Events app documentation

Table of Contents

Introduction

Aims and objectives of the project

Our Events web app is intended to simplify finding out what's available in select cities within a specific category of events and specified date range. The user is able to add their own likes to the list of events so these are updated accordingly in the "Event details" page. This page also displays the location of the event venue on Google Maps, which the user can click into to access Google Maps directly and plan their route to the venue.

Roadmap of the project

1. Discussed potential features of the project and decided on core ones we wanted to focus on
2. Realised that all project ideas hinged on the quality of the API so set out researching API options and discussed pros and cons of each, mainly regarding accessibility and what the API could return.
3. Selected best API fit for purpose (in this case, Ticketmaster)
4. Discussed everyone's strengths and desired contributions to the project and allocated tasks for completion accordingly
5. Agreed to meet up regularly on Zoom and share updates on Slack to discuss next steps/distribution of work
6. Set up project in Flask
7. Built UI
8. Created database in SQL Workbench to later connect to web app
9. Connected Ticketmaster API to web app
10. Worked on writing unit tests
11. Worked on connecting the database to the API

12. Wrote project documentation

Background

The idea for the web app was originally Yasemin's, which she posted on the group in Slack. It was one of three and in the first meeting when the shape of the app/project was taking place, Patricia discussed how she really likes going to live events in London (especially theatre) so this idea piqued her interest. Patricia is often asked by friends who know she likes seeing events for recommendations of what to see or suggestions for what to do on a particular day. The way Patricia normally finds her events is by searching on the theatres' websites, which she knows offer cheap tickets from having previously visited them, but this is obviously very time-consuming as it involves opening individual different theatre websites to get an overview of what's on and available. The Events app allows users to find out what's going on for a wide range of events and in different cities over the next seven days all from one website, using the Ticketmaster API.

Originally, Yasemin had based her idea on the London Theatre Direct API but this website didn't have a reliable, open system where we could investigate the data freely. We wanted to get started on the project as soon as possible and didn't have the time to email over the weekend when we discussed investigating the API and wait for a response to access the key to use the API. We then decided to research other APIs that met our criteria and would allow us to make unlimited calls so that we were not restricted by this. Several APIs were discussed and found to be advantageous for different reasons, but we finally settled on the Ticketmaster API as this seemed very extensive and met all our requirements for the project. In the Homework assignment, our instructor confirmed that this API looked good and was well-documented so we pressed forward using this one. After our first meeting, we realised that the entire project would hinge on the quality of the API so knew it was important to finalise this part of the project before making a start.

Bosa was keen to compare the features of our app with other similar sites to see how we could differentiate ours from the others on the market. Having looked at LOVETheatre, the only filter option is by event type. WhatsOnStage allows users to filter by city. TodayTix allows you to filter by city and event type. All of these offer the option of viewing events that have a special offer on/fall under a certain price range. None of these allows you to filter by a specific budget range (though TodayTix allows you to organise results in order of lowest to highest price) or date (though WhatsOnStage lets you filter by currently on, coming soon and closing soon) to get an overview of everything that's on offer, just featured shows. At the start of the project, we'd intended to be able to allow users to filter events by a price range but the API didn't allow for this option of filtering, so instead we adapted the app to filter by a range of dates.

In terms of specific details about the project, the app itself is very user-friendly and simple to use. As the project has been developed with a view to be assessed based on the specific project criteria, the deployment of the project was not made a priority. As such, the project was not optimised for deployment. If we'd had more time to work on this aspect, we were looking into using Heroku for this. At its current stage, the project can be viewed through local servers where the user views the website and has to choose three filters for their list of events from drop-down lists, which are event type (choice of music, sports, arts & theatre, film and miscellaneous – which seems to mainly be comedy), city (selection of London, Manchester, Leeds, Birmingham and Glasgow) and date range (today, tomorrow and next 7 days). Then either the user is told there are no events that meet the specified criteria or is provided with a list of events within the requested criteria. The user can see this list of events (up to a maximum of 100 events) that matches the selected requirements and includes the following information:

1. The event description/name
2. The event poster
3. The date and time of the event
4. The venue location address
5. A button with “More Details”

When the user clicks on the “More details” button for a specified event, they can access the same details as above (excluding the button), in addition to the event type, the city, and they also have the option of clicking the “Like” button to like that event. We had originally envisioned Finally, this page also includes the event venue’s location on Google Maps, allowing the user to click into the map to open Google Maps directly.

