

İTÜ



250 YIL
1773 – 2023

Database Systems

BLG 317E

Term Project Report ONLYF1S

Mustafa Can Çalışkan - caliskanmu20@itu.edu.tr

Emircan Erol - erole20@itu.edu.tr

Yusuf Şahin - sahiny20@itu.edu.tr

Yusuf Emir Sezgin - sezginy20@itu.edu.tr

Faculty of Computer and Informatics Engineering
Department of Computer Engineering

CONTENTS

1	Introduction	3
1.1	Introduction	3
1.2	Application Overview	3
2	Database	4
2.1	Entity Relation Diagram (ERD).....	4
2.2	Database Schema.....	5
3	Implementation	16
3.1	Database Management System and General Tools	16
3.2	Dockerizing	16
3.3	BULK - CUD Operations	17
3.4	Before Page Implementations	18
3.4.1	Generalization	18
3.4.2	SQL Injection.....	18
3.4.3	Authentication and Authorization.....	18
3.5	Page Implementations	18
3.5.1	Drivers.....	19
3.5.1.1	CRUD on Drivers Page.....	19
3.5.1.2	Driver Details.....	22
3.5.2	Circuits	23
3.5.3	Constructors	23
3.5.3.1	Constructor Details	24
3.5.4	Seasons	25
3.5.5	Others	25
3.5.6	Quiz	26
3.5.6.1	CRUD Operations On Quiz Page	29
3.5.7	Blink Game	30
3.5.8	Race Results	31
3.5.9	Rankings.....	33
3.5.9.1	A Ranking Example.....	33
3.5.10	User	34
3.5.11	Contact Us	36

4 Conclusion	38
4.1 Discussions	38
4.2 Conclusion	38

1. Introduction

1.1. Introduction

This project assembles a Term Project for this course. In this project, we created a Formula 1 fandom website named "ONLYF1S" using the dataset that includes years between 1953–2023 data about Formula 1 Sports. We worked on the CRUD operations, dockerizing, rabbitMQ, generic page creation, operations, authentication, and authorization.

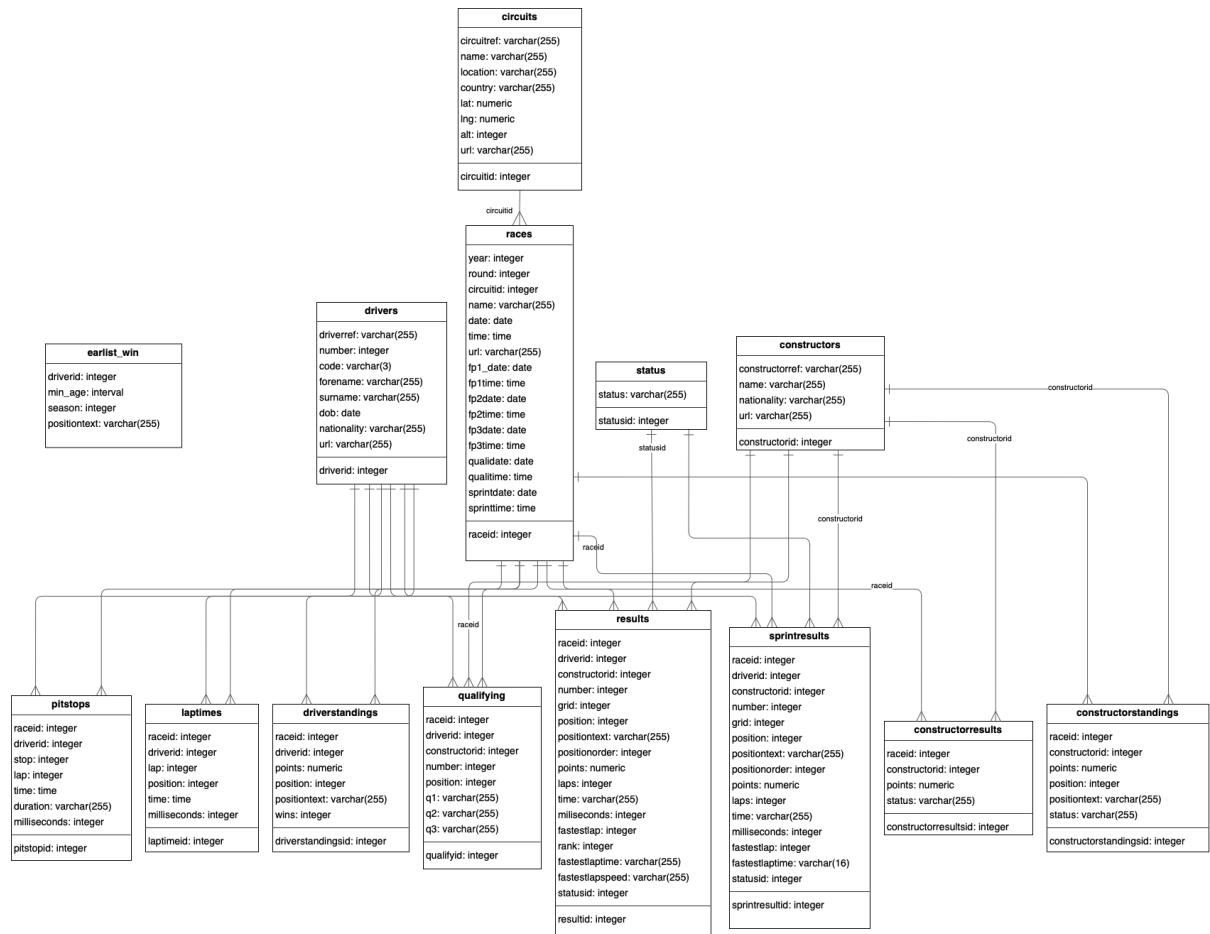
1.2. Application Overview

All users can find any drivers, constructors, circuits, etc. on their unique page. Users can search and find the specific data on every page. Users can try themselves on Quiz and F1 Blink Game. However, users that have authorization as admin can create new data, update existing data, and remove specific data as extra of the regular users' abilities to do on the website.

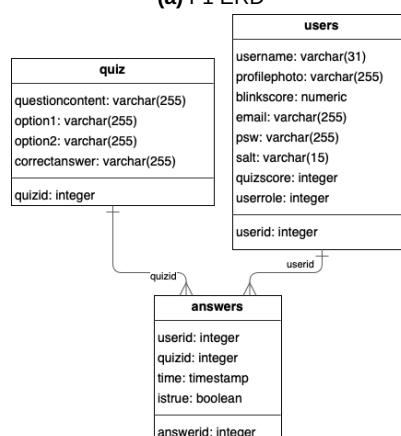
2. Database

2.1. Entity Relation Diagram (ERD)

We used Formula 1 World Championship Data set (1950 - 2023). Also, we created user, answers, and quiz tables.



(a) F1 ERD



(b) User ERD

2.2. Database Schema

Database SQL schema is as follows:

```
1  driverId INT GENERATED ALWAYS AS IDENTITY (START WITH 860),
2  driverRef VARCHAR(255),
3  number INT,
4  code VARCHAR(3),
5  forename VARCHAR(255),
6  surname VARCHAR(255),
7  dob date,
8  nationality VARCHAR(255),
9  url VARCHAR(255),
10 PRIMARY KEY (driverId)
11 );
12
13 CREATE TABLE IF NOT EXISTS constructors(
14   constructorId INT GENERATED ALWAYS AS IDENTITY (START WITH 215)
15   ,
16   constructorRef VARCHAR(255),
17   name VARCHAR(255),
18   nationality VARCHAR(255),
19   url VARCHAR(255),
20   PRIMARY KEY(constructorId)
21 );
22
23 CREATE TABLE IF NOT EXISTS circuits(
24   circuitId INT GENERATED ALWAYS AS IDENTITY (START WITH 80),
25   circuitRef VARCHAR(255),
26   name VARCHAR(255),
27   location VARCHAR(255),
28   country VARCHAR(255),
29   lat DECIMAL,
30   lng DECIMAL,
31   alt INT,
32   url VARCHAR(255),
33   PRIMARY KEY(circuitId)
34 );
35 -- We've changed the primary key constraint, ERD was wrong in terms
36 -- of logic.
37 -- Year primary key constraint has been deleted.
38 CREATE TABLE IF NOT EXISTS races(
```

```

39     raceId INT GENERATED ALWAYS AS IDENTITY (START WITH 1121),
40     year INT,
41     round INT,
42     circuitId INT,
43     name VARCHAR(255),
44     date DATE,
45     time TIME,
46     url VARCHAR(255),
47     fp1date DATE,
48     fp1time TIME,
49     fp2date DATE,
50     fp2time TIME,
51     fp3date DATE,
52     fp3time TIME,
53     qualiDate DATE,
54     qualiTime TIME,
55     sprintDate DATE,
56     sprintTime TIME,
57
58 PRIMARY KEY(raceId),
59 CONSTRAINT fkCircuitsRaces
60     FOREIGN KEY(circuitId) REFERENCES circuits(circuitId)
61     ON DELETE CASCADE
62     ON UPDATE CASCADE
63
64 );
65
66 CREATE TABLE IF NOT EXISTS qualifying (
67     qualifyId INT GENERATED ALWAYS AS IDENTITY (START WITH 10033),
68     raceId INT,
69     driverId INT,
70     constructorId INT,
71     number INT,
72     position INT,
73     q1 VARCHAR(255),
74     q2 VARCHAR(255),
75     q3 VARCHAR(255),
76
77 PRIMARY KEY (qualifyId),
78
79 CONSTRAINT fkQualifyingRaces
80     FOREIGN KEY(raceId) REFERENCES races(raceId)
81     ON DELETE CASCADE

