Project 2 Part 3

Project:

a. Include name of the team members.

Alan Xu, Jai Bansal

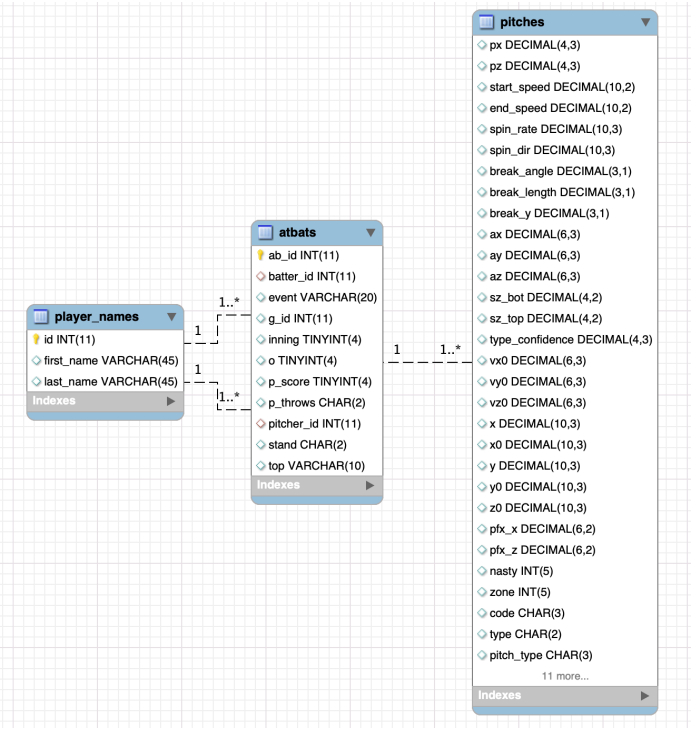
b. What is the general status of the project? Describe the roadblocks (if you have any) 3-4 sentences.

As of right now, the project is at a pretty good stage, where we are able to visualize all of the raw data that we are working with on a functional front-end client. Our project has been set up well, as we were careful to consider future functional requirements, and how our code will need to be altered/extended in the future. Now, with the app and database backend set up, we are shifting our focus towards analytics using the data that we have, so that our project actually has meaningful functionality. So far, we have been able to implement a multitude of advanced features (mainly using stored procedures), which will be shown in more detail later in this report. The roadblocks at the moment have to do with version control, while also contending with certain gaps in our individual knowledge (whether that's in relation to each other or the project itself) that have made it difficult to incorporate some of the advanced features we've learned throughout this course.

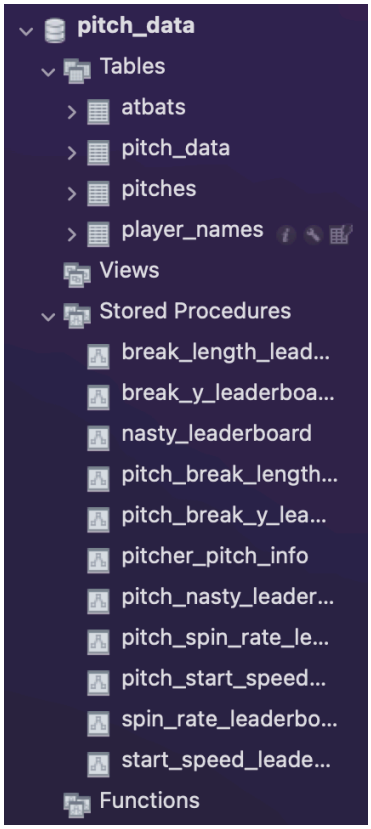
Database:

c. By this assignment submission date, you should have your decomposition completed with all the tables ready in database. You should have also implemented many of the advanced features. Clearly label the screenshots.

i. Include the UML screenshot.



ii. Include the screenshots of decomposed tables.



iii. Include screenshots of your implementation of some advanced features (Stored procedures, views, triggers etc.)