In creating the database, we wanted it to be as simple as possible with only three columns for event id, likes and dislikes (the original idea was to have the “Event details” page list the existing likes and dislikes for events with mock data; however despite best efforts, connecting the database to the web app was not successful (connection error) given limited time). Similarly, we created simple functions which would enable us to add the event id, likes and dislikes to the database as well as to update any records already held. It was also crucial that we be able to retrieve data from the database in order to display the amount of likes and dislikes within the web app.

Specifications and Design

Technical requirements

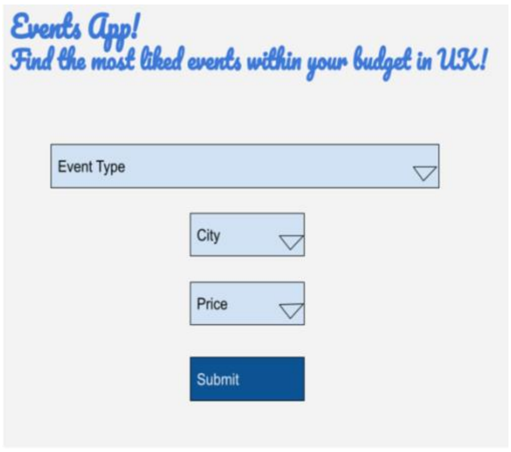
We used Flask and PyCharm and SQL Workbench to create the code for the project and connected to the Ticketmaster and Google Maps API

Design and architecture

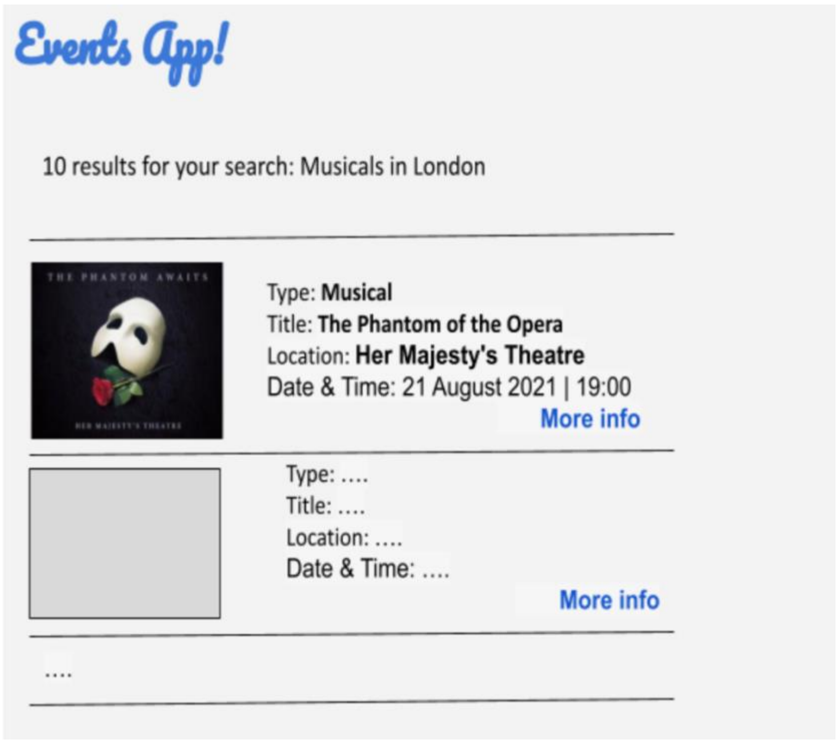
Most of the team members did not have any JavaScript or CSS experience, so Yasemin took on the task of creating the UI and designing the wireframe. It is functional and self-explanatory, without any distractions from the app’s core use. She also educated us on the functions she had implemented and how this would all come together to execute the project to light.

<u>Wireframes:</u>

Main/Search page:



Results page:


Events App!