```

```

82     ON UPDATE CASCADE ,
83
84     CONSTRAINT fkQualifyingDrivers
85         FOREIGN KEY(driverId) REFERENCES drivers(driverId)
86         ON DELETE CASCADE
87         ON UPDATE CASCADE ,
88
89     CONSTRAINT fkQualifyingConstructors
90         FOREIGN KEY (constructorId) REFERENCES constructors (
91             constructorId)
92         ON DELETE CASCADE
93         ON UPDATE CASCADE
94 );
95
96 CREATE TABLE IF NOT EXISTS status (
97     statusId INT GENERATED ALWAYS AS IDENTITY (START WITH 142),
98     status VARCHAR(255),
99     PRIMARY KEY(statusId)
100 );
101
102 CREATE TABLE IF NOT EXISTS sprintResults(
103     sprintResultId INT GENERATED ALWAYS AS IDENTITY (START WITH
104         241),
105     raceId INT,
106     driverId INT,
107     constructorId INT,
108     number INT,
109     grid INT,
110     position INT,
111     positionText VARCHAR(255),
112     positionOrder INT,
113     points DECIMAL,
114     laps INT,
115     time VARCHAR(255),
116     milliseconds INT,
117     fastestLap INT,
118     fastestLapTime VARCHAR(16),
119     statusID INT,
120     PRIMARY KEY(sprintResultId),
121     CONSTRAINT fkRacesSprintResults
122         FOREIGN KEY(raceID) REFERENCES races(raceId)

```

```

123     ON DELETE CASCADE
124     ON UPDATE CASCADE ,
125
126     CONSTRAINT fkDriversSprintResults
127         FOREIGN KEY(driverId) REFERENCES drivers(driverId)
128         ON DELETE CASCADE
129         ON UPDATE CASCADE ,
130
131     CONSTRAINT fkConstructorsSprintResults
132         FOREIGN KEY(constructorId) REFERENCES constructors(
133             constructorId)
134         ON DELETE CASCADE
135         ON UPDATE CASCADE ,
136
137     CONSTRAINT fkStatusSprintResults
138         FOREIGN KEY(statusId) REFERENCES status(statusId)
139         ON DELETE CASCADE
140         ON UPDATE CASCADE
141 );
142 CREATE TABLE IF NOT EXISTS results(
143     resultId INT GENERATED ALWAYS AS IDENTITY (START WITH 26246),
144     raceId INT ,
145     driverId INT ,
146     constructorId INT ,
147     number INT ,
148     grid INT ,
149     position INT ,
150     positionText VARCHAR(255) ,
151     positionOrder INT ,
152     points DECIMAL ,
153     laps INT ,
154     time VARCHAR(255) ,
155     miliseconds INT ,
156     fastestLap INT ,
157     rank INT ,
158     fastestLapTime VARCHAR(255) ,
159     fastestLapSpeed VARCHAR(255) ,
160     statusId INT ,
161
162     PRIMARY KEY(resultId),
163
164     CONSTRAINT fkRacesResults

```

```

165     FOREIGN KEY(raceId) REFERENCES races(raceId)
166     ON DELETE CASCADE
167     ON UPDATE CASCADE ,
168
169     CONSTRAINT fkDriverResults
170         FOREIGN KEY(driverId) REFERENCES drivers(driverId)
171         ON DELETE CASCADE
172         ON UPDATE CASCADE ,
173
174     CONSTRAINT fkConstructorsResults
175         FOREIGN KEY(constructorId) REFERENCES constructors(
176             constructorId)
177         ON DELETE CASCADE
178         ON UPDATE CASCADE ,
179
180     CONSTRAINT FkStatusResults
181         FOREIGN KEY(statusId) REFERENCES status(statusId)
182         ON DELETE CASCADE
183         ON UPDATE CASCADE
184 );
185
186 CREATE TABLE IF NOT EXISTS pitStops (
187     pitstopid INT GENERATED ALWAYS AS IDENTITY (START WITH 10479),
188     raceId INT ,
189     driverId INT ,
190     stop INT ,
191     lap INT ,
192     time TIME ,
193     duration VARCHAR(255) ,
194     milliseconds INT ,
195
196     PRIMARY KEY(pitstopid),
197
198     CONSTRAINT fkRacesPitStops
199         FOREIGN KEY(raceId) REFERENCES races(raceId)
200         ON DELETE CASCADE
201         ON UPDATE CASCADE ,
202
203     CONSTRAINT fkDriversPitStops
204         FOREIGN KEY(driverId) REFERENCES drivers(driverId)
205         ON DELETE CASCADE
206         ON UPDATE CASCADE
207 );

```

```

207
208 CREATE TABLE IF NOT EXISTS lapTimes(
209     laptimeid INT GENERATED ALWAYS AS IDENTITY (START WITH 560408),
210     raceId INT,
211     driverId INT,
212     lap INT,
213     position INT,
214     time TIME,
215     milliseconds INT,
216
217     PRIMARY KEY(laptimeid),
218
219     CONSTRAINT fkRacesLapTimes
220         FOREIGN KEY(raceId) REFERENCES races(raceId)
221         ON DELETE CASCADE
222         ON UPDATE CASCADE,
223
224     CONSTRAINT fkDriversLapTimes
225         FOREIGN KEY(driverId) REFERENCES drivers(driverId)
226         ON DELETE CASCADE
227         ON UPDATE CASCADE
228 );
229
230 CREATE TABLE IF NOT EXISTS driverStandings (
231     driverStandingsId INT GENERATED ALWAYS AS IDENTITY (START WITH
232         72452),
233     raceId INT,
234     driverId INT,
235     points DECIMAL,
236     position INT,
237     positionText VARCHAR(255),
238     wins INT,
239
240     PRIMARY KEY (driverStandingsId),
241
242     CONSTRAINT fkRacesDriverStandings
243         FOREIGN KEY(raceId) REFERENCES races(raceId)
244         ON DELETE CASCADE
245         ON UPDATE CASCADE,
246
247     CONSTRAINT fkDriversDriverStandings
248         FOREIGN KEY(driverId) REFERENCES drivers(driverId)
249         ON DELETE CASCADE

```

```

249     ON UPDATE CASCADE
250 );
251
252 CREATE TABLE IF NOT EXISTS constructorStandings(
253     constructorStandingsId INT GENERATED ALWAYS AS IDENTITY (START
254         WITH 28693),
255     raceId INT,
256     constructorId INT,
257     points DECIMAL,
258     position INT,
259     positionText VARCHAR(255),
260     status VARCHAR(255),
261
262     PRIMARY KEY(constructorStandingsId),
263
264     CONSTRAINT fkRacesConstructorStandings
265         FOREIGN KEY(raceId) REFERENCES races(raceId)
266         ON DELETE CASCADE
267         ON UPDATE CASCADE,
268
269     CONSTRAINT fkConstructorsConstructorStandings
270         FOREIGN KEY(constructorId) REFERENCES constructors(
271             constructorId)
272         ON DELETE CASCADE
273         ON UPDATE CASCADE
274 );
275
276 CREATE TABLE IF NOT EXISTS constructorResults(
277     constructorResultsId INT GENERATED ALWAYS AS IDENTITY (START
278         WITH 16870),
279     raceId INT,
280     constructorId INT,
281     points DECIMAL,
282     status VARCHAR(255),
283
284     PRIMARY KEY(constructorResultsId),
285
286     CONSTRAINT fkRacesConstructorResults
287         FOREIGN KEY(raceId) REFERENCES races(raceId)
288         ON DELETE CASCADE
289         ON UPDATE CASCADE,
290
291     CONSTRAINT fkConstructorsConstructorResults

```

```

289     FOREIGN KEY(constructorId) REFERENCES constructors(
290         constructorId)
291         ON DELETE CASCADE
292         ON UPDATE CASCADE
293     );
294
295 CREATE TABLE IF NOT EXISTS quiz (
296     quizId INT GENERATED ALWAYS AS IDENTITY (START WITH 2),
297     questionContent VARCHAR(255),
298     option1 VARCHAR(255),
299     option2 VARCHAR(255),
300     correctAnswer VARCHAR(255),
301
302     PRIMARY KEY (quizId)
303 );
304
305 CREATE TABLE IF NOT EXISTS users(
306     userId INT GENERATED ALWAYS AS IDENTITY ,
307     username VARCHAR(31) UNIQUE ,
308     profilePhoto VARCHAR(255) UNIQUE ,
309     blinkScore DECIMAL ,
310     email VARCHAR(255) UNIQUE ,
311     psw VARCHAR(255) ,
312     salt VARCHAR(15) ,
313     quizScore INT ,
314     userRole INT ,
315
316     PRIMARY KEY(userId) ,
317     CONSTRAINT usernameEmailNotNull CHECK (
318         NOT (
319             ( username IS NULL OR username = '' )
320             AND
321             ( email IS NULL OR email = '' )
322         )
323     ),
324     CONSTRAINT usersIntegrity CHECK (
325         quizScore > 0 AND
326         blinkScore > 0 AND
327         userRole = 0 OR userRole = 1
328     )
329 );
330

