



בניית מערכות ממוחשבות

מבוססות WEB

פרויקט הקורס

חלק 3 – צד שרת



קבוצה 21

205632763

205611098

315390930

311459101

הקדמה

כפי שתיארנו בחלקים הקודמים, מטרת האתר שלנו היא ניהול טורנירי שחמט לעמותת "צ.ל.ש.". משתמשי האתר הם צוות העמותה ובעזרתו יוכלו לבצע מעקב אחר הטורנירים השונים ואחר הביצוע השוטף שלהם. מאחר והאתר מיועד לעמותה בישראל בלבד הוחלט ליישם את האתר בשפה העברית.

הנחות יסוד:

- השימוש במערכת נעשה גם במובייל ולכן האתר הינו אתר מותאם בו כל המסכים מותאמים לגודל של טלפון נייד.
- חשובה לנו חווית משתמש טובה ולכן לאורך כל הפרויקט מופיעות התראות למשתמש- בעת הזנת נתונים, מחיקה או כניסה לאתר.
- האתר שלנו מדמה מערכת סגורה רק לצוות העמותה ולכן לא ניתן להיכנס לאתר ללא שם משתמש וסיסמה. הרישום לאתר נעשה על ידי האדמין (שלו כבר יש שם משתמש וסיסמה) והוא רושם את שאר חברי הצוות.

יצירת בסיס נתונים

יצרנו בסיס נתונים וטבלאות חדשות בהתאם למסמך האפיון ובהתאם לצרכי האתר. (נספח- קוד

ליצירת הטבלאות והכנסת הנתונים).

בנוסף, הקוד נמצא בקובץ בשם PART3_DB בתוך הפרויקט.

• **Lookup Tables:**

LookupTypeSeason{ Season} [('סוף'), ('אמצע')]

LookupType{Type} [('אזורי'), ('ארצי')]

- **tournaments** { id, name, date, location, type, Season, details, create_date }
- **Staff** { email, first_name, last_name, phone, role, password }
- **Players** { id, first_name, last_name, level, age, school }
- **GameScores**{ id_tournament, Round, desk, id_player1, id_player2, score_player1, score_player2 }

הסברים על הטבלאות:

- שתי טבלאות **lookup** שמטרתן לוודא שמכניסים רק את הסוג והעונה שקיימים במערכת.
- **Tournaments** - טבלה שמכילה את כל הפרטים על הטורנירים. מפתח ראשי id שנוצר אוטומטית בעזרת `auto_increment`. מכיל מפתחות זרים לטבלאות `Lookup`.
- **Staff** - טבלה שמכילה את כל אנשי הצוות של העמותה. מפתח ראשי `email`.
- **Players** - טבלה שמכילה את כל שחקני השחמט. מפתח ראשי `id`.
- **GameScores** - טבלה שמכילה את תוצאות המשחקים. המפתח הראשי בנוי משלושה שדות: `(id_tournament, Round, desk)` בנוסף קיימים 3 מפתחות זרים: מזהה הטורניר `(id_tournament)` ומזהים של שני השחקנים `(id_player1/2)`. הטבלה מתמלאת לאחר ביצוע שאילתה וחישוב התוצאות בהתאם להזנת תוצאות המשחק (תיקו/ שחקן 1 ניצח / שחקן 2 ניצח) בדף השיפוט.

יצירת מחלקות

בפרויקט יצרנו 3 מחלקות פונקציונליות שמרכזות בתוכן פקודות שונות להרצת שאילתות.