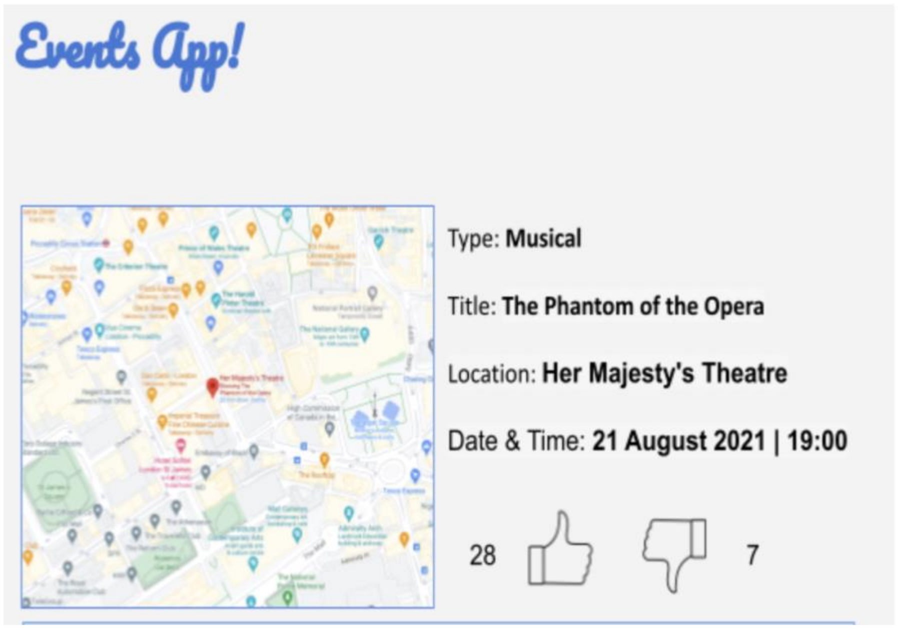
10 results for your search: Musicals in London

Type: Musical
Title: The Phantom of the Opera
Location: Her Majesty's Theatre
Date & Time: 21 August 2021 | 19:00
More info

Type: ....
Title: ....
Location: ....
Date & Time: ....
More info

....

Event Details page:


Events App!

Type: Musical

Title: The Phantom of the Opera

Location: Her Majesty's Theatre

Date & Time: 21 August 2021 | 19:00

28 7

Implementation and Execution

Development approach

We didn’t decide on a development approach before the project started, but we seemed to follow the Waterfall approach of trying to plan the project from start to finish and who would work on what and go away and do an initial phase before adding to that content to keep building on the project. However we also followed an Agile approach in terms of reacting to challenges/realising that the original plan needed adapting according to options that weren’t possible that we had initially envisioned. Once the Ticketmaster API connection was established, we all tried to work on tasks simultaneously, so it could be argued we took a blended approach of Agile and Waterfall given that we did not have a Scrum master or sprints and tried to plan the project at the start and work through the different phases but ended up having to react to planned features not working as expected/being possible and adapt the plan as we went along, while also working on different aspects of it simultaneously.

Team member roles

All team members discussed and agreed on the final project idea. Team members research different API possibilities and decided to work with the Ticketmaster API

Yasemin: project set up, UI creation, connection to Ticketmaster and Google Maps API

Hannah: Database creation, exception handling and unit testing

Patricia: Documentation and unit testing

Lyondra: Database connection to the app and unit testing

Bosa: Exception handling of the API. Worked with Yasemin on the Ticketmaster API connection and with Lyondra on connecting the database to the app

Tools and libraries used

- Ticketmaster API,
- PyCharm,
- SQL Workbench,
- Flask,
- GitHub,
- JavaScript,
- HTML,
- CSS and
- datetime library.
- In terms of project management, we considered using Miro and Metro Retro, at the start, but we found Google docs and Slack were more convenient to update each other regarding the progress of the project.

Implementation process

Achievements

- The web app is usable and displays events as desired!!
- It is simple and easy to use and the “bonus” Google Maps was able to be embedded.
- We pulled the project together in a relatively short amount of time due to long discussion stages of desired end product, different commitments/availabilities and demanding course content.
- We managed to consolidate learning done during the course

- We gained knowledge and experience of how larger projects work and how elements interact with each other

Challenges

- It took quite some time to finalise the end project concept and how it would work/who would work on what and this had to be adjusted according to requirements
- Connecting the SQL database to the web app for the number of likes and dislikes to display on the "Event Details" page proved more difficult than anticipated with several people attempting it without success within the limited time planned for this task.
- Writing unit tests (especially to test specific parts of objects rather than entire objects)
- Exception handling (we think the errors raised below in the screenshots are due to the API not including the location for some events in the requested search and thus throwing this error up)
- Navigating GitHub (we all felt our knowledge and experience of GitHub was very limited and therefore struggled with making changes on our ends of the project and having to constantly pull the project again shortly after the latest version was pushed.)
- Took a while for everyone to set up the project environment with people having different systems and versions installed
- Sometimes agreeing on times to meet and agenda items/tasks for completion proved tricky
- External concerns, such as course content, homework submission deadlines and exam following the project submission deadline in addition to other personal commitments such as work and personal events.
- Difficult to work as a team, for any future projects it would be useful to have a more structured approach such as Agile and make use of Kanban tools such as Metro retro
- Establishing everyone's strong points was quite hard as this was our first big project and we didn't feel confident on things we could execute.