Below, I included some screenshots of some of the stored procedures we wrote for this component of the project. Basically, what these stored procedures do is take in some parameters (pitch name, pitch type etc) depending on the function and return either a sorted leaderboard of pitchers with that attribute or the information about a specific pitcher. These stored procedures are going to be the backbone of our application, because the most important functionality is going to be users getting to choose different pitchers and attributes and seeing information based on their choices. These stored procedures are then called in our front end – back end connection in Python.

```
CREATE PROCEDURE pitcher_pitch_info(IN pitch_name_param VARCHAR(55), IN pitch_type_param CHAR(3))

BEGIN

    SELECT pitch_type, AVG(start_speed), AVG(spin_rate), AVG(break_length), AVG(break_y)
    FROM pitch_data
    WHERE pitcher_name =  pitch_name_param AND pitch_type = pitch_type_param
    GROUP BY pitch_type;

END //

CALL pitcher_pitch_info('Aroldis Chapman', 'FF');

-- Pitch / Start Speed Leaderboard

DROP PROCEDURE IF EXISTS pitch_start_speed_leaderboard;

DELIMITER //

CREATE PROCEDURE pitch_start_speed_leaderboard(IN pitch_type_param CHAR(3), IN sample_size INT)

BEGIN

    SELECT pitcher_name, AVG(start_speed)
    FROM pitch_data
    WHERE pitch_type = pitch_type_param
    GROUP BY pitcher_name
    HAVING COUNT(start_speed) > sample_size
    ORDER BY AVG(start_speed) DESC;

END //

CALL pitch_start_speed_leaderboard('FF',500);
```

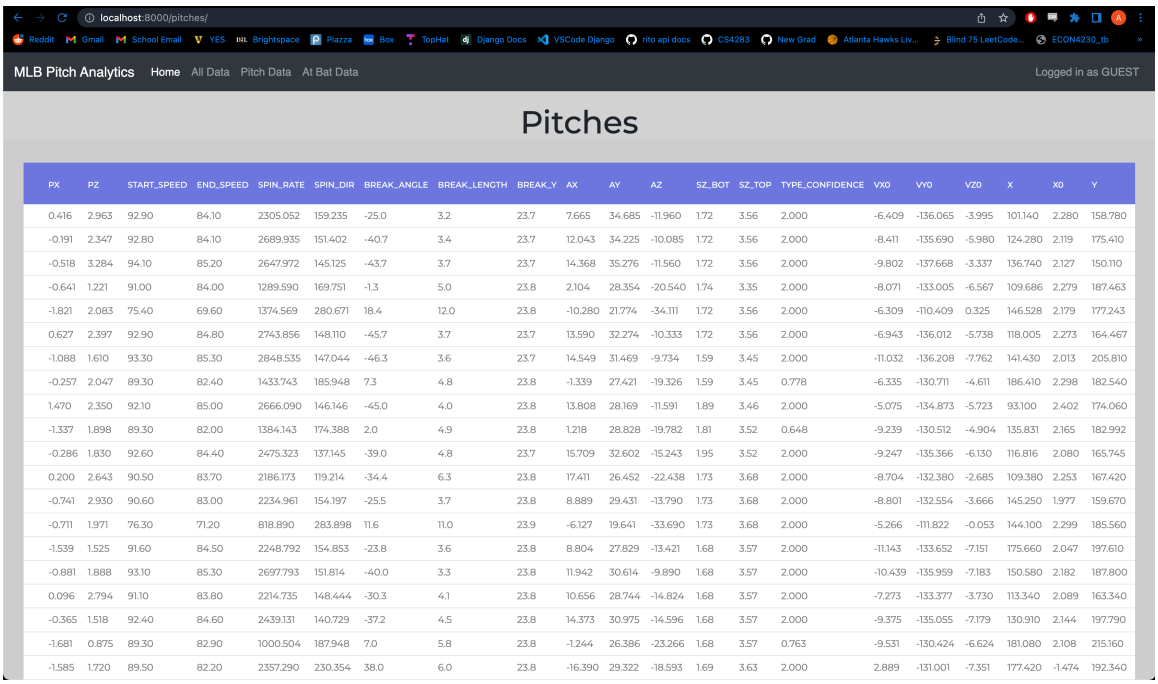
d. In terms of percentage how much you think you have completed on database side of the project? Describe the completed work in 2-3 sentences. Describe roadblocks, if any.

