

## Front-End Web Development

### פרק שלישי – JavaScript + OOP + Communication – שיעורים 7 + 8 + 9

#### סיכום והיערכות לפרויקט – 75 דקות – יבוצע בסוף שיעור 6

#### סיכום פרק שלישי – 15 דקות – יבוצע בסוף שיעור 6

כיתתי: סיכום – נושאים שנלמדו בפרק השלישי

Single Page Application	JS Objects
Client-Side Routing	JS Object Prototype
Internet Protocols	JS Prototype Chain
HTTP	JS Functions
GET / POST / PUT / DELETE	JS Constructor Functions
REST API	JS Classes
JSON	JS this
AJAX	call / apply / bind
JavaScript Exceptions	ES6 const + let
Try / Throw / Catch / Finally	ES6 Arrow Functions
Regular Expressions	ES6 Classes
Client-Side Code	Object-Oriented Design
Server-Side Code	Browser Object Model

#### הגדרה כללית של הפרויקט השלישי – 15 דקות – יבוצע בסוף שיעור 6

כיתתי: פרויקט – מימוש מערכת end-to-end – הנחיות כלליות:

**צד לקוח – Client** – יישום לגישה ושינוי מידע הנשמר בצד השרת. יישומים אפשריים יכולים להיות – מערכת ניהול אנשי קשר (Contacts), מערכת ניהול רשימת משימות (To-Do List), מערכת ניהול פגישות (Meetings), וכדומה. היישום יאפשר למשתמשים שונים לגשת למידע שלהם (כאשר בכל כניסה רק משתמש אחד פועל ביישום). היישום יכלול עמוד כניסה למשתמש קיים, עמוד רישום למשתמש חדש, ועמוד פעילויות במידע למשתמש פעיל. העמודים ייבנו בארכיטקטורת SPA.

**צד שרת – Server** – תוכנה המקבלת מיישומי לקוח מתאימים פניות לגישה ושינוי מידע. תוכנת השרת בודקת את הפניות ומטפלת בהן בהתאם (גישה למידע, הוספת מידע, שינוי מידע, מחיקת מידע, וכדומה). תוכנת השרת אינה נוגעת במידע בעצמה – הטיפול נעשה על-ידי התוכנה של מאגר המידע (ראו בהמשך) אשר השרת משתמש בשירותיה. השרת מחזיר ליישום הלקוח את המידע הרלוונטי כולל שדות המציינים את מצב הטיפול בפנייה.

**מאגר מידע – Data Base** – תוכנה השומרת מידע רלוונטי של משתמשים שונים ומספקת מגוון פעולות שונות של גישה ושינוי המידע. לצרכי הפרויקט, ניתן לממש את מאגר המידע כמערכים ואובייקטים של פריטי מידע הנשמרים ב- Local Storage עם מפתחות מתאימים. גישה למאגר המידע תהיה רק דרך השרת תוך שימוש ב-API מוגדר (סט של פונקציות למימוש פעולות שונות במאגר המידע).

**הדמיית תקשורת** – ה"תקשורת" מהלקוח לשרת ובחזרה תתבצע דרך הדמיית פניות בסגנון AJAX עם פעולות REST API מתאימות ותוך אריזת המידע המועבר בפורמט JSON. לשם כך, קוד הלקוח יזום פניות מטיפוס FAJAX (Fake AJAX) שיוגדרו בפרויקט. פניות אלו "ישלחו" לשרת אשר יפענח אותן, יגיב בהתאם תוך שימוש במאגר המידע (Data Base), ו- "יחזיר" את התגובות המתאימות ללקוח.

**הגדרה מפורטת של הפרויקט השלישי – 30 דקות – יבוצע בסוף שיעור 6****כיתתי: פרויקט – מימוש מערכת end-to-end – הנחיות מפורטות:****Client** – יישום צד לקוח לגישה ושינויי מידע הנשמר אצל השרת

- **מסך יישום** – הצגת פריטים / הצגת פריט / חיפוש פריט / הוספת פריט / שינוי פריט / מחיקת פריט
- **מסך כניסה** – משתמש קיים, בדיקת טופס, וידוא פרטים (מול מאגר רישום ב- local storage)
- **מסך רישום** – משתמש חדש, בדיקת טופס, וידוא פרטים (מול מאגר רישום ב- local storage)

**SPA** – כל מרכיבי היישום בצד הלקוח טעונים במלואם בדפדפן ומוצגים בהתאמה

- **עמודים** – קוד HTML כולל חלקים להצגת עמודים שונים / קוד JS מחליף בין העמודים
- **ניתוב** – במעבר בין עמודים יש להשתמש בניתוב צד לקוח כפי שנלמד בסרטון המתאים

**Server** – תוכנה המקבלת מיישומי לקוח מתאימים פניות לגישה ושינויי מידע במאגר המידע

- **קוד השרת** – פונקציות המבוצעות כאשר "מגיעה" פניה מצד הלקוח לנהל מידע ב- DB
- קוד השרת מקבל את פרטי הפניה (JSON עם שדות מתאימים) ופועל על סמך ההנחיות בשדות
- קוד השרת פונה ל- DB לביצוע הפעולות דרך ה- DB API