Desires to change things along the way

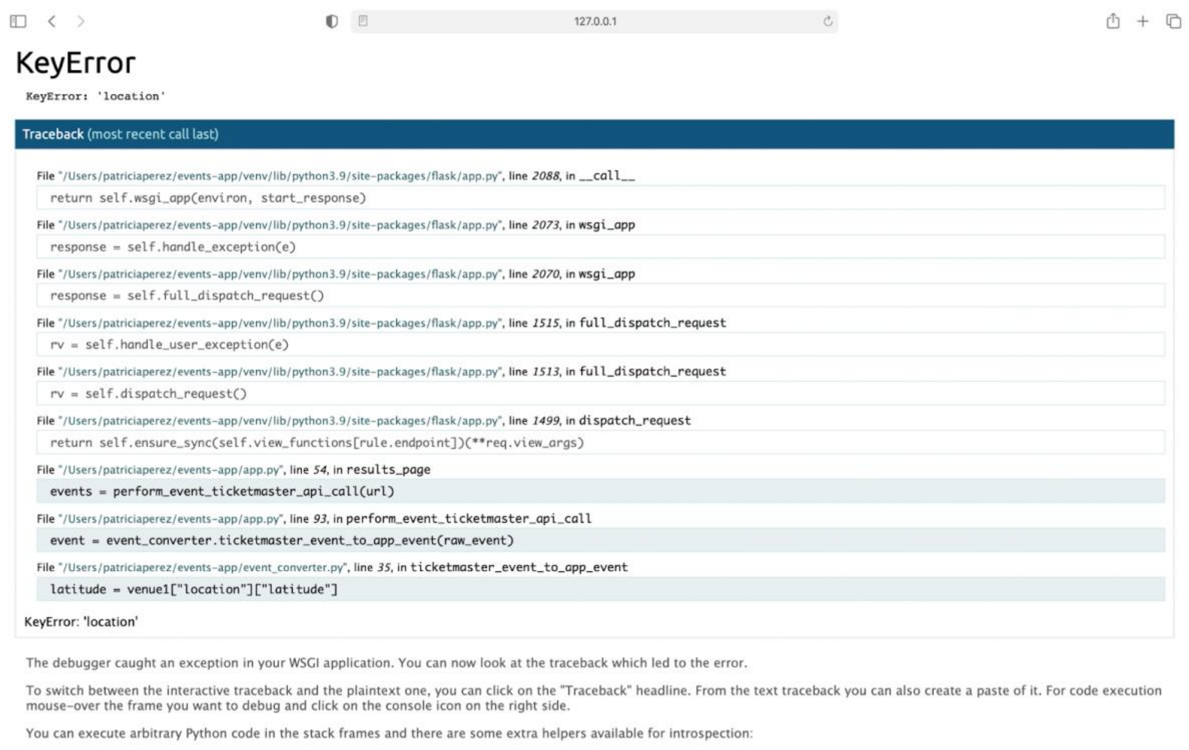
- Amended price range filter to date/time filter due to API limitations
- Today & Tomorrow button on the search page would ideally just say "Tomorrow"
- The date and time information was used as a string rather than datetime data type, which makes it difficult to use the data as flexibly as desired.
- Due to time limitations, scrapped idea of including Twitter sentiment analysis of the events listed in the "Event Details" page
- If had had more time, would have liked to deploy the web app through Heroku.

Testing and Evaluation

What was the testing strategy?