We think we are about 75% done with the back end database side of the project. Our decomposition is pretty much done into smaller tables, which was pretty trivial because the data came nicely in a way that was already decomposed. We then wrote a bunch of core stored procedures that will serve as the core functionality for users to filter and look up stats and attributes for their favorite pitchers. We really only have two things left: adding some DML scripts with some triggers that allow users to add/update data (although this isn't really too practical) and then some more advanced stored procedures that will allow the users more flexibility in searching the database.

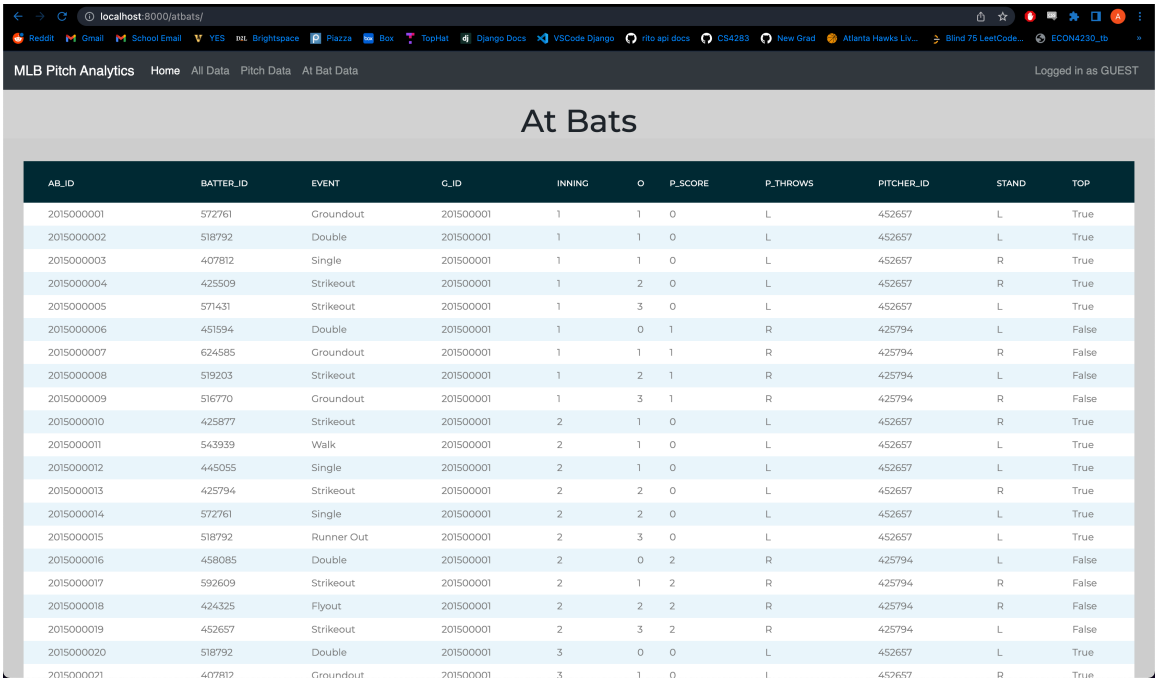
Front end:

e. By this assignment submission date, you should have a working interface showing data from the database in live connection. You should also have multiple UI pages. Clearly label the screenshots.