```

```

331 CREATE TABLE IF NOT EXISTS answers(
332     answerId INT GENERATED ALWAYS AS IDENTITY ,
333     userId INT ,
334     quizId INT ,
335     time TIMESTAMP ,
336     isTrue BOOLEAN ,
337
338     PRIMARY KEY (answerId) ,
339     UNIQUE (userId, quizId, time) ,
340
341     CONSTRAINT fkUserAnswers
342         FOREIGN KEY(userId) REFERENCES users(userId)
343         ON DELETE CASCADE
344         ON UPDATE CASCADE ,
345
346     CONSTRAINT fkQuizAnswers
347         FOREIGN KEY(quizId) REFERENCES quiz(quizId)
348         ON DELETE CASCADE
349         ON UPDATE CASCADE
350 );
351
352 CREATE TABLE IF NOT EXISTS lkp_tables(
353     tableId INT GENERATED ALWAYS AS IDENTITY ,
354     tableName varchar(30) ,
355
356     PRIMARY KEY (tableId)
357 );
358
359 INSERT INTO lkp_tables (tableName)
360 SELECT table_name
361 FROM information_schema.tables
362 WHERE table_schema = 'public' AND table_name != 'lkp_tables';
363
364 CREATE VIEW EARLIST_WIN AS
365 SELECT re.driverid,
366     MIN(AGE(r.date, d.dob)) as min_age ,
367     MIN(r.year) as season ,
368     re.positiontext
369     FROM races r JOIN results re ON re.raceid = r.raceid
370     JOIN drivers d ON d.driverid = re.driverid
371     WHERE re.positiontext = '1'
372     GROUP BY re.driverid, re.positiontext;
373

```

```

374 CREATE VIEW PER_WINS AS
375 SELECT d.forename, d.surname,
376 (SELECT CONCAT(MIN(r.year), ' - ', MAX(r.year))
377 FROM races r JOIN results re ON re.raceid = r.raceid
378 WHERE re.positiontext = '1' AND re.driverid = d.driverid) AS
379 seasons,
380 COUNT(re.driverid) AS ENTRY,
381 COUNT(CASE WHEN re.positiontext = '1' THEN 1 ELSE NULL END) AS
382 total_wins,
383 ROUND((COUNT(CASE WHEN re.positiontext = '1' THEN 1 ELSE NULL END)
384 ::NUMERIC / COUNT(re.driverid))::NUMERIC * 100, 2) AS win_rate
385 FROM drivers d JOIN results re ON d.driverid = re.driverid
386 GROUP BY d.forename, d.surname, d.driverid
387 ORDER BY win_rate
388 DESC;
389
390
391
392
393
394
395
396 ) AS $$

397 BEGIN
398 RETURN QUERY
399 SELECT d.forename,
400 d.surname,
401 a.max_age AS age,
402 r.name,
403 a.season,
404 r.round,
405 re.positiontext AS result
406 FROM races r JOIN results re ON re.raceid = r.raceid
407 JOIN drivers d ON d.driverid = re.driverid
408 JOIN status s ON re.statusid = s.statusid
409 JOIN (SELECT re.driverid,
410 MAX(AGE(r.date, d.dob)) AS max_age,
411 MIN(r.year) AS season,
412 s.statusid
413 FROM races r JOIN results re ON re.raceid = r.raceid

```

```

414     JOIN drivers d ON d.driverid = re.driverid
415     JOIN status s ON re.statusid = s.statusid WHERE s.statusid = 1
416     GROUP BY re.driverid, s.statusid
417 ) a ON d.driverid = a.driverid
418 AND AGE(r.date, d.dob) = a.max_age AND s.statusid = a.statusid
419 ORDER BY age DESC
420 LIMIT slimNumber;
421 END; $$ LANGUAGE plpgsql;
422
423 CREATE FUNCTION user_insertion_logger() RETURNS TRIGGER AS $$%
424 BEGIN
425     RAISE LOG 'A new row with id % and username % has been inserted'
426             , NEW.userId, NEW.username;
427     RETURN NEW;
428 END;
429 $$ LANGUAGE plpgsql;
430
431 CREATE TRIGGER user_logger
432 AFTER INSERT ON users
433 FOR EACH ROW
434 EXECUTE PROCEDURE user_insertion_logger();

```

3. Implementation

3.1. Database Management System and General Tools

In our project, we utilized Flask [4], the web development framework of Python, alongside PostgreSQL [5] as the database management system. We supported the site's appearance using HTML, CSS, and JavaScript codes.

3.2. Dockerizing

To ensure consistent environmental setup and overcome local constraints, we leverage a Docker [3] application for our project. The F1 project consists of three main components: the first is the PostgreSQL server for database management, the second is the application image for executing Flask, and the third is the RabbitMQ [2] server, a messaging system that facilitates communication between different components of the project. The PostgreSQL server is connected to the application container, ensuring seamless database management. The project's front end is hosted through a proxy at the application layer, providing a comprehensive and interconnected environment. Although using Docker has lots of advantages, typing the same commands again and again can be boring. Therefore, we have implemented a Makefile to make it easier to handle Docker-based commands. Makefile commands are as follows:

```
1 up:
2     docker-compose up
3     # This method builds all the images
4
5 down:
6     docker-compose down
7     # This method stops all the containers
8
9 down-volumes:
10    docker-compose down -v
11    # This method removes the created volumes besides stopping all
12      the containers
13
14 rmi:
15     docker image rm itfdb2319_app itfdb2319-app postgres rabbitmq
16         :3-management
17     # This method removes all the images
18
19 rm-app:
20     docker image rm itfdb2319-app itfdb2319_app
21     # This method removes the app image
```

```

20
21 rm-db:
22     docker image rm postgres
23     # This method removes the Postgresql image
24
25 rm-rabbitmq:
26     docker image rm rabbitmq:3-management
27     # This method removes the RabbitMQ image

```

An example command to stop the containers, keep the volumes and remove the RabbitMQ image

```
make down rm-rabbitmq
```

Warning: The make commands might give some errors due to permission based your group settings. In this case, you should run this command

```
sudo make down rm-rabbitmq
```

3.3. BULK - CUD Operations

As expected, the Admin can perform Create, Update or Delete operations. However, if the Admin wants to affect so many rows, it might be undesirable. Therefore, we have implemented Bulk CUD. This method generates queries based on the uploaded CSV file's name and the file itself. There is a convention for naming the CSV file:

- Every table has an ID, which is kept in the lkp_tables table
- Operation abbreviation: UP, IN and DEL

Example file name: 4IN.csv

- 4 is the ID of driver table
- IN stands for the INSERT operation
- Each line in the 4IN.csv file represents a record
- {column_name} : {column_value} represents a key-value relation
- - is the delimiter for the {column_name} : {column_value} pairs
- , is the delimiter for the primary keys

Using this method, one-table-related operations become much easier. This is an example content of the file 4UP.csv:

```
6,forename:'Example'-surname:'Driver'-driverref:'example'
3,forename:'Existing'-surname:'Driver'-driverref:'BLG'
```

- 6 is the primary key which will be used in WHERE driverid = %s condition
- Every key and value are separated by :
- Every key-value pair is separated by -

3.4. Before Page Implementations

3.4.1. Generalization

Rewriting the implementation for CRUD operations in the front end was highly time-consuming and confusing. A generic blueprint and HTML is created for this purpose. After generic templates, the front side of the page could be created with just one line of code.

3.4.2. SQL Injection

Opting for parametrized execution of queries is the preferred approach when dealing with dynamic queries. By utilizing parameters, psycopg (PostgreSQL adapter for Python) gains the ability to discern and mitigate potential SQL attacks. This method enhances security by ensuring that user inputs are treated as parameters rather than embedded directly into the SQL query, thus minimizing the risk of malicious exploits. It's a proactive measure that fortifies the system against potential vulnerabilities associated with dynamic queries.

3.4.3. Authentication and Authorization

Access control in the system is managed through distinct admin and user roles. Specifically, administrative privileges are granted solely to administrators, enabling them to create, update, and delete records from tables for enhanced security. On the other hand, regular logged-in users are granted access to the quiz game functionality.

User accounts are created through the signup section, requiring a username and password. Subsequently, authorization occurs via the login screen. After signing in, users can update their usernames and email addresses by navigating to the user panel and clicking on their usernames. To conclude a session, users can log out using the designated button on the user panel. This hierarchical approach to authorization and user account management ensures a secure and streamlined experience within the system.

3.5. Page Implementations

Specific page details are given in this section.

3.5.1. Drivers

On this page, users can see drivers who raced or currently racing in Formula 1. Users can interact with a driver to see its details.

#	Driverref	Number	Code	Forename	Surname	Dob	Nationality	Url	Update
1	abate	None	None	Carlo	Abate	1932-07-10	Italian	link	Update
2	abecassis	None	None	George	Abecassis	1913-03-21	British	link	Update
3	acheson	None	None	Kenny	Acheson	1957-11-27	British	link	Update
4	adamich	None	None	Andrea	de Adamich	1941-10-03	Italian	link	Update
5	adams	None	None	Philippe	Adams	1969-11-19	Belgian	link	Update
6	ader	None	None	Wait	Ader	1913-12-15	American	link	Update
7	adolff	None	None	Kurt	Adolff	1921-11-05	German	link	Update
8	agabashian	None	None	Fred	Agabashian	1913-08-21	American	link	Update
9	ahrens	None	None	Kurt	Ahrens	1940-04-19	German	link	Update
10	aitken	89	AIT	Jack	Aitken	1995-09-23	British	link	Update

[Create New Row](#)

©2023, OnlyF1s
Contact Us

(a) Drivers Page

#	Driverref	Number	Code	Forename	Surname	Dob	Nationality	Url
1	abate	None	None	Carlo	Abate	1932-07-10	Italian	link
2	abecassis	None	None	George	Abecassis	1913-03-21	British	link
3	acheson	None	None	Kenny	Acheson	1957-11-27	British	link
4	adamich	None	None	Andrea	de Adamich	1941-10-03	Italian	link
5	adams	None	None	Philippe	Adams	1969-11-19	Belgian	link
6	ader	None	None	Wait	Ader	1913-12-15	American	link
7	adolff	None	None	Kurt	Adolff	1921-11-05	German	link
8	agabashian	None	None	Fred	Agabashian	1913-08-21	American	link
9	ahrens	None	None	Kurt	Ahrens	1940-04-19	German	link
10	aitken	89	AIT	Jack	Aitken	1995-09-23	British	link

©2023, OnlyF1s
Contact Us

(a) Generic User View of The Drivers Page

3.5.1.1. CRUD on Drivers Page

A User can perform CRUD operations if it is authorized as an admin.

