12.04.2022

# 151026 הנדסת תכנה לאינטרנט – תרגיל 4

בס"ד

מגישות: שמחה פרנקו 209166776

עדי מלאכי יוסף 209244250

הנושאים: מבנה הפרויקט כולל ארכיטקטורת,**MVC** העלאת קבצים

# א) express + מבנה MVC + העלאת קבצים

**)1 איזה סוג אובייקט נבנה בקליינט על מנת לשלוח קבצים במסגרת רגילה type( content**

# מתאים) של שליחת נתוני טופס ?HTML

אובייקט מסוג data form multipart

# )2 איזו קטגוריה של אבטיפוס משמשת קלט\פלט (רשת, קבצים וכדומה) במסגרת ?nodejs

קטגורית אירועים. .)events(

**)3 מה הוא ערך מוסף של חבילת ?express**

הערך המוסף של Express הוא לספק כלי עבודה קטנים וחזקים עבור שרתי ,HTTP מה שהופך אותו לפתרון מצוין עבור יישומי עמוד בודד, אתרי אינטרנט, היברידיות או ממשקי

HTTP ציבוריים.

Express לא מכריח להשתמש במנוע ORM או תבנית ספציפיים. עם תמיכה בלמעלה מ14-

מנועי תבנית דרך ,Consolidate.js ניתן ליצור במהירות את המסגרת המושלמת שלך.

**)4 מה הם התפקיד והשיטה של middleware בחבילת ?express**

express היא מסגרת אינטרנט לניתוב ותיווך )middleware( שיש לה פונקציונליות מינימלית

משלה: יישום אקספרס הוא בעצם סדרה של קריאות לפונקציות של תוכנת האמצע. פונקציות middleware הן פונקציות שיש להן גישה לאובייקט הבקשה ,)req( לאובייקט התגובה )res( ולפונקציית התווך הבאה במחזור הבקשה-תגובה של היישום. פונקציית התווך

הבאה מסומנת בדרך כלל על ידי משתנה בשם .next

פונקציות של תוכנת ביניים יכולות לבצע את המשימות הבאות:

 הפעלת כל קוד.

 ביצוע שינויים באובייקטים של הבקשה והתגובה.

 סיום מחזור הבקשה-תגובה.

 לקרוא לפונקציית התווך הבאה בערימה.

אם פונקציית התווך הנוכחית אינה מסיימת את מחזור הבקשה-תגובה, עליה לקרוא ל-

)(next כדי להעביר את השליטה לפונקציית התווך הבאה. אחרת, הבקשה תישאר תלויה.

**)5 אלו תיקיות במבנה הפרויקט משמשות את החלקים של MVC – ,view ,model ו-**

**?controllers**

חלוקת התיקיות במבנה :MVC **תיקיית model** - התיקייה מייצגת את הModel שבMVC שם שמנו את כל הקבצים של

הגדרת המודלים עם חיבור לשכבת ה.DAL

**תיקיית views** - שמנו את דפי הViews שב.MVC

**תיקיית routes** - התיקייה מייצגת את הController שבMVC שם שמנו את כל הבקרים

(נתבים) של שרות האינטרנט שלנו.

**ב) עבודה עם MongoDB ומודול mongoose**

# )1 מה הם השלבים של עבודה עם בסיס נתונים מאפליקציית שרת?

התחברות למידע, הכנת האפליקציה לקבלת המידע, העברת המידע לאפליקציה, הצגת

המידע בצורה המתאימה, עדכון/מניפולציה כלשהיא, בדיקת המידע ושמירת המידע בשרת.

**?connection string זה מה )2**

בעברית מחרוזת גישה, מספקת/כוללת בתוכה פרטים על מקור המידע וסוג הקשר (בין

האפליקציה הרלוונטית אליו.)

# )3 מה הן המגבילות של MongoDB למושגים הבאים בבסיסי נתונים רלציונים: טבלה,

**רשומה, עמודה?**

טבלה – אוסף )collection(

רשומה – מסמך )document(

עמודה – שדה )period(

**)4 במה עוזרת הסכימה בmongoose- ומה הקשר שלה ל?MongoDB-**

הסכימה בmongoose- מגדירה איך יראה כל מסמך באוסף.

# ג) החלק המעשי מתבסס על תרגיל בית 4 – מוגש בקובץ זיפ

)1 יש להתקין את **MongoDB** ולהריץ **mongod** בהתאם להנחיות

)2 יש לבצע **refactoring** של כל הקוד בצד שרת ע"י הפרדה למודולים ושיבוץ בתיקיות לפי מבנה

הפרויקט כנ"ל (מודל **MVC** בצד שרת)

)3 יש לתכנן את כל האוספים ושדותיהם האפשריים על מפתחותיהם וכו' על מנת לשמור את כל

המידע בבסיס הנתונים **MongoDB**

)4 יש להוסיף תיקיית **models** למבנה הפרויקט ולהעביר למודולים בתיקיה הזו את הנתונים ואת הטיפול בהם. יש לבנות את כל המודלים בנפרד לכל אוסף. יש לחשוב ולהוסיף בחכמה

**middleware** של **mongoose** במודולים האלה והוספה\עדכון\בדיקה של מידע בכתיבה ו\-או

טעינה של נתונים. מומלץ להתבסס על הדוגמה מהמעבדה שבוצעה בכיתה.

)5 יש לבצע **refactoring** של הקוד על מנת לבסס את העבודה עם הנתונים על פעולות **CRUD**

 יש להוסיף אפשרות של הוספת מוצר ע"י משתמש מקטגוריית ספק או חלופתו, כולל העלאת

קבצים של תמונה ו\-או ספציפיקציה של המוצר או פונקציונליות דומה באפליקציה שלכם

o יש לאפשר העלאת תמונה\קובץ גם מאינטרנט וגם מהמחשב דרך הדפדפן (יש

לממש את שתי האפשרויות) בחלקים המתאימים של האתר (למשל תמונת המוצר) o על מנת להעלות קבצים מהמחשב חובה להשתמש בצד שרת במודול הרחבה **multer** של node (תמצאו ב)**npmjs.org**- או מודול הרחבה אחר לבחירתכם להעלאת קבצים בפורמט של **multipart/form-data** (זה יעשה את הקוד בדף להרבה יותר פשוט ממה

שהוצג\מופיע בהנחיות מעבדה .)4 עליכם למצוא באינטרנט דוגמאות הכנת הקובץ

לשליחה בצק הלקוח.

בהצלחה.