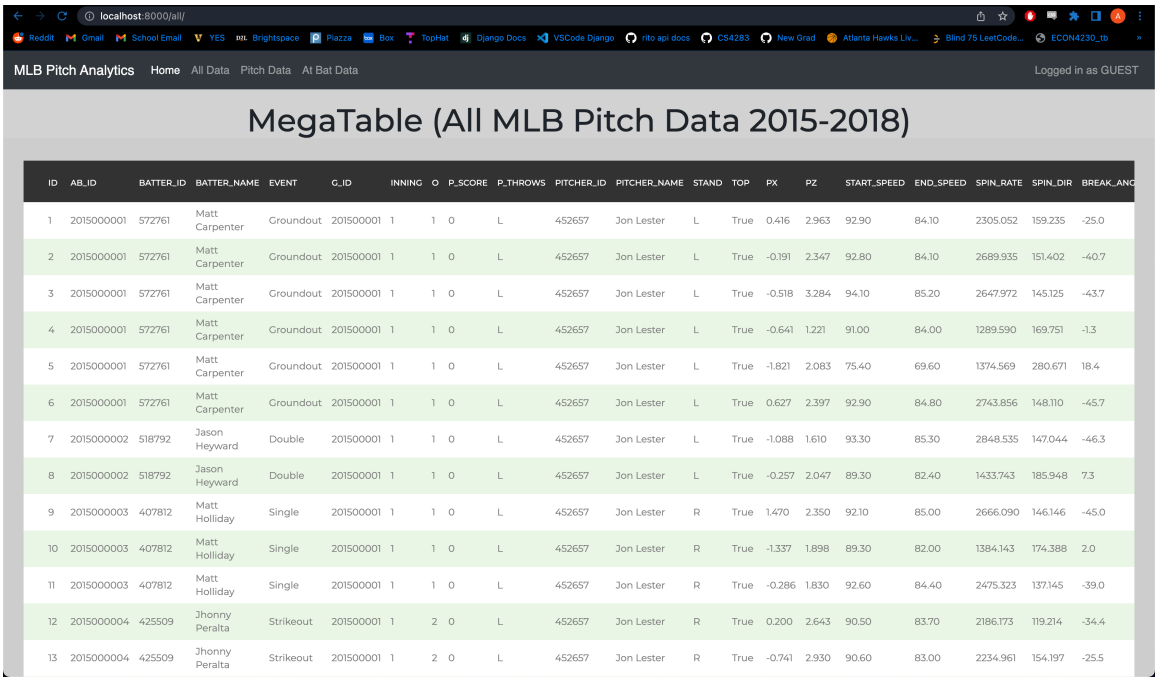
i. Attach screenshots of the browser showing different UI pages. (At least 2 excluding home page)