OnlyF1s

Drivers Circuits Constructors Seasons Others Quizzes Blink Game Race Results Rankings admin Log Out

[Create New Row](#)

driverref	fallen
number	123
code	FF
forename	emir
surname	sezgin
dob	2002-07-26
nationality	Turkish
url	fallen.com

[Insert](#)

©2023, OnlyF1s
Contact Us

(a) Creating a Driver

OnlyF1s

Drivers Circuits Constructors Seasons Others Quizzes Blink Game Race Results Rankings admin Log Out

Drivers

[Delete Selected](#)

Search fallen

#	Driverref	Number	Code	Forename	Surname	Dob	Nationality	Url	Update
1	fallen	123	FF	emir	sezgin	2002-07-26	Turkish	link	Update

[Create New Row](#)

©2023, OnlyF1s
Contact Us

(a) Searching for a Driver

Drivers

[Delete Selected](#)

 Search

 10 [Show](#)
[Order By](#)

#	Driverref	Number	Code	Forename	Surname	Dob	Nationality	Url	Update
<input type="checkbox"/>	1 fallen	123	FF	emir	sezgin	2002-07-26	Turkish	link	Update
1	mollen	321	AA	emir	sezgin	2002-	Turkis	1.com	Apply

[Create New Row](#)

©2023, OnlyF1s

[Contact Us](#)

(a) Update a Driver

Drivers

[Delete Selected](#)

 Search

 10 [Show](#)
[Order By](#)

#	Driverref	Number	Code	Forename	Surname	Dob	Nationality	Url	Update
<input type="checkbox"/>	1 mollen	321	AA	emir	sezgin	2002-07-26	Turkish	link	Update

[Create New Row](#)

©2023, OnlyF1s

[Contact Us](#)

(a) Updated Data

Drivers

Search 10

Show

Order By

#	Driverref	Number	Code	Forename	Surname	Dob	Nationality	Url
1	abate	None	None	Carlo	Abate	1952-07-10	Italian	Link
2	abecassis	None	None	George	Abecassis	1913-03-21	British	Link
3	acheson	None	None	Kenny	Acheson	1957-11-27	British	Link
4	adamich	None	None	Andrea	de Adamich	1941-10-03	Italian	Link
5	adams	None	None	Philippe	Adams	1969-11-19	Belgian	Link
6	ader	None	None	Walt	Ader	1913-12-15	American	Link
7	adolff	None	None	Kurt	Adolff	1921-11-05	German	Link
8	agabashian	None	None	Fred	Agabashian	1913-08-21	American	Link
9	ahrens	None	None	Kurt	Ahrens	1940-04-19	German	Link
10	aitken	89	AIT	Jack	Aitken	1995-09-23	British	Link

(a) Generic User View of The Drivers Page

3.5.1.2. Driver Details

A driver's specific career info and personal info are shown on this page. Career info shows results about the season of raced years which are podium, win, DNF counts, and rates.

 Drivers Circuits Constructors Seasons Others Quizzes Blink Game Race Results Rankings								
#	Season	Race	Point	Wins	Podiums	DNFs		
1	2007	17	109	4 (24%)	12 (71%)	3		
2	2008	18	98	5 (28%)	10 (56%)	2		
3	2009	17	49	2 (12%)	5 (29%)	8		
4	2010	19	240	3 (16%)	9 (47%)	4		
5	2011	19	227	3 (16%)	6 (32%)	3		
6	2012	20	190	4 (20%)	7 (35%)	6		
7	2013	19	189	1 (5%)	5 (26%)	2		
8	2014	19	384	11 (58%)	16 (84%)	3		
9	2015	19	381	10 (53%)	17 (89%)	1		
10	2016	21	380	10 (48%)	17 (81%)	2		
11	2017	20	363	9 (45%)	13 (65%)	1		
12	2018	21	408	11 (52%)	17 (81%)	1		
13	2019	21	413	11 (52%)	17 (81%)	0		
14	2020	16	347	11 (69%)	14 (88%)	0		
15	2021	22	385.5	8 (36%)	17 (77%)	1		
16	2022	22	233	0 (0%)	9 (41%)	3		
17	2023	20	209	0 (0%)	6 (30%)	2		
#	Name	Surname	Seasons	Date Of Birth	Nationality	Number		
1	Lewis	Hamilton	2007-2023	1985-01-07	British	44		

(a) A Specific Driver Details Page

3.5.2. Circuits

On this page, users can see circuits around the world that are used and currently used.

The screenshot shows a table of 10 circuits. Each row includes a checkbox, the circuit's ID, its reference, name, location, country, coordinates (Lat, Lng), altitude (Alt), URL (Url), and an 'Update' button. A red 'Delete Selected' button is at the top left, and a search bar with a dropdown for item count (10) and buttons for 'Show' and 'Order By' are at the top right. A 'Create New Row' button is at the bottom left, and a footer at the bottom right contains copyright information and a 'Contact Us' link.

#	Circuitref	Name	Location	Country	Lat	Lng	Alt	Url	Update
1	adelade	Adelaide Street Circuit	Adelaide	Australia	-34.9272	138.617	58	link	Update
2	ain-diab	Ain Diab	Casablanca	Morocco	33.5786	-7.6875	19	link	Update
3	aintree	Aintree	Liverpool	UK	53.4769	-2.94056	20	link	Update
4	albert_park	Albert Park Grand Prix Circuit	Melbourne	Australia	-37.8497	144.968	10	link	Update
5	americas	Circuit of the Americas	Austin	USA	30.1328	-97.6411	161	link	Update
6	anderstorp	Scandinavian Raceway	Anderstorp	Sweden	57.2653	13.6042	153	link	Update
7	avus	AVUS	Berlin	Germany	52.4806	13.2514	53	link	Update
8	bahrain	Bahrain International Circuit	Sakhir	Bahrain	26.0325	50.5106	7	link	Update
9	baku	Baku City Circuit	Baku	Azerbaijan	40.3725	49.8533	-7	link	Update
10	boavista	Circuito da Boavista	Oporto	Portugal	41.1705	-8.67325	28	link	Update

(a) Circuits Page

3.5.3. Constructors

On this page, users can see constructors who raced or currently racing in Formula 1. Users can interact with a constructor to see its details.

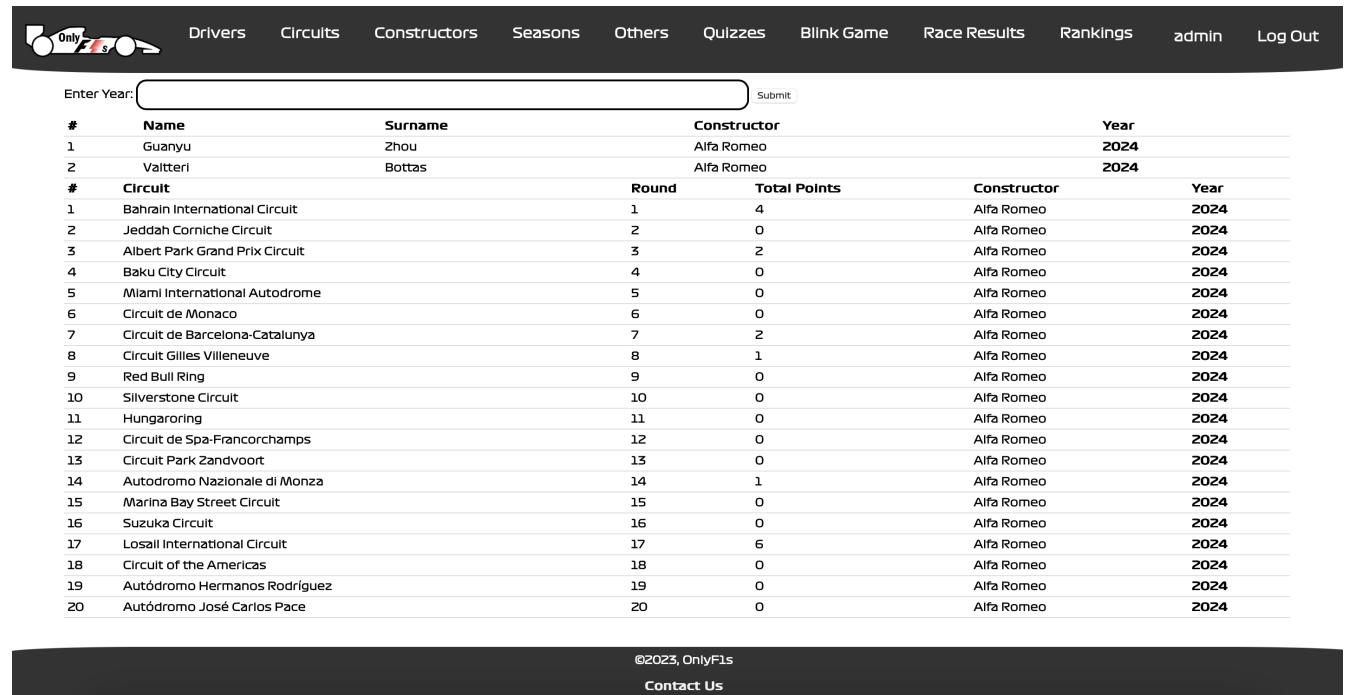
The screenshot shows a table of 10 constructors. Each row includes a checkbox, the constructor's ID, its reference, name, nationality, URL (Url), and an 'Update' button. A red 'Delete Selected' button is at the top left, and a search bar with a dropdown for item count (10) and buttons for 'Show' and 'Order By' are at the top right. A 'Create New Row' button is at the bottom left, and a footer at the bottom right contains copyright information and a 'Contact Us' link.