מחלקת **Players**:

המחלקה כוללת את הפונקציות הבאות:

- **get_players()** - מחזירה את כל המידע על השחקנים
- **get_players_id()** - מחזירה את רשימת ה-id של השחקנים
- **insert_player()** - מכניסה שחקן חדש לDB
- **delete_player()** - מוחקת שחקן מסוים מהDB
- **update_player()** - מעדכנת ערכים מסוימים בשחקן מסוים בDB

מחלקת **Tournaments**:

המחלקה כוללת את הפונקציות הבאות:

- **get_all_tournaments()** - מחזירה את כל המידע על הטורנירים במערכת
- **get_tournament()** - מחזיר את כל המידע על טורניר ספציפי (לפי ה-id)
- **create_tournament()** - מכניסה טורניר חדש לDB
- **get_results()** - מחזירה את תוצאות הסופיות של הטורניר (שאילתה חישובית)

- **get_scores()** - מחזירה את כל הניקוד שניתן באותו טורניר (שולחן, סבב, שני שחקנים והניקוד לכל שחקן)
- **insert_scores()** - מכניסה את הניקוד שניתן בסבב ספציפי לשני שחקנים.
- **delete_scores()** - מוחקת רשומה של ניקוד בסבב ספציפי.
- **get_ranked_tables()** - מחזירה את כמות השולחנות שנוקדו בטורניר ספציפי
- **get_num_of_rounds()** - מחזירה את מספר הסבבים שבוצעו בטורניר ספציפי
- **get_num_of_players()** - מחזירה את מספר השחקנים בטורניר
- **update_tournament ()** – מעדכנת ערכים מסוימים בטורניר מסוים בDB

מחלקת Staff:

המחלקה כוללת את הפונקציות הבאות:

- **get_staff()** – מחזירה את כל המידע על אנשי הצוות
- **get_staff_email()** – מחזירה את רשימת emails של אנשי הצוות
- **get_staff_password()** – מחזירה את הסיסמה של איש צוות ספציפי
- **get_staff_name()** – מחזירה את השם הפרטי של איש צוות ספציפי
- **insert_staff()** – מכניסה איש צוות חדש לDB
- **delete_staff()** – מוחקת איש צוות מסוים מהDB
- **update_staff()** – מעדכנת ערכים מסוימים באיש צוות מסוים בDB

שאלות SQL

הפרויקט מכיל שאלות SQL שמבצעות שליפה, עדכון ומחיקת רשומות. בחלק זה נפרט את העיקריות שבהן.

1. **שאלת SELECT** שמחזירה את תוצאות השחקנים. השאלתה לוקחת את פרטי השחקן מטבלת Players, את תוצאות הסבבים של אותו השחקן מטבלת gamrscores, סוכמת את כלל הניקוד ומציגה את התוצאה הסופית בדף התוצאות. (צילום המסך שמצורף צולם console_dbt לצורך נראות והבנת השאילה בדו"ח בלבד, כל השאלות בפרויקט נכתבו בקבצי ה-Python)

```
select id_player, concat(P.first_name," ", P.last_name) AS name, score, ROW_NUMBER() OVER(ORDER BY score desc) RowNumber
from
  (select id_player1 as id_player, sum(score_player1) as score
   from(
     (select id_player1,score_player1 from gamescores
      where id_tournament=%)
     union all
     (select id_player2,score_player2
      from gamescores where id_tournament=%)
   ) as scores
   group by id_player1) as S
join Players AS P ON S.id_player = P.id
order by Score desc;
'%' (tournament_id, tournament_id)
```

2. **שאלות SELECT** לעמוד טורניר מסוים שמציגות את מספר השולחנות שנוקדו, את מספר הסיבובים שבוצעו ואת מספר השחקנים בטורניר.

```
'select count(*) as tables from GameScores where id_tournament=%s;'
```

```
'select count(distinct round) as rounds from GameScores where id_tournament=%s;'
```

```
'select count(distinct player) as players from ((select id_player1 as player from GameScores where id_tournament=%s
```

3. **שאלת SELECT** שמחזירה את שם המשתמש בהתאם לאימייל שהזין בכניסה למערכת.

```
query = "select password from staff where email='%s';"
```

4. **שאלת DELETE** בעמוד שיפוט.

מחיקת תוצאת סבב שהוזנה. במידה והתוצאה של הסבב שהוזנה לא נכונה ניתן למחוק אותה מהDB.

```
'DELETE FROM GameScores WHERE id_tournament="%s" and Round="%s" and desk="%s";'
```

5. **שאלות UPDATE** לעדכון פרטי שחקן/טורניר/צוות. נדגים שאילתה זו עבור עדכון פרטי טורניר.