PX	PZ	START_SPEED	END_SPEED	SPIN_RATE	SPIN_DIR	BREAK_ANGLE	BREAK_LENGTH	BREAK_Y	AX	AY	AZ	SZ_BOT	SZ_TOP	TYPE_CONFIDENCE	VX0	VY0	VZ0	X	X0	Y
0.416	2.963	92.90	84.10	2305.052	159.235	-25.0	3.2	23.7	7.665	34.685	-11.960	1.72	3.56	2.000	-6.409	-136.065	-3.995	101.140	2.280	158.780
-0.191	2.347	92.80	84.10	2689.935	151.402	-40.7	3.4	23.7	12.043	34.225	-10.085	1.72	3.56	2.000	-8.411	-135.690	-5.980	124.280	2.119	175.410
-0.518	3.284	94.10	85.20	2647.972	145.125	-43.7	3.7	23.7	14.368	35.276	-11.560	1.72	3.56	2.000	-9.802	-137.668	-3.337	136.740	2.127	150.110
-0.641	1.221	91.00	84.00	1289.590	169.751	-1.3	5.0	23.8	2.104	28.354	-20.540	1.74	3.35	2.000	-8.071	-133.005	-6.567	109.686	2.279	187.463
-1.821	2.083	75.40	69.60	1374.569	280.671	18.4	12.0	23.8	-10.280	21.774	-34.111	1.72	3.56	2.000	-6.309	-110.409	0.325	146.528	2.179	177.243
0.627	2.397	92.90	84.80	2743.856	148.110	-45.7	3.7	23.7	13.590	32.274	-10.333	1.72	3.56	2.000	-6.943	-136.012	-5.738	118.005	2.273	164.467
-1.088	1.610	93.30	85.30	2848.535	147.044	-46.3	3.6	23.7	14.549	31.469	-9.734	1.59	3.45	2.000	-11.032	-136.208	-7.762	141.430	2.013	205.810
-0.257	2.047	89.30	82.40	1433.743	185.948	7.3	4.8	23.8	-1.339	27.421	-19.326	1.59	3.45	0.778	-6.335	-130.711	-4.611	186.410	2.298	182.540
1.470	2.350	92.10	85.00	2666.090	146.146	-45.0	4.0	23.8	13.808	28.169	-11.591	1.89	3.46	2.000	-5.075	-134.873	-5.723	93.100	2.402	174.060
-1.337	1.898	89.30	82.00	1384.143	174.388	2.0	4.9	23.8	1.218	28.828	-19.782	1.81	3.52	0.648	-9.239	-130.512	-4.904	135.831	2.165	182.992
-0.286	1.830	92.60	84.40	2475.323	137.145	-39.0	4.8	23.7	15.709	32.602	-15.243	1.95	3.52	2.000	-9.247	-135.366	-6.130	116.816	2.080	165.745
0.200	2.643	90.50	83.70	2186.173	119.214	-34.4	6.3	23.8	17.411	26.452	-22.438	1.73	3.68	2.000	-8.704	-132.380	-2.685	109.380	2.253	167.420
-0.741	2.930	90.60	83.00	2234.961	154.197	-25.5	3.7	23.8	8.889	29.431	-13.790	1.73	3.68	2.000	-8.801	-132.554	-3.666	145.250	1.977	159.670
-0.711	1.971	76.30	71.20	818.890	283.898	11.6	11.0	23.9	-6.127	19.641	-33.690	1.73	3.68	2.000	-5.266	-111.822	-0.053	144.100	2.299	185.560
-1.539	1.525	91.60	84.50	2248.792	154.853	-23.8	3.6	23.8	8.804	27.829	-13.421	1.68	3.57	2.000	-11.143	-133.652	-7.151	175.660	2.047	197.610
-0.881	1.888	93.10	85.30	2697.793	151.814	-40.0	3.3	23.8	11.942	30.614	-9.890	1.68	3.57	2.000	-10.439	-135.959	-7.183	150.580	2.182	187.800
0.096	2.794	91.10	83.80	2214.735	148.444	-30.3	4.1	23.8	10.656	28.744	-14.824	1.68	3.57	2.000	-7.273	-133.377	-3.730	113.340	2.089	163.340
-0.365	1.518	92.40	84.60	2439.131	140.729	-37.2	4.5	23.8	14.373	30.975	-14.596	1.68	3.57	2.000	-9.375	-135.055	-7.179	130.910	2.144	197.790
-1.681	0.875	89.30	82.90	1000.504	187.948	7.0	5.8	23.8	-1.244	26.386	-23.266	1.68	3.57	0.763	-9.531	-130.424	-6.624	181.080	2.108	215.160
-1.585	1.720	89.50	82.20	2357.290	230.354	38.0	6.0	23.8	-16.390	29.322	-18.593	1.69	3.63	2.000	2.889	-131.001	-7.351	177.420	-1.474	192.340



AB_ID	BATTER_ID	EVENT	G_ID	INNING	O	P_SCORE	P_THROWS	PITCHER_ID	STAND	TOP
2015000001	572761	Groundout	201500001	1	1	0	L	452657	L	True
2015000002	518792	Double	201500001	1	1	0	L	452657	L	True
2015000003	407812	Single	201500001	1	1	0	L	452657	R	True
2015000004	425509	Strikeout	201500001	1	2	0	L	452657	R	True
2015000005	571431	Strikeout	201500001	1	3	0	L	452657	L	True
2015000006	451594	Double	201500001	1	0	1	R	425794	L	False
2015000007	624585	Groundout	201500001	1	1	1	R	425794	R	False
2015000008	519203	Strikeout	201500001	1	2	1	R	425794	L	False
2015000009	516770	Groundout	201500001	1	3	1	R	425794	R	False
2015000010	425877	Strikeout	201500001	2	1	0	L	452657	R	True
2015000011	543939	Walk	201500001	2	1	0	L	452657	L	True
2015000012	445055	Single	201500001	2	1	0	L	452657	L	True
2015000013	425794	Strikeout	201500001	2	2	0	L	452657	R	True
2015000014	572761	Single	201500001	2	2	0	L	452657	L	True
2015000015	518792	Runner Out	201500001	2	3	0	L	452657	L	True
2015000016	458085	Double	201500001	2	0	2	R	425794	L	False
2015000017	592609	Strikeout	201500001	2	1	2	R	425794	R	False
2015000018	424325	Flyout	201500001	2	2	2	R	425794	R	False
2015000019	452657	Strikeout	201500001	2	3	2	R	425794	L	False
2015000020	518792	Double	201500001	3	0	0	L	452657	L	True
2015000021	407812	Groundout	201500001	3	1	0	L	452657	R	True