#	Constructorref	Name	Nationality	Url	Update
1	adams	Adams	American	link	Update
2	afm	AFM	German	link	Update
3	ags	AGS	French	link	Update
4	alfa	Alfa Romeo	Swiss	link	Update
5	alphatauri	AlphaTauri	Italian	link	Update
6	alpine	Alpine F1 Team	French	link	Update
7	alta	Alta	British	link	Update
8	amon	Amon	New Zealander	link	Update
9	apollon	Apollon	Swiss	link	Update
10	arrows	Arrows	British	link	Update

(a) Constructors Page

3.5.3.1. Constructor Details

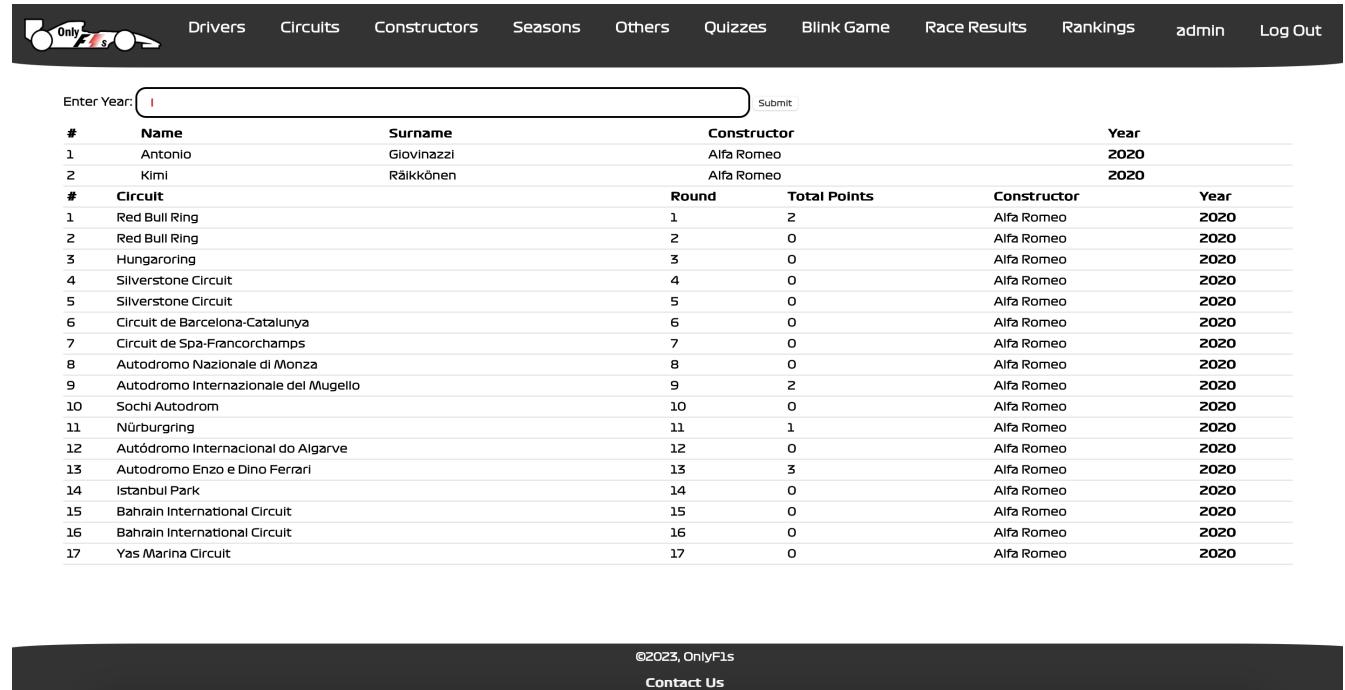
Seasonal driver and circuit details are shown on this page. Users can search a season year and data about the drivers and circuits that have a relation with the specific constructor.



The screenshot shows a table of race results for the Alfa Romeo team in the 2024 season. The table includes columns for # (Race Number), Name (Driver), Surname, Constructor (Alfa Romeo), Year (2024), Circuit (Race Track), Round, Total Points, and Constructor (Alfa Romeo). The table lists 20 races from Bahrain International Circuit to Autódromo José Carlos Pace.

#	Name	Surname	Constructor	Year	
1	Guanyu	Zhou	Alfa Romeo	2024	
2	Valtteri	Bottas	Alfa Romeo	2024	
#	Circuit	Round	Total Points	Constructor	Year
1	Bahrain International Circuit	1	4	Alfa Romeo	2024
2	Jeddah Corniche Circuit	2	0	Alfa Romeo	2024
3	Albert Park Grand Prix Circuit	3	2	Alfa Romeo	2024
4	Baku City Circuit	4	0	Alfa Romeo	2024
5	Miami International Autodrome	5	0	Alfa Romeo	2024
6	Circuit de Monaco	6	0	Alfa Romeo	2024
7	Circuit de Barcelona-Catalunya	7	2	Alfa Romeo	2024
8	Circuit Gilles Villeneuve	8	1	Alfa Romeo	2024
9	Red Bull Ring	9	0	Alfa Romeo	2024
10	Silverstone Circuit	10	0	Alfa Romeo	2024
11	Hungaroring	11	0	Alfa Romeo	2024
12	Circuit de Spa-Francorchamps	12	0	Alfa Romeo	2024
13	Circuit Park Zandvoort	13	0	Alfa Romeo	2024
14	Autodromo Nazionale di Monza	14	1	Alfa Romeo	2024
15	Märina Bay Street Circuit	15	0	Alfa Romeo	2024
16	Suzuka Circuit	16	0	Alfa Romeo	2024
17	Losail International Circuit	17	6	Alfa Romeo	2024
18	Circuit of the Americas	18	0	Alfa Romeo	2024
19	Autódromo Hermanos Rodríguez	19	0	Alfa Romeo	2024
20	Autódromo José Carlos Pace	20	0	Alfa Romeo	2024

(a) A Specific Constructor Details Page



The screenshot shows a table of race results for the Alfa Romeo team in the 2020 season. The table includes columns for # (Race Number), Name (Driver), Surname, Constructor (Alfa Romeo), Year (2020), Circuit (Race Track), Round, Total Points, and Constructor (Alfa Romeo). The table lists 17 races from Red Bull Ring to Yas Marina Circuit.

#	Name	Surname	Constructor	Year	
1	Antonio	Giovinazzi	Alfa Romeo	2020	
2	Kimi	Räikkönen	Alfa Romeo	2020	
#	Circuit	Round	Total Points	Constructor	Year
1	Red Bull Ring	1	2	Alfa Romeo	2020
2	Red Bull Ring	2	0	Alfa Romeo	2020
3	Hungaroring	3	0	Alfa Romeo	2020
4	Silverstone Circuit	4	0	Alfa Romeo	2020
5	Silverstone Circuit	5	0	Alfa Romeo	2020
6	Circuit de Barcelona-Catalunya	6	0	Alfa Romeo	2020
7	Circuit de Spa-Francorchamps	7	0	Alfa Romeo	2020
8	Autodromo Nazionale di Monza	8	0	Alfa Romeo	2020
9	Autodromo Internazionale del Mugello	9	2	Alfa Romeo	2020
10	Sochi Autodrom	10	0	Alfa Romeo	2020
11	Nürburgring	11	1	Alfa Romeo	2020
12	Autódromo Internacional do Algarve	12	0	Alfa Romeo	2020
13	Autodromo Enzo e Dino Ferrari	13	3	Alfa Romeo	2020
14	Istanbul Park	14	0	Alfa Romeo	2020
15	Bahrain International Circuit	15	0	Alfa Romeo	2020
16	Bahrain International Circuit	16	0	Alfa Romeo	2020
17	Yas Marina Circuit	17	0	Alfa Romeo	2020

(a) Searched Info on the Constructor Details Page

3.5.4. Seasons

On this page, users can find the corresponding races on the searched season year.

The screenshot shows a table of 2020 Grand Prix events. The columns are 'Name', 'Year', 'Circuit', and 'Location'. The table includes 17 entries, each with a date, circuit name, and city/country. A search bar at the top is set to '2010'.

