

Full-Stack Web Development

פרק שביעי – NodeJS + Databases – יחידות 5 עד 8

הסברים והנחיות

יחידות 5 עד 8 מהוות את הפרויקט המסכם של הפרק השביעי.

פרויקט שביעי – הגדרות + דרישות

פרויקט – יישום React עם מידע דרך REST API מול שרת NodeJS + Express ומול דאטהבייס MySQL.
 טכנולוגיות לקוח – React Router // React Hooks // React Forms // JS Async-Await // JS Fetch
 טכנולוגיות שרת – MySQL Database // Express Library // NodeJS Server

הסבר כללי –

- בניית לקוח + שרת + מאגר מידע בסגנון jsonplaceholder.typicode.com שפגשתם בפרק החמישי.
- הפרויקט מחולק לשלבים לפי הרכיבים. קראו היטב את כל השלבים והשתדלו להתקדם בהדרגה.
- במהלך ובסיום כל שלב – בדקו היטב את התנהגות הרכיבים שבניתם לפני שתעברו לשלב הבא.

שלב א – בניית מאגר מידע MySQL

- בנו דאטהבייס MySQL עם פריטי מידע רק עבור המשאבים – users, todos, posts, comments.
- לכל user יהיו מספר todos ומספר posts השייכים לו, ובנוסף לכל post יהיו מספר comments.
- ניתן לצמצם לכמות סבירה את מספר פריטי המידע בדאטהבייס וגם את כמות השדות ב-users.
- המידע בשרת יישמר בטבלאות בדאטהבייס MySQL. יש לתכנן היטב את ארכיטקטורת הטבלאות.
- הוסיפו בדאטהבייס טבלה נפרדת עבור passwords של users. (תכננו להגביל גישה לטבלה זו).

שלב ב – פיתוח שרת NodeJS מול מאגר המידע ופיתוח Express עבור REST API

- בנו שרת NodeJS המתחבר לדאטהבייס של MySQL לעיל ומסוגל לבצע מולו שאילתות שונות.
- כתבו פונקציות ייעודיות לביצוע פעולות שונות על המידע בדאטהבייס ובדקו את הפונקציות שכתבתם.
- כתבו מסגרת של Express ובנו נתיבים באופן דומה לקיים באתר jsonplaceholder.typicode.com.
- ממשו פעולות REST API רלוונטיות – GET, POST, PUT, DELETE – על כל אחד מהנתיבים.
- השתמשו ב-postman כדי לבדוק גם בשרת וגם בדאטהבייס סוגים שונים של פניות לטיפול במידע.

שלב ג – פיתוח לקוח React – עמוד הכניסה + עמוד היישום

- כתבו את קוד הלקוח מחדש, למרות שכתבתם קוד דומה בפרויקט של הפרק החמישי.
- עמוד כניסה – עמוד login – כתובת העמוד צריכה להיות `/login`.
- העמוד יכלול טופס עם שני שדות – username + password. בנוסף, כפתור login.
- משתמש מורשה יהיה אחד ה-users המופיעים בדאטהבייס עם סיסמה המתאימה למשתמש.
- ניסיונות כניסה עם משתמש לא מורשה יידחו עם הודעה מתאימה, והיישום יישאר בעמוד הכניסה.
- משתמש מורשה שביצע כניסה – יישמר במערכת (LS = Local Storage) ויועבר לעמוד היישום.
- בעמוד היישום יהיו בצד או למעלה 4 כפתורים / לינקים – Info, Posts, Todos, Logout.
- לחיצה על כפתור / לינק Info – תגרום להצגת מידע אישי של המשתמש (אבל לא הסיסמה שלו).
- לחיצה על כפתור / לינק Logout תוציא את המשתמש (תמחק את ה-LS) ותחזיר לעמוד הכניסה.
- כל עמוד ביישום הלקוח יוצג עם URL פנימי אינפורמטיבי, כגון `/users/shlomo/posts`.