ID	AB_ID	BATTER_ID	BATTER_NAME	EVENT	G_ID	INNING	O	P_SCORE	P_THROWS	PITCHER_ID	PITCHER_NAME	STAND	TOP	PX	PZ	START_SPEED	END_SPEED	SPIN_RATE	SPIN_DIR	BREAK_ANG
1	2015000001	572761	Matt Carpenter	Groundout	201500001	1	1	0	L	452657	Jon Lester	L	True	0.416	2.963	92.90	84.10	2305.052	159.235	-25.0
2	2015000001	572761	Matt Carpenter	Groundout	201500001	1	1	0	L	452657	Jon Lester	L	True	-0.191	2.347	92.80	84.10	2689.935	151.402	-40.7
3	2015000001	572761	Matt Carpenter	Groundout	201500001	1	1	0	L	452657	Jon Lester	L	True	-0.518	3.284	94.10	85.20	2647.972	145.125	-43.7
4	2015000001	572761	Matt Carpenter	Groundout	201500001	1	1	0	L	452657	Jon Lester	L	True	-0.641	1.221	91.00	84.00	1289.590	169.751	-1.3
5	2015000001	572761	Matt Carpenter	Groundout	201500001	1	1	0	L	452657	Jon Lester	L	True	-1.821	2.083	75.40	69.60	1374.569	280.671	18.4
6	2015000001	572761	Matt Carpenter	Groundout	201500001	1	1	0	L	452657	Jon Lester	L	True	0.627	2.397	92.90	84.80	2743.856	148.110	-45.7
7	2015000002	518792	Jason Heyward	Double	201500001	1	1	0	L	452657	Jon Lester	L	True	-1.088	1.610	93.30	85.30	2848.535	147.044	-46.3
8	2015000002	518792	Jason Heyward	Double	201500001	1	1	0	L	452657	Jon Lester	L	True	-0.257	2.047	89.30	82.40	1433.743	185.948	7.3
9	2015000003	407812	Matt Holliday	Single	201500001	1	1	0	L	452657	Jon Lester	R	True	1.470	2.350	92.10	85.00	2666.090	146.146	-45.0
10	2015000003	407812	Matt Holliday	Single	201500001	1	1	0	L	452657	Jon Lester	R	True	-1.337	1.898	89.30	82.00	1384.143	174.388	2.0
11	2015000003	407812	Matt Holliday	Single	201500001	1	1	0	L	452657	Jon Lester	R	True	-0.286	1.830	92.60	84.40	2475.323	137.145	-39.0
12	2015000004	425509	Jhonny Peralta	Strikeout	201500001	1	2	0	L	452657	Jon Lester	R	True	0.200	2.643	90.50	83.70	2186.173	119.214	-34.4
13	2015000004	425509	Jhonny Peralta	Strikeout	201500001	1	2	0	L	452657	Jon Lester	R	True	-0.741	2.930	90.60	83.00	2234.961	154.197	-25.5

ii. Attach screenshots of relevant part of your front-end code showing calls to the database. This can be embedded SQL statements or calls to procedures etc.