בטופס העדכון ניתן לעדכן רק חלק מהנתונים ולכן בעת העדכון בבסיס הנתונים בדקנו כל שדה בנפרד. בעזרת לולאת for, עברנו על כל השדות האפשריים לעדכון בטבלה (fields) ועל הערכים שהתקבלו מה-form (result). אם התקבל ערך, כלומר השדה לא ריק, שלחנו אותו ואת השדה לעדכון לפונקציית ה-update, המורכבת מהפרמטרים הנ"ל.

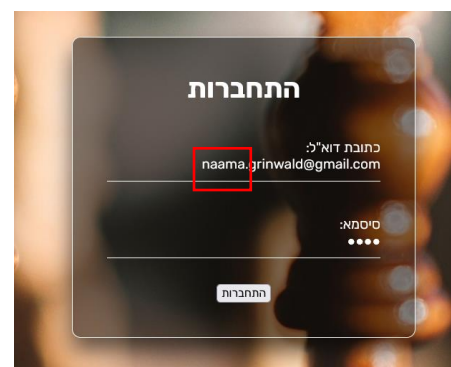
```
# update
fields=['name', 'date', 'location', 'type', 'Season', 'details']
results=[name, date, location, type, Season, details]
zip_object = zip(fields, results)
for field, result in zip_object:
    if result != "":
        Tournaments.update_tournament(field, result, tournament_id)

def update_tournament(self, field, result, tournament_id):
    query = "UPDATE tournaments SET %s='%s' WHERE id='%s';" % (field, result, tournament_id)
    return dbManager.commit(query, (tournament_id))
```

תוכן דינאמי- מימוש עמודי התבנית

את דפי ה-HTML שיצרנו בחלק ב' הפכנו לעמודים דינאמיים. למשל, פרטי הטורניר שמוצגים בדף נשלפים מתוך ה-DB. כלל הטבלאות שמוצגות בעמודים השונים נלקחים מה-DB ומתעדכנים ברגע של שינוי/עדכון בנתונים.

בנוסף, השתמשנו במשתנים גלובליים בעזרת Session להצגת שם המשתמש שהתחבר למערכת. ברגע שמשתמש מתחבר למערכת עם כתובת האימייל שלו והסיסמה, נשמר השם הפרטי לטובת הצגה בכל העמודים השונים.



דפים באתר:

- **דף התחברות –** הדף הראשון אליו מגיעים. ללא שם משתמש וסיסמה לא ניתן להיכנס למערכת. זוהי מערכת סגורה רק למשתמשים רשומים. (לכן האדמין יוצר לאנשי הצוות את שם המשתמש)
- **עמוד הבית- טורנירים-** בעמוד זה ניתן לנווט בין הטורנירים השונים הקיימים במערכת. בנוסף ניתן להגיע **לדף ליצירת טורניר חדש** בעזרת לחיצת כפתור.
- **צוות-** מטרת העמוד היא ניהול הצוות. ניתן לצפות ברשומות הקיימות, לערוך ולמחוק אותן במידת הצורך. בנוסף ניתן להגיע **לדף ליצירת איש צוות חדש** בעזרת לחיצת כפתור.
- **שחקנים-** מטרת העמוד היא ניהול השחקנים. ניתן לצפות ברשומות הקיימות, לערוך ולמחוק אותן במידת הצורך. בנוסף ניתן להגיע **לדף ליצירת שחקן חדש** בעזרת לחיצת כפתור.
- **שיפוט-** לדף זה מגיעים מתוך טורניר ספציפי. מטרת הדף היא הזנת תוצאת הסבב בין שני שחקנים.
- **תוצאות-** לדף זה מגיעים מתוך טורניר ספציפי. הדף מציג את תוצאות הטורניר.