From all the project requirements, the testing was the aspect we found most challenging. All members of the team tried writing unit tests for different functions and classes, but only Yasemin was able to get hers to work and we were all able to work together based on her suggestions. Despite asking for advice from our instructor, we all found this incredibly difficult and struggled to understand how to approach it. We did however try out all the options possible on the web app to see what results were obtained and clicking through a random sample of these to check everything worked as expected with the "Event details" page. Some of the errors on the page we noticed quite late while doing user testing, but we were able to fix them with basic try / except exception handling. (see below)

For Arts & Theatre, London and Next 7 Days, it raised a location error:


127.0.0.1

KeyError

```
KeyError: 'location'
```

Traceback (most recent call last)

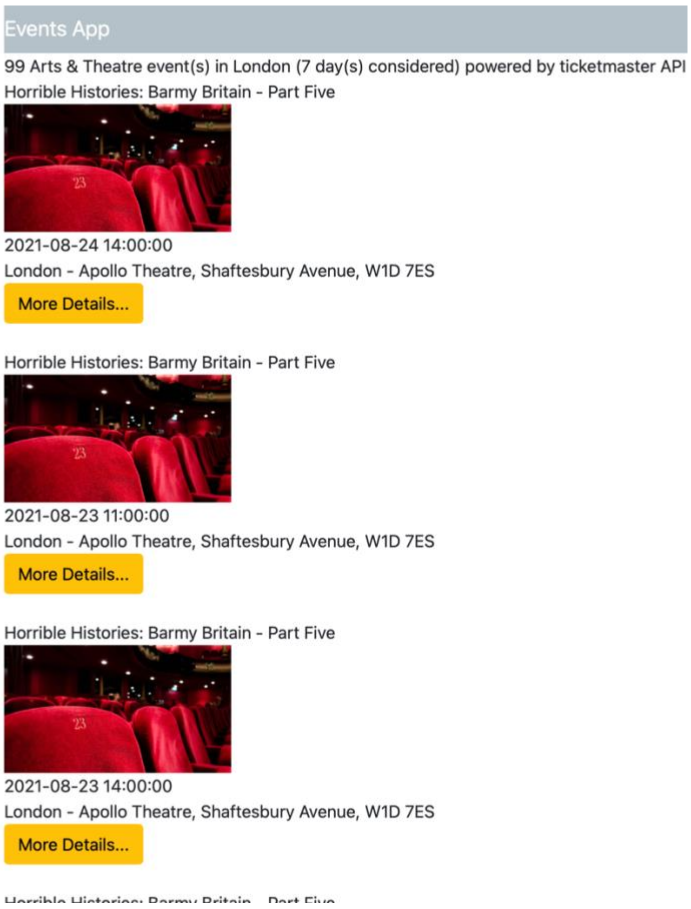
```
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2088, in __call__
  return self.wsgi_app(environ, start_response)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2073, in wsgi_app
  response = self.handle_exception(e)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2070, in wsgi_app
  response = self.full_dispatch_request()
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1515, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1513, in full_dispatch_request
  rv = self.dispatch_request()
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1499, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
File "/Users/patriciaperez/events-app/app.py", line 54, in results_page
  events = perform_event_ticketmaster_api_call(url)
File "/Users/patriciaperez/events-app/app.py", line 93, in perform_event_ticketmaster_api_call
  event = event_converter.ticketmaster_event_to_app_event(raw_event)
File "/Users/patriciaperez/events-app/event_converter.py", line 35, in ticketmaster_event_to_app_event
  latitude = venue1["location"]["latitude"]
KeyError: 'location'
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

Following the exception handling:


Events App

99 Arts & Theatre event(s) in London (7 day(s) considered) powered by ticketmaster API

Horrible Histories: Barmy Britain - Part Five

2021-08-24 14:00:00

London - Apollo Theatre, Shaftesbury Avenue, W1D 7ES

More Details...

Horrible Histories: Barmy Britain - Part Five

2021-08-23 11:00:00

London - Apollo Theatre, Shaftesbury Avenue, W1D 7ES

More Details...

Horrible Histories: Barmy Britain - Part Five

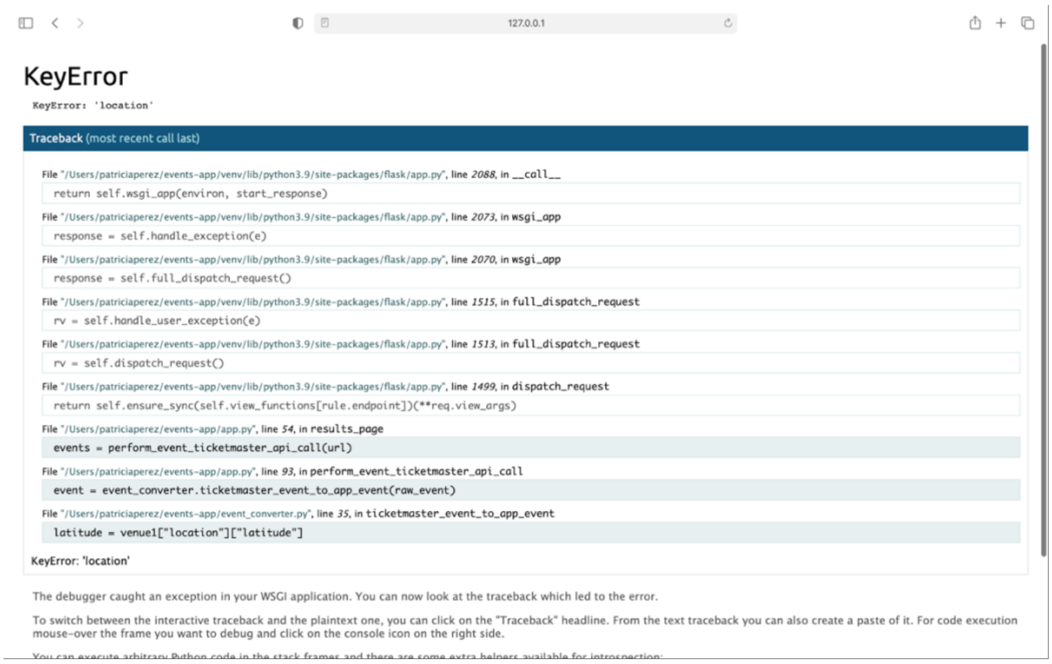
2021-08-23 14:00:00

London - Apollo Theatre, Shaftesbury Avenue, W1D 7ES

More Details...

Horrible Histories: Barmy Britain - Part Five

Similarly with Music, Manchester in the Next 7 days:



127.0.0.1

KeyError