Django view functions (query database from the front end)

```python
def sp(request, sp_name):
    with connection.cursor() as cursor:
        cursor.callproc(sp_name, [100])
    sp_data = cursor.fetchall()[:50];
    context = {
        'sp_data': sp_data,
        'sp_name': sp_name,
        'sp_name_formatted': sp_name.replace("_", " ").title()
    }
    return render(request, 'dashboard/sp.html', context)

def sp_detail(request, page_num, sp_name):
    with connection.cursor() as cursor:
        cursor.callproc(sp_name, [100])
    start_index = page_num * 50
    end_index = start_index + 50
    sp_data = cursor.fetchall()[start_index:end_index];
    context = {
        'sp_data': sp_data,
        'sp_name': sp_name,
        'sp_name_formatted': sp_name.replace("_", " ").title()
    }
    return render(request, 'dashboard/sp.html', context)

def all_data(request):
    pitch_data = PitchData.objects.all()[:50]
    context = {'pitch_data': pitch_data}
    return render(request, 'dashboard/all_data.html', context)

def all_data_detail(request, page_num):
    start_index = page_num * 50
    end_index = start_index + 50
    pitch_data = PitchData.objects.all()[start_index:end_index]
    context = {'pitch_data': pitch_data}
    return render(request, 'dashboard/all_data.html', context)

def atbats(request):
    at_bats = Atbats.objects.all()[:50]
    context = {'at_bats': at_bats}
    return render(request, 'dashboard/atbats.html', context)
```

```python
def atbats(request):
    at_bats = Atbats.objects.all()[:50]
    context = {'at_bats': at_bats}
    return render(request, 'dashboard/atbats.html', context)

def atbats_detail(request, page_num):
    start_index = page_num * 50
    end_index = start_index + 50
    at_bats = Atbats.objects.all()[start_index:end_index]
    context = {'at_bats': at_bats}
    return render(request, 'dashboard/atbats.html', context)

def pitches(request):
    pitches = Pitches.objects.all()[:50]
    context = {'pitches': pitches}
    return render(request, 'dashboard/pitches.html', context)

def pitches_detail(request, page_num):
    start_index = page_num * 50
    end_index = start_index + 50
    pitches = Pitches.objects.all()[start_index:end_index]
    context = {'pitches': pitches}
    return render(request, 'dashboard/pitches.html', context)

def pitches(request):
    pitches = Pitches.objects.all()[:50]
    context = {'pitches': pitches}
    return render(request, 'dashboard/pitches.html', context)

def pitches_detail(request, page_num):
    start_index = page_num * 50
    end_index = start_index + 50
    pitches = Pitches.objects.all()[start_index:end_index]
    context = {'pitches': pitches}
    return render(request, 'dashboard/pitches.html', context)
```

iii. Have you implemented any DML components (INSERT, UPDATE, DELETE) yet in your project (Note that you must have DML components in the final version of your project). If yes, include screenshot.

Because we are working with historical (static) data and performing analytics using that data, there is not much actual manipulation that we are doing. Specifically, inserting, updating, or deleting rows from the tables representing our original data sets could possibly affect the accuracy of our analytics. Therefore, as of right now, we have not implemented any DML components. However, we may create some functionality that can allow for new data to be inserted into the existing tables, in which case we would implement INSERT and UPDATE capability.

f. What is the status of front-end application? In terms of percentage how much do you think, you have completed on front-end side of the project? Describe what is completed and what is left. 2-3 sentences.

The front-end application is in good shape, as we have written the majority of the code for all of our necessary page templates (useable for all different query result sets). We have also set up routing and navigation, so all compoonents of the application are connected. Because we have not yet completed all of the desired styling and UI functionality (i.e. page navigation, filtering, sorting, etc.), we would say we are about 70% done with the front-end side.

Report and demo video

- Have you started working on your report? If yes, how much is completed (percentage)

We have not yet begun work on the final report, but we have made sure to maintain documentation of what we have accomplished so that when we get to the report, we will have all the necessary info.

- Which tool you are planning to use to create a demo video of your application?

We will most likely create a screen recording using QuickTime and a voice over using Voice Memos to demo our application.

Final Submission

- Please read the final project submission grading criteria posted on Brightspace.

Submission:

Complete this document and save it as pdf. You must submit a PDF file named **p2-part3-lastname1-lastname2.pdf** (For example if I submit this document with John Smith, I would name it p2-part3-singh-smith.pdf). Submit your files on Brightspace.

You must include your name and your partner name in the Brightspace submission text box.

Each member of the team must make the submission of same file.

Grading:

This Assignment will be graded on the following criteria:

1. Completeness of document.
2. Completeness of required components at this stage of project.
3. Clear evidence of work completed.

NO grading will be done on file/s sent through email or not uploaded to Brightspace.