עיצוב האתר

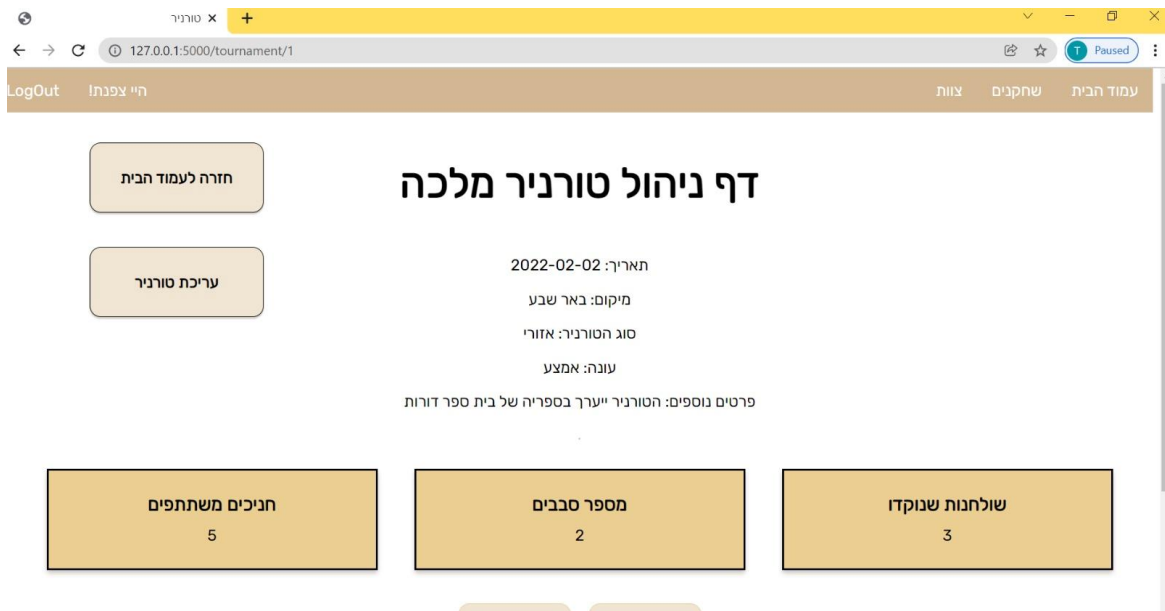
Base & components:

בפרויקט הגדרנו base עליו מבוססים כל הדפים האחרים. (בעזרת extend)
בנוסף הגדרנו שלוש קומפוננטות שמתווספות לעמודי התוכן: headline, footer, navigation bar.

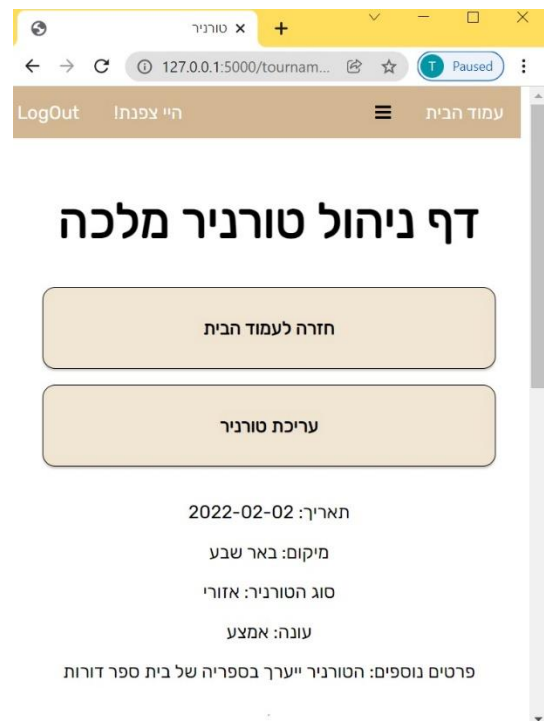
מימוש אתר רספונסיבי:

האתר מותאם גם למובייל. ההתאמה באה לידי ביטוי ב-CSS וב-JS (לטובת navbar).