Name	Year	Circuit	Location
Austrian Grand Prix	05/07/2020	Red Bull Ring	Spielberg Austria
Styrian Grand Prix	12/07/2020	Red Bull Ring	Spielberg Austria
Hungarian Grand Prix	19/07/2020	Hungaroring	Budapest Hungary
British Grand Prix	02/08/2020	Silverstone Circuit	Silverstone UK
70th Anniversary Grand Prix	09/08/2020	Silverstone Circuit	Silverstone UK
Spanish Grand Prix	16/08/2020	Circuit de Barcelona-Catalunya	Montmeló Spain
Belgian Grand Prix	30/08/2020	Circuit de Spa-Francorchamps	Spa Belgium
Italian Grand Prix	06/09/2020	Autodromo Nazionale di Monza	Monza Italy
Tuscan Grand Prix	13/09/2020	Autodromo Internazionale del Mugello	Mugello Italy
Russian Grand Prix	27/09/2020	Sochi Autodrom	Sochi Russia
Eifel Grand Prix	11/10/2020	Nürburgring	Nürburg Germany
Portuguese Grand Prix	25/10/2020	Autódromo Internacional do Algarve	Portimão Portugal
Emilia Romagna Grand Prix	03/11/2020	Autodromo Enzo e Dino Ferrari	Imola Italy
Turkish Grand Prix	15/11/2020	Istanbul Park	Istanbul Turkey
Bahrain Grand Prix	29/11/2020	Bahrain International Circuit	Sakhir Bahrain
Sakhir Grand Prix	06/12/2020	Bahrain International Circuit	Sakhir Bahrain
Abu Dhabi Grand Prix	13/12/2020	Yas Marina Circuit	Abu Dhabi UAE

(a) Seasons Page

3.5.5. Others

The pages were not explicitly utilized but were shown and generated generically.

The screenshot shows a dropdown menu from the 'Others' tab. The menu includes links for Qualifying, Driver Standings, Sprint Results, Results, Pit Stops, Status, Constructor Standings, Lap Times, Races, and Constructor Results. Below the menu, there is a welcome message and developer credits.

Welcome to OnlyF1s!

OnlyF1s is a fandom F1 website created for the enjoyment of Formula 1 racing fans. You can access driver, constructor and team statistics, as well as race results and qualifying data from the years 1950 to 2023. Moreover, you can access the other tabs on the top menu, such as 'Qualifying', 'Driver Standings', 'Sprint Results', 'Results', 'Pit Stops', 'Status', 'Constructor Standings', 'Lap Times', 'Races' and 'Constructor Results'. Additionally, you can access the mini quizzes we've prepared for you. Furthermore, you can have fun and test your knowledge of F1 racing with the 'Blink Game' we've created. Lastly, you can access special race results and lap times. For your feedback and suggestions, you can contact us via our 'Contact Us' page located in the footer.

To view the details of each race, click here.

Developers: Emircan Erol, Mustafa Can Çalışkan, Yusuf Emir Sezgin, Yusuf Şahin

(a) Others Page



Drivers Circuits Constructors Seasons Others Quizzes Blink Game Race Results Rankings admin Log Out

Qualifying

[Delete Selected](#) [Search type...](#) [10 ↕](#) [Show](#) [Order By](#)

#	RaceId	DriverId	ConstructorId	Number	Position	Q1	Q2	Q3	Update	
<input type="checkbox"/>	1	1	10	7	10	6	1:25.499	1:25.281	1:26.975	Update
<input type="checkbox"/>	2	1	8	6	4	9	1:25.899	1:25.380	1:27.163	Update
<input type="checkbox"/>	3	1	20	9	15	3	1:25.938	1:25.121	1:26.830	Update
<input type="checkbox"/>	4	1	3	3	16	5	1:25.846	1:25.123	1:26.973	Update
<input type="checkbox"/>	5	1	13	6	3	7	1:25.844	1:25.319	1:27.033	Update
<input type="checkbox"/>	6	1	15	7	9	8	1:26.194	1:25.265	1:27.127	Update
<input type="checkbox"/>	7	1	18	23	22	1	1:25.211	1:24.855	1:26.202	Update
<input type="checkbox"/>	8	1	22	23	23	2	1:25.006	1:24.783	1:26.505	Update
<input type="checkbox"/>	9	1	9	2	5	4	1:25.922	1:25.152	1:26.914	Update
<input type="checkbox"/>	10	1	17	9	14	10	1:25.427	1:25.241	1:27.246	Update

[Create New Row](#)

©2023, OnlyF1s
Contact Us

(a) An Example for the Others Page



Drivers Circuits Constructors Seasons Others Quizzes Blink Game Race Results Rankings admin Log Out

Laptimes

[Delete Selected](#) [Search type...](#) [10 ↕](#) [Show](#) [Order By](#)

#	RaceId	DriverId	Lap	Position	Time	Milliseconds	Update	
<input type="checkbox"/>	1	1	13	27	3	00:01:30.262000	90262	Update
<input type="checkbox"/>	2	1	13	30	3	00:01:29.141000	89141	Update
<input type="checkbox"/>	3	1	13	24	3	00:02:22.712000	142712	Update
<input type="checkbox"/>	4	1	13	26	3	00:01:31.513000	91513	Update
<input type="checkbox"/>	5	1	13	28	3	00:01:29.872000	89872	Update
<input type="checkbox"/>	6	1	13	29	3	00:01:29.559000	89559	Update
<input type="checkbox"/>	7	1	13	22	3	00:02:35.808000	155808	Update
<input type="checkbox"/>	8	1	13	23	3	00:02:31.699000	151699	Update
<input type="checkbox"/>	9	1	13	25	3	00:01:33.747000	93747	Update
<input type="checkbox"/>	10	1	13	31	13	00:01:50.326000	110326	Update

[Create New Row](#)

©2023, OnlyF1s
Contact Us

(a) An Example for the Others Page

3.5.6. Quiz

On this page, users can interact with quiz questions that show randomly, and their score is stored in the database. The initially established many-to-many relationship between the "Users" and "Quizzes" tables has been altered to one-to-many relations, specifically with the "Users" table. The adjustment aims to streamline the database model and enhance clarity in the representation of user-quiz relationships.

Quizzes

[Create a Quiz](#)[Update the Quiz](#)[Delete the Quiz](#)

Who is the youngest F1 pilot?

- Thomas Monarch
- Lance Stroll
- Max Verstappen

[Submit](#)

©2023, OnlyF1s

[Contact Us](#)

(a) Quiz Page

Quizzes

[Create a Quiz](#)[Update the Quiz](#)[Delete the Quiz](#)

Who is the youngest F1 pilot?

- Thomas Monarch
- Lance Stroll
- Max Verstappen

[Submit](#)

©2023, OnlyF1s

[Contact Us](#)

(a) Answering a Question



True

Your Quiz Score: 1

[Next](#)



(a) Answer of the Question



blinkscore: 50

email: None

pp: None

quizscore: 1

role: 1

userId: 1

userName: admin

[Update Informations](#)

[Delete Account](#)



(a) Quiz Score Updated on User Page

3.5.6.1. CRUD Operations On Quiz Page

Question:
Is there any admin?

Option 1:
Yes

Option 2:
No

Correct Answer:
Correct Answer[]

Create Quiz

©2023, OnlyF1s
Contact Us

(a) Create a New Question

Is there any admin?

Question:
Is there any user?

Option 1:
Yes

Option 2:
No

Correct Answer:
User needs to choose!!

Update Quiz

©2023, OnlyF1s
Contact Us

(a) Update a Question



Quizzes

[Create a Quiz](#) [Update the Quiz](#) [Delete the Quiz](#)

Is there any user?

- No
- User needs to choose()
- Yes

[Submit](#)

©2023, OnlyF1s

[Contact Us](#)

(a) Updated Question as Shown In the Page

3.5.7. Blink Game

On this page, users can test their response time to the F1 starting light test. The best response time is stored in the database for each user.



F1 Blink Test

Welcome to the Grid Light Test which is known as "The Blink Test".

For a second think yourself as a F1 driver. The race will be start soon and you are waiting for the lights.

Even milliseconds are important! You cannot blink now!!

Lights slowly turns red aaaaand...

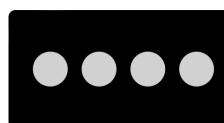
GREEN!!! GO!!

