אותו.
לכן מומלץ להשתמש ב- version controls, כמו git, ובשרת repo של git כמו- github ו- gitlab.

Git גם נועד לעבודה בצוותים וזו הדרך בה אנו נעבוד בצוותים בענף על מנת לשמור על סדר ותקינות הקוד.

ניתן ללמוד git מהמקורות הבאים:

- Pro Git
- Try Github
- Learn Git Branching

מומלץ להתמקד בדרך נכונה לעבוד ב- Branches, מפני שזו הדרך התקינה לעבוד בפרויקט עם מספר מתכנתים ואפילו בפרויקט עם מתכנת בודד.

בסיס נתונים

בחלק גדול מהפרויקטים, עליכם יהיה לשמור מידע עבור מטרות מסוימות.
על מנת לשמור את המידע, אנו נשתמש בבסיס נתונים (Database).

בחפיפה זו, נתמקד בשימוש ב- MongoDB, זהו בסיס נתונים לא רלציוני, המבוסס על documents, לכן, אין בו SQL.

תקראו בקצרה על MongoDB ותבינו איך הוא עובד.
בצד השרת אותו תבנו, תוכלו להשתמש בחבילות כמו Mongoose אשר יעזור לכם לתקשר עם ה- DB.

אפילו אם הצוות שלכם לא עובד עם MongoDB מומלץ להכיר אותו, על מנת לשמוע על דרכי עבודה שונים.

במקרה ואתם כבר יודעים איך עובדים עם MongoDB, אך לא יודעים איך עובדים עם בסיס נתונים רלציוני (כמו SQLServer, MySQL וכדומה) מומלץ ללמוד SQL. במקרה זה, פנו אל החופף שלכם.

סטנדרטים

כאשר מספר מתכנתים עובדים על אותו הפרויקט, מומלץ לשמור על אחידות.
כאשר עובדים באופן אחיד בענף, קל יותר למתכנתים לקרוא קוד של מישהו אחר ולהבין איך הוא פועל.

בנוסף, כאשר מישהו יגיע לקוד שלכם לאחר שכבר עזבתם, הוא לא יצטרך את העזרה שלכם על מנת להבין את פעולתו הבסיסית של הקוד, ולא רק זה, גם כאשר אתם תחזרו לקוד לאחר מספר חודשים יהיה לכם קל יותר להבין אותו.

על מנת שנשמור על אחידות,אתם נדרשים לעבוד על פי הסטנדרטים הבאים ב- NodeJS.

משימת חפיפה - חלק 1 - צד שרת

כתבו צד שרת אשר ישמש כתשתית למשימת החפיפה שלכם.
משימת החפיפה יכולה להיות אחת מהאפשרויות הבאות:

- צא'ט (בזמן אמת)
- הצעה שלכם (להתייעץ עם החופף) - ניתן לחשוב על מערכת שתעזור לבסיס.
- הצעה של המפקד שלכם.

בחלק זה של משימת החפיפה ניתן להשתמש בכל חבילה (npm) או framework שתרצו.

במהלך כתיבת הפרויקט, עליכם לבצע בדיקות, לעבוד על פי הסטנדרטים ולהשתמש ב- git.

חלק שלישי - Frameworks

הגעתם רחוק... אתם כבר יודעים את הבסיס של פיתוח אתרים ואתם אפילו יודעים איך לבנות את צד השרת.
כעת, נשאר ללמוד איך לייעל את העבודה שלכם בצד הלקוח. לשם כך, נלמד להשתמש באחד מה- frameworks הידועים והמשומשים ביותר כיום (בענף, ובאזרחות): Angular ו - React.

במה אותם frameworks עוזרים לנו ?
הם עוזרים לבצע בקלות יותר משימות שלרוב היו לוקחות הרבה זמן, הם עוזרים בעיצוב, במבנה ובפונקציונלית של האתר.

כעת, תצטרכו לבחור ללמוד רק framework אחד, React או Angular, על פי הצוות אליו הגעתם והחלטת המפקד\ת שלכם.

Angular

על מנת ללמוד Angular, צפו בקורס The Complete Angular Course: Beginner to Advanced שב- Udemy.

לאחר הצפייה בקורס, עליכם לעמוד בסטנדרטים הבאים: קראו את Angular Style Guide.

לאחר שלמדתם איך עובדים נכון עם Angular, עבורו אל החלק השני של משימת החפיפה.

משימת חפיפה - חלק 2 - צד לקוח

כעת, לאחר שכתבתם את צד השרת כתבו את צד הלקוח של המערכת אותה בחרתם בחלק הראשון בעזרת ה- Framework שלמדתם.

במשימה זו אתם יכולים להשתמש בכל חבילה (npm) או framework שתרצו, בתנאי שהוא עובד כראוי עם ה- framework שלמדתם.

כפי שאמרנו בתחילת החפיפה, בחברות אזרחיות קיימים מעצבים אשר מעצבים בשבילכם תבנית (template) לאתר ומספקים לכם תמונה או sketch על מנת שתעתיקו אותו. אבל, אצלנו בענף, אין מעצבים, לכן, אנו צרכים ללמוד לעצב בעצמנו ולנסות שהעיצובים יהיו כמה שיותר יפים, מקצועיים, קלים לשימוש ונוחים לעין.
.
מסיבה זו, מומלץ ללמוד עקרונות עיצוב אתרים, אחד מהעקרונות הידועים לעיצוב אתרים נקרא Material Design, מומלץ שתקראו עליו.

ישנם חבילות אשר מתבסוסות על Material Design:

- Angular Material - בשביל Angular
- Material-UI - בשביל React
- קיימות חבילות נוספות ו- components סינגולריים.

גם במשימה זו עליכם לשמור על הסטנדרטים ולהשתמש ב- git ולשתף אותנו עם ה- repo שלכם.

סיכום החפיפה

כל הכבוד, הגעתם לסוף.
למדתם איך אתרים עובדים, איך בונים אותם, איך מעצבים אותם, איך גורמים להם לעשות בדיוק מה שתרצו שהם יעשו.

גם למדתם איך לבנות את התשתית לאתר, את צד השרת, ואיך לבנות את צד הלקוח בצורה מהירה ואיכותית.
בנוסף, למדתם איך כותבים קוד נכון, איך מנהלים את הקוד שלכם ואיך דואגים שהוא ישאר תקין לאורך כל תהליך הפיתוח.

כעת, אתם מוכנים להתחיל לעבוד על פרויקטים.

ברוכים הבאים לענף שלנו.

בהצלחה.