גודל דף רגיל:



אתר מותאם:



מימוש הטפסים:

כלל הטפסים מומשו בעזרת תכונות הaction וה- Method המתאימות. בכל הטפסים יש ולידציות להכנסת הנתונים. בכל מקום שהשדה הוא שדה חובה, לא ניתן לשלוח את הטופס ללא הכנסה של נתונים (required). מעבר לכך בהתאם לכל טופס הכנסו ולידציות נוספות שקשורות שסוג המידע שמוכנס. על כל ולידציה נפרט בטופס הרלוונטי.

- **מסך כניסה לאתר:** בשלב הזה נדרש המשתמש להכניס שם משתמש וסיסמה על מנת להיכנס למערכת. קיימת ולידציה של הכנסת מייל תקין.
- **הרשמת משתמש לאתר Create Staff :** בהתאמה לדרישת האפיון של האתר, יצרנו טופס רישום של איש צוות חדש לפני ההתחברות. את איש הצוות החדש ניתן ליצור רק לאחר חיבור למערכת מיוזר אחר, כלומר האדמין הוא זה שיוצר משתמש חדש לאתר. בטופס קיימות ולידציות על שדה האימייל ועל שדה מספר הפלאפון.
- **יצירת שחקן חדש Create Player:** בטופס קיימות ולידציות ל'id, לגיל ולרמה.
- **יצירת טורניר :** טופס ליצירת טורניר חדש. בטופס קיימת ולידציה להכנסת תאריך- לא ניתן להכניס ערך שאינו תאריך. בנוסף על מנת להזין עונה וסוג ניתן לבחור רק מרשימה מוצגת (ללא הזנה חופשית)
- **דף שיפוט :** דף בו השופט יכול להזין שני שחקנים ואת תוצאת הסבב.