So this test determines your respond time. Have it a try!!

|

|

|



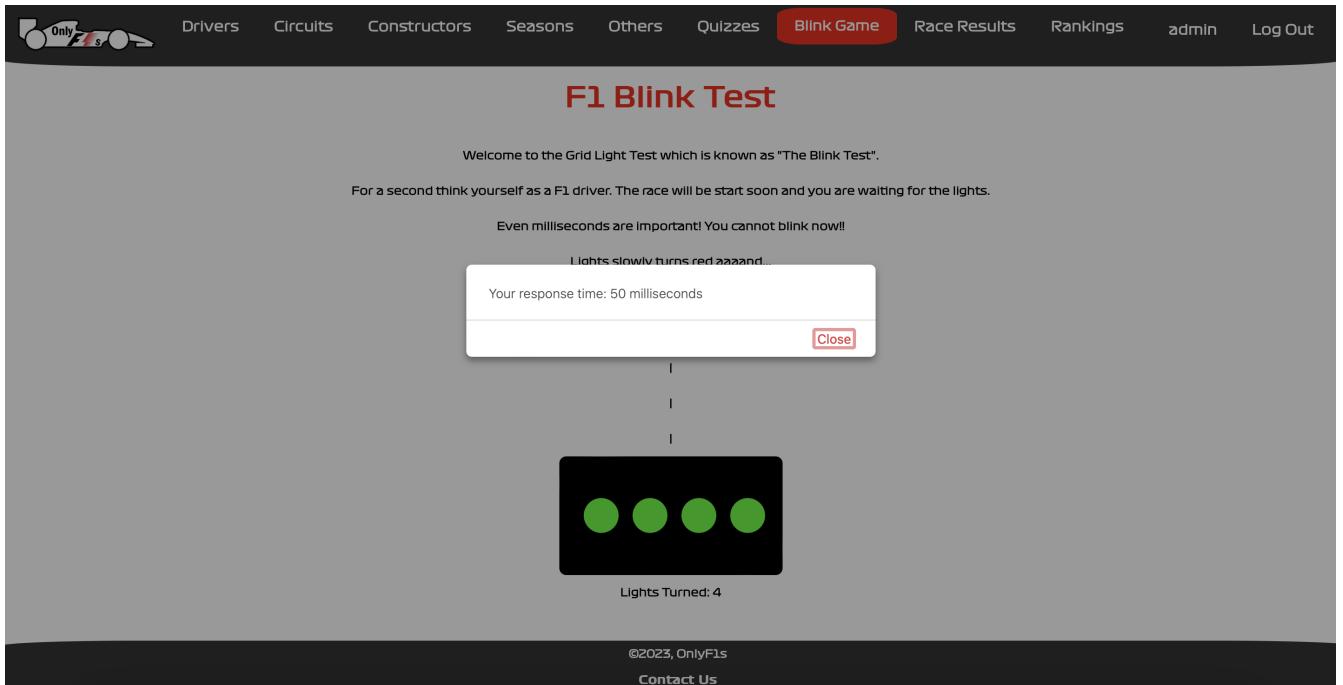
[Start Game](#)

Lights Turned: 0

©2023, OnlyF1s

[Contact Us](#)

(a) Blink Game Page



(a) Respond Time Pop-Up Page

blinkscore:	50
email:	None
pp:	None
quizscore:	None
role:	1
userId:	1
userName:	admin

(a) Blink Game Score Updated on User Page

3.5.8. Race Results

The "Race Results" table, integral to our database project, consolidates data from races, drivers, circuits, and results to provide a comprehensive overview of race-related information. This table is designed to offer a detailed representation of race outcomes, featuring an incorporated response limiter and a search bar for enhanced functionality. The response limiter empowers users to manage the displayed data volume effectively,

fostering a focused examination of race results. Simultaneously, the search bar enables users to perform precise searches based on specific parameters, such as the year or circuit, facilitating efficient retrieval of race-related details from the database.

The screenshot shows a dark-themed web application interface. At the top, there is a navigation bar with links: Drivers, Circuits, Constructors, Seasons, Others, Quizzes, Blink Game, Race Results (which is highlighted in red), Rankings, admin, and Log Out. Below the navigation bar are two search input fields: 'Year' (containing '2020') and 'Circuit' (containing 'Istanbul'). A dropdown menu labeled '10' and a red 'Submit' button are positioned next to the circuit field. Below the search fields, a message says 'No results found.'

©2023, OnlyF1s
Contact Us

(a) Race Results Page

The screenshot shows the same dark-themed web application interface as above. The search results for 'monaco' in 2020 are displayed in a table. The table has columns: Name, Surname, Nationality, No., Circuit name, Year, fastest lap, position, and position text*. The table contains 10 rows, each representing a driver's result at Istanbul Park, Turkey in 2020. The drivers listed are Lewis Hamilton, Sergio Pérez, Sebastian Vettel, Charles Leclerc, Carlos Sainz, Max Verstappen, Alexander Albon, Lando Norris, Lance Stroll, and Daniel Ricciardo.

Name	Surname	Nationality	No.	Circuit name	Year	fastest lap	position	position text*
Lewis	Hamilton	British	44	Istanbul Park, Turkey	2020	56/58	1	
Sergio	Pérez	Mexican	11	Istanbul Park, Turkey	2020	50/58	2	
Sebastian	Vettel	German	5	Istanbul Park, Turkey	2020	53/58	3	
Charles	Leclerc	Monégasque	16	Istanbul Park, Turkey	2020	47/58	4	
Carlos	Sainz	Spanish	55	Istanbul Park, Turkey	2020	56/58	5	
Max	Verstappen	Dutch	33	Istanbul Park, Turkey	2020	53/58	6	
Alexander	Albon	Thai	23	Istanbul Park, Turkey	2020	54/58	7	
Lando	Norris	British	4	Istanbul Park, Turkey	2020	58/58	8	
Lance	Stroll	Canadian	18	Istanbul Park, Turkey	2020	55/58	9	
Daniel	Ricciardo	Australian	3	Istanbul Park, Turkey	2020	54/58	10	

©2023, OnlyF1s
Contact Us

(a) Searched Results

3.5.9. Rankings

On this page, different comparison tables are shown as a list. Users can interact and see the specific comparison table.

The screenshot shows a dark-themed web interface for the OnlyF1s website. At the top, there is a navigation bar with links: Drivers, Circuits, Constructors, Seasons, Others, Quizzes, Blink Game, Race Results, Rankings (which is highlighted in red), admin, and Log Out. Below the navigation bar, the word "Rankings" is centered in red. A vertical sidebar on the left lists various ranking categories: Total Entries, Youngest Entries, Oldest Entries, Youngest Finish, Oldest Finish, Total Wins, Percentage Wins, Youngest Wins, and Oldest Wins. The "Oldest Entries" category is currently selected, indicated by a grey background. At the bottom of the page, there is a dark footer bar with the text "©2023, OnlyF1s" and a "Contact Us" link.

(a) Rankings Page

3.5.9.1. A Ranking Example

PerWins Page shows drivers, drivers' race entry counts, win rates, the season of the first wins, and the season of the last wins.

perWins

#	Name	Surname	Seasons	Entries	Wins	%
1	Lee	Wallard	1951 - 1951	2	1	50.00
2	Juan	Fangio	1950 - 1957	58	24	41.38
3	Bill	Vukovich	1953 - 1954	5	2	40.00
4	Alberto	Ascari	1951 - 1953	36	13	36.11
5	Jim	Clark	1962 - 1968	73	25	34.25
6	Lewis	Hamilton	2007 - 2021	330	103	31.21
7	Michael	Schumacher	1992 - 2006	308	91	29.55
8	Max	Verstappen	2016 - 2023	183	52	28.42
9	Jackie	Stewart	1965 - 1973	100	27	27.00
10	Ayrton	Senna	1985 - 1993	162	41	25.31

©2023, OnlyF1s

[Contact Us](#)

(a) perWins Ranking Page

perWins

#	Name	Surname	Seasons	Entries	Wins	%
1	Lee	Wallard	1951 - 1951	2	1	50.00
2	Juan	Fangio	1950 - 1957	58	24	41.38
3	Bill	Vukovich	1953 - 1954	5	2	40.00
4	Alberto	Ascari	1951 - 1953	36	13	36.11
5	Jim	Clark	1962 - 1968	73	25	34.25
6	Lewis	Hamilton	2007 - 2021	330	103	31.21
7	Michael	Schumacher	1992 - 2006	308	91	29.55
8	Max	Verstappen	2016 - 2023	183	52	28.42
9	Jackie	Stewart	1965 - 1973	100	27	27.00
10	Ayrton	Senna	1985 - 1993	162	41	25.31
11	Alain	Prost	1981 - 1993	202	51	25.25
12	Stirling	Moss	1955 - 1961	73	16	21.92
13	Bob	Sweikert	1955 - 1955	5	1	20.00
14	Damon	Hill	1993 - 1998	122	22	18.03
15	Sebastian	Vettel	2008 - 2019	300	53	17.67
16	Pat	Fitzherity	1956 - 1956	6	1	16.67
17	Nigel	Mansell	1985 - 1994	192	31	16.15
18	Tony	Brooks	1957 - 1959	41	6	14.63
19	Niki	Lauda	1974 - 1985	174	25	14.37
20	Nino	Farina	1950 - 1953	37	5	13.51