```
KeyError: 'location'
```

Traceback (most recent call last)

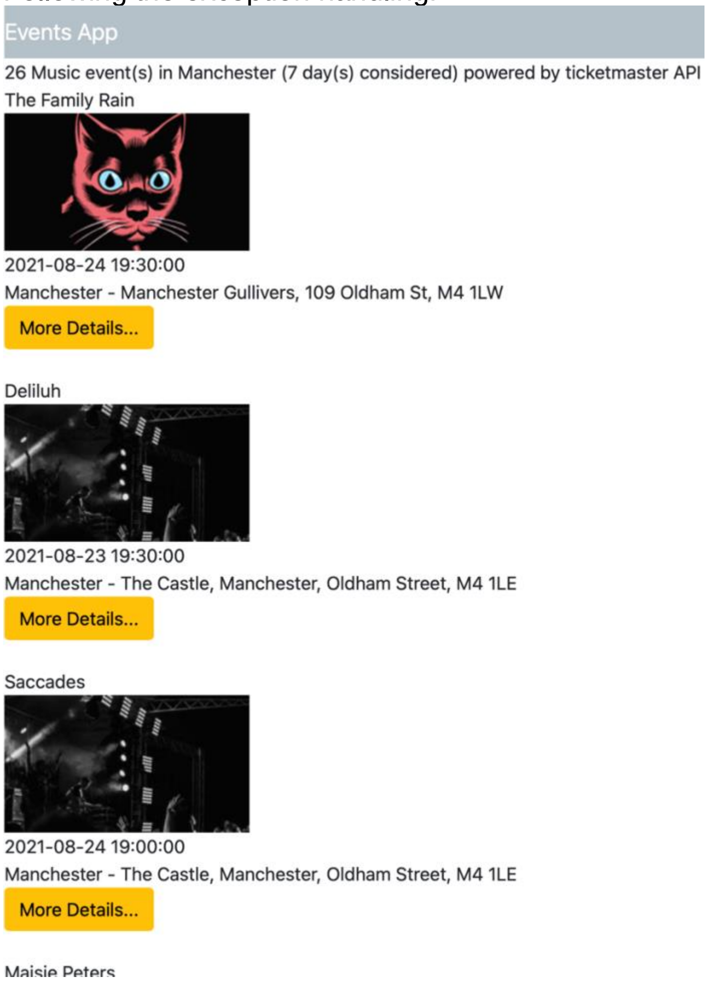
```
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2088, in __call__
    return self.wsgi_app(environ, start_response)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2073, in wsgi_app
    response = self.handle_exception(e)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2070, in wsgi_app
    response = self.full_dispatch_request()
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1515, in full_dispatch_request
    rv = self.handle_user_exception(e)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1513, in full_dispatch_request
    rv = self.dispatch_request()
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1499, in dispatch_request
    return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
File "/Users/patriciaperez/events-app/app.py", line 54, in results_page
    events = perform_event_ticketmaster_api_call(url)
File "/Users/patriciaperez/events-app/app.py", line 93, in perform_event_ticketmaster_api_call
    event = event_converter.ticketmaster_event_to_app_event(raw_event)
File "/Users/patriciaperez/events-app/event_converter.py", line 35, in ticketmaster_event_to_app_event
    latitude = venue1["location"]["latitude"]
KeyError: 'location'
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