נספח: יצירת DB וטבלאות והכנסת רשומות:

```
# DB
create schema web_project_g21;
use web_project_g21;

# Lookup Tables
CREATE TABLE LookupTypeSeason(
    Season varchar(20) not null primary key);
INSERT INTO LookupTypeSeason(Season) VALUES ('אמצע'), ('סוף');

CREATE TABLE LookupType(
    Type varchar(20) not null primary key);
INSERT INTO LookupType(Type) VALUES ('אזורי'), ('ארצי');

# Tournaments Table
CREATE TABLE tournaments (
    id int auto_increment primary key,
    name varchar(255) not null ,
    date date not null ,
    location varchar(255) not null ,
    type varchar(20) not null ,
    Season varchar(20) not null ,
    details varchar(255) not null ,
    create_date timestamp default CURRENT_TIMESTAMP,

    constraint FK_type FOREIGN KEY (type) REFERENCES LookupType (type),
    constraint FK_Season FOREIGN KEY (Season) REFERENCES LookupTypeSeason
    (Season)
);

INSERT INTO tournaments(name,date,location,type,Season,details) VALUES
('מלכה','2022-02-02','שבוע באר','אזורי','אמצע','הטורניר של יעריך הטורניר,
(דורות ספר בית)');
INSERT INTO tournaments(name,date,location,type,Season,details) VALUES
('פרטים','2022-04-01','ירושלים','ארצי','סוף','פרטים');
INSERT INTO tournaments(name,date,location,type,Season,details) VALUES
('פרטים','2022-05-01','אביב תל','אזורי','אמצע','פרטים');
INSERT INTO tournaments(name,date,location,type,Season,details) VALUES
('רץ','2022-04-10','חיפה','ארצי','אמצע','פרטים');
INSERT INTO tournaments(name,date,location,type,Season,details) VALUES
('סוס','2022-03-06','שבוע באר','אזורי','סוף','פרטים');

# Staff Table
create table Staff (
    email varchar(100) not null,
    first_name varchar(100) not null,
    last_name varchar(100) not null,
    phone varchar(20) null,
    role varchar(100) not null,
    password varchar(100) not null,

    constraint Staff_pk
        primary key (email)
);
```

```

INSERT INTO staff(email, first_name, last_name, phone, role, password)
VALUES ('naama.grinwald@gmail.com', 'נעמה', 'גרינוולד', 0505917191,
'מדריך', '1234');
INSERT INTO staff(email, first_name, last_name, phone, role, password)
VALUES ('ori@gmail.com', 'אורי', 'חן', 0503425674, 'מדריך', '2222');
INSERT INTO staff(email, first_name, last_name, phone, role, password)
VALUES ('dana12@gmail.com', 'דנה', 'אליאס', 0525936191, 'שופט',
'4567');
INSERT INTO staff(email, first_name, last_name, phone, role, password)
VALUES ('tzofnat@gmail.com', 'צפנת', 'כהן', 0505922191, 'מדריך',
'7777');
INSERT INTO staff(email, first_name, last_name, phone, role, password)
VALUES ('niv@gmail.com', 'ניב', 'גרינברג', 0505556743, 'שופט', '8888');
INSERT INTO staff(email, first_name, last_name, phone, role, password)
VALUES ('tal@gmail.com', 'טל', 'לוי', 0543267894, 'מדריך', '4901');

# Players Table
create table Players (
    id int not null,
    first_name varchar(100) not null,
    last_name varchar(100) not null,
    level int not null,
    age int not null,
    school varchar(100) not null,

    constraint Players_pk
        primary key (id)
);

INSERT INTO Players(id, first_name, last_name, level, age, school)
VALUES (1, 'אלון', 'ברינה', 2, 10, 'עמית');
INSERT INTO Players(id, first_name, last_name, level, age, school)
VALUES (2, 'יעל', 'אליאס', 3, 11, 'ניצנים');
INSERT INTO Players(id, first_name, last_name, level, age, school)
VALUES (3, 'ספיר', 'כהן', 1, 8, 'אריאל');
INSERT INTO Players(id, first_name, last_name, level, age, school)
VALUES (4, 'בוטז', 'לוי', 3, 11, 'דורות');
INSERT INTO Players(id, first_name, last_name, level, age, school)
VALUES (5, 'אופיר', 'חן', 3, 9, 'אריאל');

# Game Scores Table
CREATE TABLE GameScores(
    id_tournament int not null,
    Round int not null,
    desk int not null,
    id_player1 int not null,
    id_player2 int not null,
    score_player1 float not null,
    score_player2 float not null,

    constraint PK_tournments PRIMARY KEY (id_tournament, Round, desk),
    constraint FK_tournament FOREIGN KEY (id_tournament) REFERENCES
tournaments (id),
    constraint FK_id_player1 FOREIGN KEY (id_player1) REFERENCES

```

```
Players (id),
    constraint FK_id_player2 FOREIGN KEY (id_player2) REFERENCES
Players (id)
);

INSERT INTO
GameScores(id_tournament, Round, desk, id_player1, id_player2, score_player1
, score_player2) VALUES (1, 1, 4, 1, 2, 0.5, 0.5);
INSERT INTO
GameScores(id_tournament, Round, desk, id_player1, id_player2, score_player1
, score_player2) VALUES (1, 1, 5, 3, 4, 1, 0);
INSERT INTO
GameScores(id_tournament, Round, desk, id_player1, id_player2, score_player1
, score_player2) VALUES (1, 2, 10, 4, 5, 0.5, 0.5);
INSERT INTO
GameScores(id_tournament, Round, desk, id_player1, id_player2, score_player1
, score_player2) VALUES (2, 1, 7, 1, 2, 0, 1);
INSERT INTO
GameScores(id_tournament, Round, desk, id_player1, id_player2, score_player1
, score_player2) VALUES (2, 2, 8, 3, 5, 0.5, 0.5);
INSERT INTO
GameScores(id_tournament, Round, desk, id_player1, id_player2, score_player1
, score_player2) VALUES (3, 1, 4, 1, 2, 1, 0);

#####
```