©2023, OnlyF1s

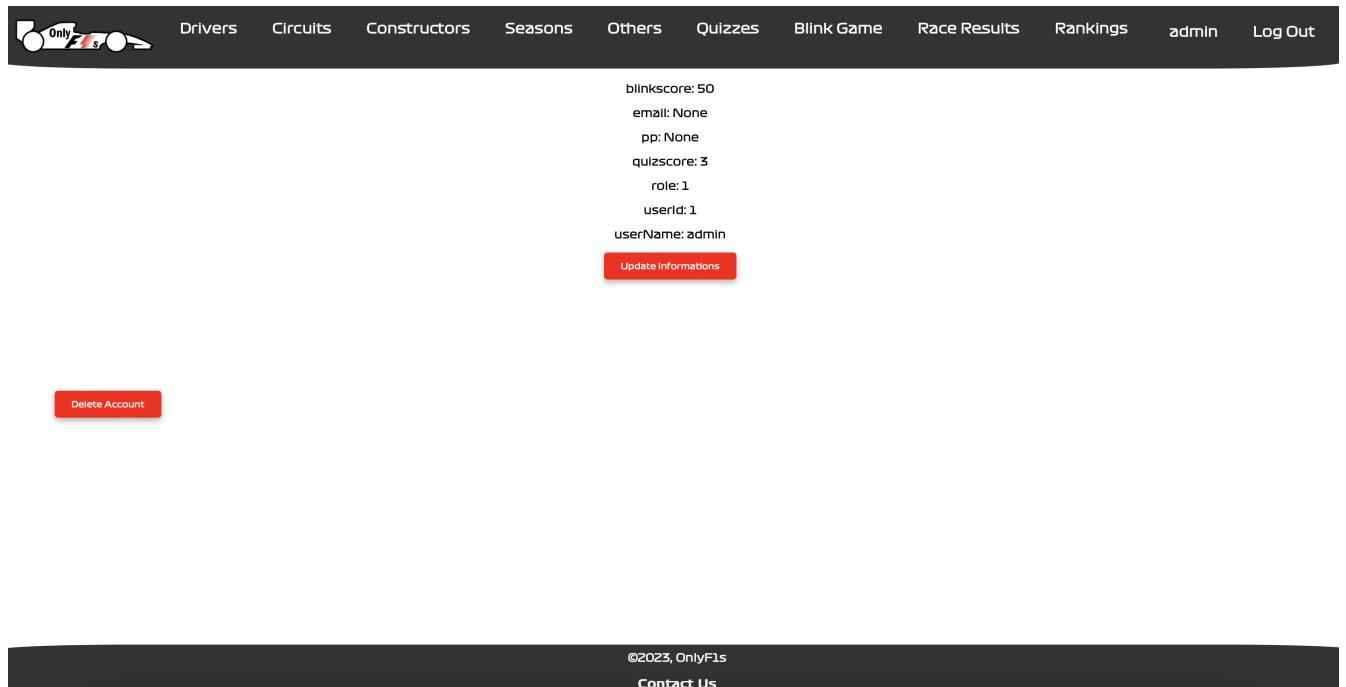
[Contact Us](#)

(a) Limited Ranking Table

3.5.10. User

Upon logging in, users have access to their personalized information within the system, including their username, game scores, and email. For added flexibility, users are empowered to update their email and username if desired. Additionally, an option is provided for users to delete their accounts, granting them control over their account.

management. This user-centric approach ensures that individuals can easily manage and customize their profiles based on their preferences and changing information.

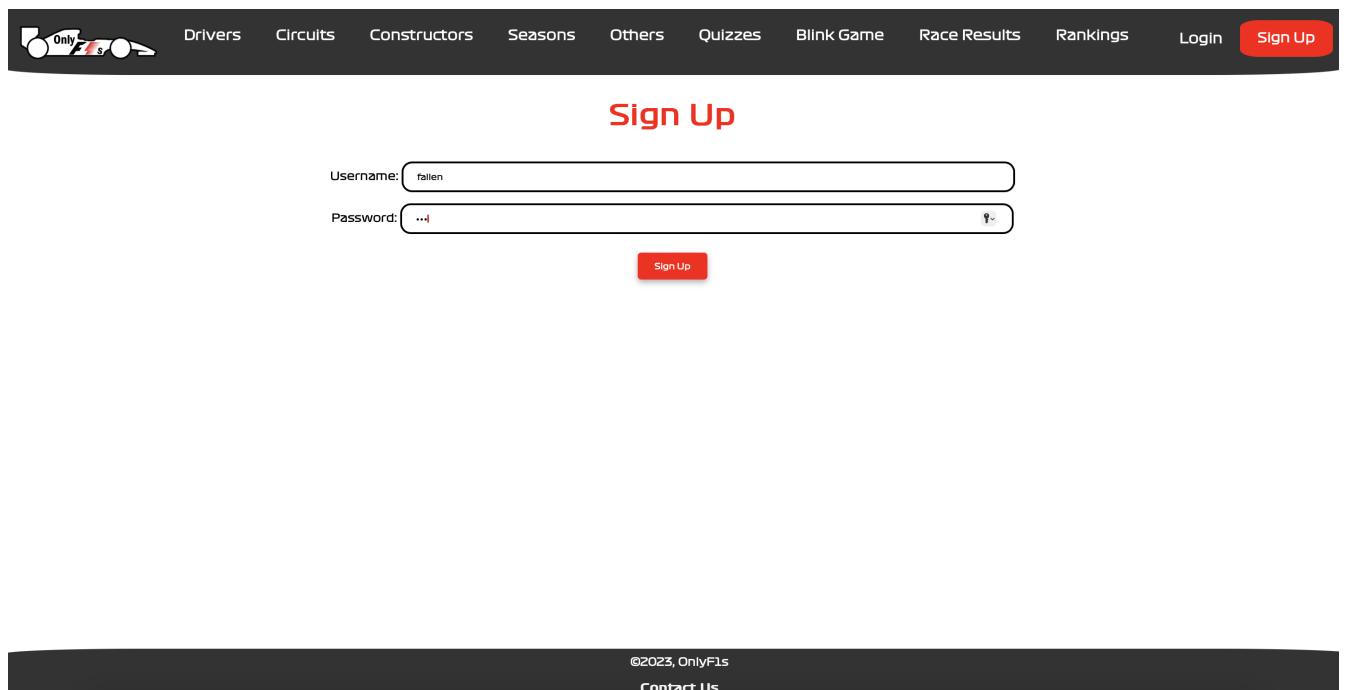


The screenshot shows the user profile page for an admin user. At the top, there's a navigation bar with links for Drivers, Circuits, Constructors, Seasons, Others, Quizzes, Blink Game, Race Results, Rankings, and Log Out. The admin user's profile information is displayed below the navigation bar:

- blinkscore: 50
- email: None
- pp: None
- quizscore: 3
- role: 1
- userId: 1
- userName: admin

A red "Update Informations" button is located at the bottom of this section. Below this, there's a "Delete Account" button. At the very bottom of the page, there's a footer with copyright information and a "Contact Us" link.

(a) User Information Page for Admin



The screenshot shows the sign-up page. At the top, there's a navigation bar with links for Drivers, Circuits, Constructors, Seasons, Others, Quizzes, Blink Game, Race Results, Rankings, Login, and a red "Sign Up" button. The main heading is "Sign Up". Below the heading are two input fields: "Username:" containing "fallen" and "Password:" containing "•••". To the right of the password field is a dropdown menu showing "♀". A red "Sign Up" button is located at the bottom of the form.

(a) Signing Up as a New User



blinkscore: None

email: None

pp: None

quizscore: None

role: 0

userId: 2

userName: fallen

[Update Informations](#)

[Delete Account](#)

©2023, OnlyF1s

[Contact Us](#)

(a) New Created User Information

3.5.11. Contact Us

Users can interact with "Contact Us" at the bottom of every page to send emails to the admin mail.



Email:

fallen@gmail.com

Subject:

I have a request!

Message:

Is this work?

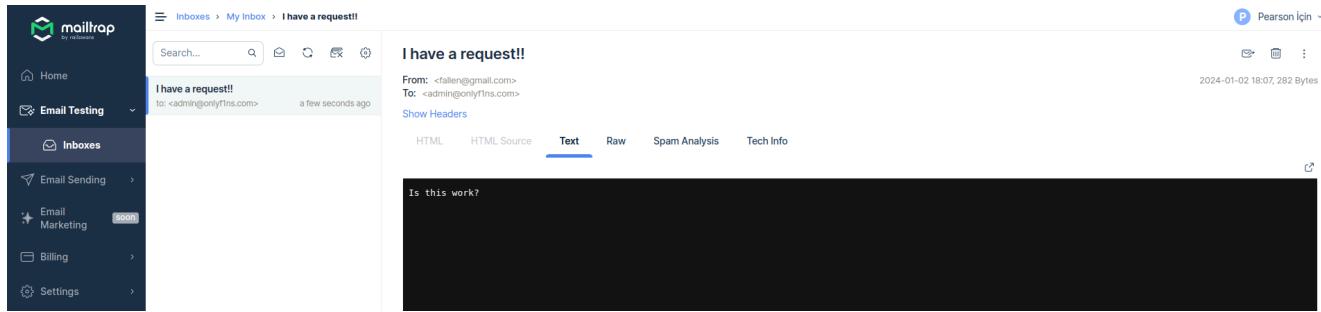


[Send Email](#)

©2023, OnlyF1s

[Contact Us](#)

(a) Contact Us Page



(a) An Email Sent By the User and Shown By the Admin Email

4 Conclusion

4.1. Discussions

We had designed a module to automatically generate and send emails for the 'Contact Us' section of the website. To ensure that the creation and sending of emails could be carried out asynchronously from the other functionalities of the site (preventing the site from remaining in a loading state until the email was sent), we planned to implement RabbitMQ. However, due to time constraints, we had to cancel the implementation.

We also planned to manage CRUD operations over a remote database (using services akin to Cockroach Labs [1]), but due to insufficient time, we had to cancel this as well.

4.2. Conclusion

Ultimately, we have established a Formula 1 Fandom website using Python's Flask framework. On this site, all users can view Formula 1 related content, participate in quizzes and Blink Test activities (with their scores visible on their respective panels), and an admin user can perform CRUD operations on the site's data. Throughout the project, all team members collaborated and contributed equally.

Bibliography

- [1] Cockroach Labs. [https://www.cockroachlabs.com/.](https://www.cockroachlabs.com/)
- [2] RabbitMQ Features. <https://www.rabbitmq.com/#features/.>
- [3] Developer Experience. Dockerizing. <https://developerexperience.io/articles/dockerizing.>
- [4] Pallets Projects. Flask Documentation. [https://flask.palletsprojects.com/en/3.0.x/.](https://flask.palletsprojects.com/en/3.0.x/)
- [5] PostgreSQL Global Development Group. PostgreSQL - About. <https://www.postgresql.org/about/.>