שלב ד – פעולות עם Todos

- לחיצה על כפתור / לינק Todos – תגרום להצגת רשימת ה-todos של המשתמש הפעיל.
- פריטים ברשימה יסודרו לפי מספר ה-id שלהם ויכללו חיווי checkbox אם בוצעו או לא.
- יישום הלקוח וקוד השרת יאפשרו הבאת (GET) מספר פריטים לפי קריטריונים ו/או שאילתות.
- יישום הלקוח וקוד השרת יאפשרו הוספת (POST) פריט חדש עם הנתונים הרלוונטיים.
- יישום הלקוח וקוד השרת יאפשרו עדכון (PUT) פריט (תוכן הפריט, מצב הביצוע, וכו').
- יישום הלקוח וקוד השרת יאפשרו מחיקת (DELETE) פריט. (נקודה למחשבה - מה זו מחיקה?).

שלב ה – פעולות עם Posts ועם Comments

- לחיצה על כפתור / לינק Posts – תגרום להצגת רשימת ה- posts של כל המשתמשים.
- פריטים ברשימה יסודרו לפי מספר ה- id שלהם, ולפי דרישה יוצגו גם ה- comments שלהם.
- יישום הלקוח וקוד השרת יאפשרו הבאת (GET) מספר פריטים / לפי קריטריונים מסוימים.
- יישום הלקוח וקוד השרת יאפשרו הוספת (POST) פריט - post או comment חדשים.
- יישום הלקוח וקוד השרת יאפשרו עדכון (PUT) פריט (תוכן post, תוכן comment), אבל זאת רק אם הפריט שייך למשתמש הפעיל.
- יישום הלקוח וקוד השרת יאפשרו מחיקת (DELETE) פריט (תוכן post, תוכן comment), אבל זאת רק אם הפריט שייך למשתמש הפעיל. (נקודה למחשבה - מה זו מחיקה?)

שלבים נוספים – פעולות מתקדמות

- צמצום פעולות הגישה מהלקוח לשרת ופעולות הגישה מהשרת לדאטהבייס.
- הוספת שאילתות מתקדמות דרך פרמטרים ב- url וטיפול בשרת ובדאטהבייס.
- תמיכה בפרמטרים לשאילתות – field=, limit=, page=, sort=, וכדומה.
- הוספת Access Token שיווצר על-ידי השרת וישלח ללקוח כדי להציג ולאמת את זהותו.
- הוספת משתמש // שינוי סיסמה למשתמש // חסימת משתמש // ניהול משתמשים.
- רישום (log) בשרת של פעולות רלוונטיות (למשל – ניסיונות כניסה, פעולות עם מידע).
- פעולות עם Albums ועם Photos – גם בדאטהבייס, גם בשרת, גם בלקוח.

דרישות מפרויקטים:

- עבודה בזוגות – שמירת הקוד באמצעות **GIT**.
- הפרדת תיקיות בין קוד שרת לקוד לקוח לקוד דאטהבייס
- הפרדה בין סוגי קבצים – קבצי HTML + קבצי CSS + קבצי JS + רכיבי React + רכיבים בשרת Node ובספריית Express + רכיבים של MySQL + מודולים שונים
- ארכיטקטורת שרת – שלוש שכבות – חלוקה לנתיבים/ לוגיקה בנתיבים שונים / פונקציות גישה לדאטהבייס
- פונקציונליות – חשיבות גבוהה
- עיצוב המרכיבים – חשיבות משנית
- סבבי פיתוח – הוספת פונקציונליות בכל סבב
- שמירת גרסאות – שימרו גרסאות קודמות של היישום
- אפיון – אפיון המסכים השונים והתרחישים השונים
- תכנון – תכנון מבנה של חלקים שונים / מידע פנימי / מידע חיצוני / פעולות / ממשקים
- סדר – הקוד חייב להיות מסודר ומחולק לקבצים שונים
- בדיקות – בדקו היטב את הפרויקט שלכם לפני הגשה
- הדגמות – תכננו היטב את הדגמת הפרויקט שלכם

בהצלחה!!!