**DB** – מאגר המידע – תוכנה לטיפול במידע – שמירה / חיפוש / שליפה / הוספה / שינוי / מחיקה

- **מימוש** – מאגר רשומות תחת שם מתאים בזיכרון מקומי / local storage
- כל רשומה תהיה אובייקט JSON עם השדות הרלוונטיים
- אפשרות להוספת מאגר מידע נפרד לרישום ואישור משתמשים

**DB API** – ממשק הפעולות והמידע בין השרת ובין מאגר המידע

- **פונקציות** – שליפת כל הרשומות, שליפת רשומה, הוספת רשומה, שינוי רשומה, מחיקת רשומה, ועוד
- שימוש של השרת ב- DB ייעשה רק דרך ה- DB API

**REST API** – ממשק תקשורת הפעולות והמידע בין הלקוח והשרת

- **GET** – שליפת כל הרשומות
- **GET** – שליפת רשומה מסוימת
- **POST** – הוספת רשומה חדשה
- **PUT** – עדכון רשומה קיימת
- **DELETE** – מחיקת רשומה קיימת

**JSON** – פורמט לאריזת ופתיחת תקשורת המידע בין הלקוח לשרת

- אריזת ופתיחת רשומות מידע להעברה בין הלקוח ל- DB
- ישמש אולי גם במימוש העברת ההוראות והמידע מהלקוח לשרת דרך מנגנון ה- FAJAX

**FAJAX** – מערכת הדמיית תהליכי התקשורת בין הלקוח לשרת ובחזרה

- מחלקה XMLHttpRequest עם מספר שדות ומתודות בדומה למחלקת XMLHttpRequest
- כל פעולת תקשורת מלקוח לשרת תיצור ותשתמש באובייקט חדש מטיפוס XMLHttpRequest
- בסיום הגדרת האובייקט התקשורת – הלקוח "שולח" (send) את הפנייה לשרת
- כשהשרת מסיים את הטיפול בפנייה – השרת "מחזיר" תגובה ללקוח

**דרישות מהפרויקט – 15 דקות – יבוצע בסוף שיעור 6**

כיתתי: דרישות מפרויקטים:

- עבודה בזוגות – שמירת הקוד על שני המחשבים (כל הזמן)
- הפרדה בין סוגי קבצים – קבצי HTML + קבצי CSS + קבצי JS + ועוד אם צריך
- עמוד HTML ראשי אחד – הכולל את כל הקודים של העמודים השונים
- קבצי CSS שונים – לפי הצורך לרכיבים שונים
- קבצי JS שונים – לרכיבי לקוח שונים, לקוד של השרת, לקוד של DB, למימוש FAJAX, וכדומה
- OOP – עבודה עם מחלקות / פונקציות בנאי / אובייקטים – DB, Server, FXHR, ועוד
- פונקציונליות – חשיבות ראשונה במעלה
- עיצוב המרכיבים – חשיבות פחותה
- סבבי פיתוח – מומלץ לבצע את הפרויקט בסבבים של הוספת פונקציונליות בכל סבב
- אפיון – אפיון החלקים השונים שבונים את הפרויקט והקשרים ביניהם
- תכנון – תכנון מבנה של חלקים שונים / מידע פנימי / מידע חיצוני / פעולות / ממשקים
- סדר – הקוד חייב להיות מסודר ומחולק לקבצים שונים

זוגות: צוות זוגות חדשים

הגדרת תפקידים וצורת עבודה

התייעצות עם צוות הדרכה

**בהצלחה!!!****מקורות לעיון בתכנון הפרויקט**

אישי: Mosh – וידאו – What is REST API? – להבנת הממשק והפעלים השונים

[www.youtube.com/watch?v=SLwpqD8n3d0](http://www.youtube.com/watch?v=SLwpqD8n3d0)

אישי: Traversy Media – סרטון – JSON Crash Course – חלקים רלוונטיים

[www.youtube.com/watch?v=w11CWzNtE-M](http://www.youtube.com/watch?v=w11CWzNtE-M)**שיעור 7 – 300 דקות**

זוגות: התמקדות בנושאים הבאים:

1. אפיון – הגדרת הפרויקט ושרטוט סקיצות מסכים ותרשימים
2. תכנון – תכנון החלקים השונים בפרויקט והשתלבותם ביחד
3. מחקר – טכנולוגיות נחוצות, אפשרויות שונות, קודים קיימים
4. ביצוע – מימוש חלקים שונים, שילוב החלקים לפרויקט שלם

**שיעור 8 – 300 דקות**

זוגות: התמקדות בנושאים הבאים:

3. מחקר – טכנולוגיות נחוצות, אפשרויות שונות, קודים קיימים
4. ביצוע – מימוש חלקים שונים, שילוב החלקים לפרויקט שלם
5. בדיקה – חלקים בפרויקט, תרחישים שונים, מכלול הפרויקט

**שיעור 9 – 300 דקות**

זוגות: התמקדות בנושאים הבאים:

6. הכנה – שיפורים אחרונים, תכנון הצגת הפרויקט – 150 דקות
7. הצגה – הצגה בפני המשתתפים, קבלת משובים – 150 דקות