Following the exception handling:



Events App

26 Music event(s) in Manchester (7 day(s) considered) powered by ticketmaster API

The Family Rain

2021-08-24 19:30:00

Manchester - Manchester Gullivers, 109 Oldham St, M4 1LW

Deliluh

2021-08-23 19:30:00

Manchester - The Castle, Manchester, Oldham Street, M4 1LE

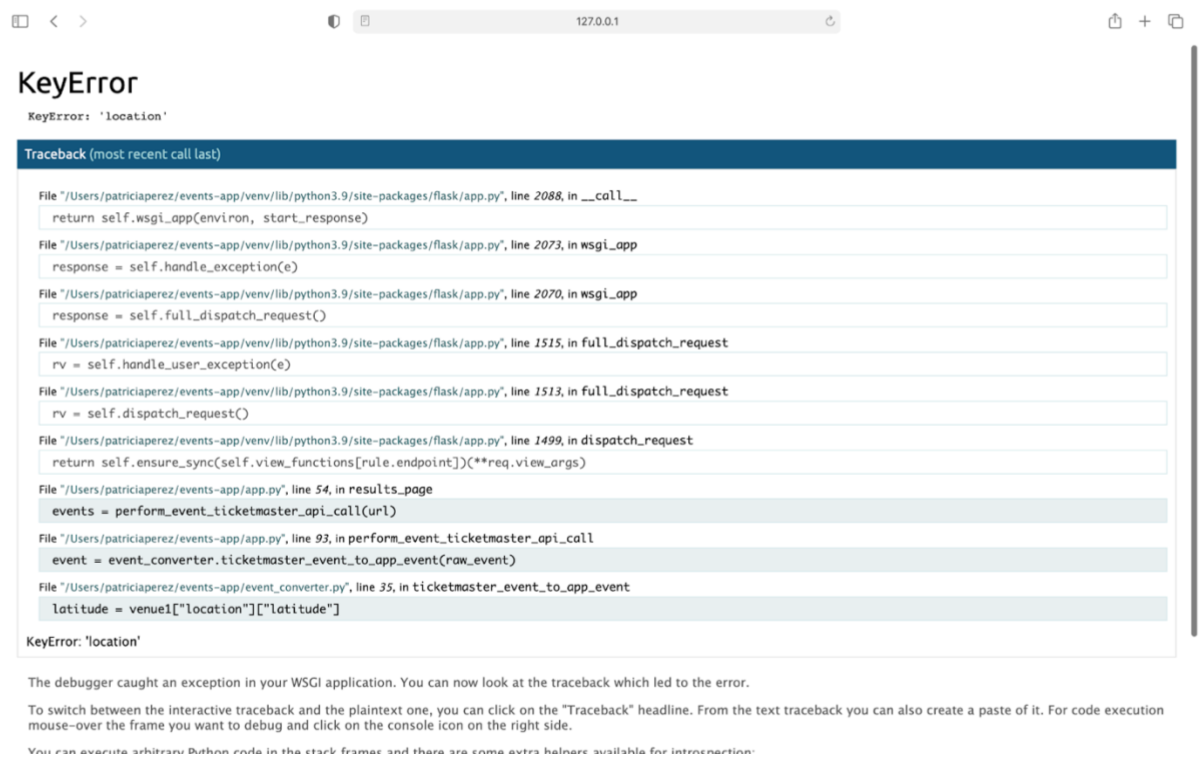
Saccades

2021-08-24 19:00:00

Manchester - The Castle, Manchester, Oldham Street, M4 1LE

Maisie Peters

And the same with Music, Glasgow and Next 7 Days:



127.0.0.1

KeyError

```
KeyError: 'location'
```

Traceback (most recent call last)

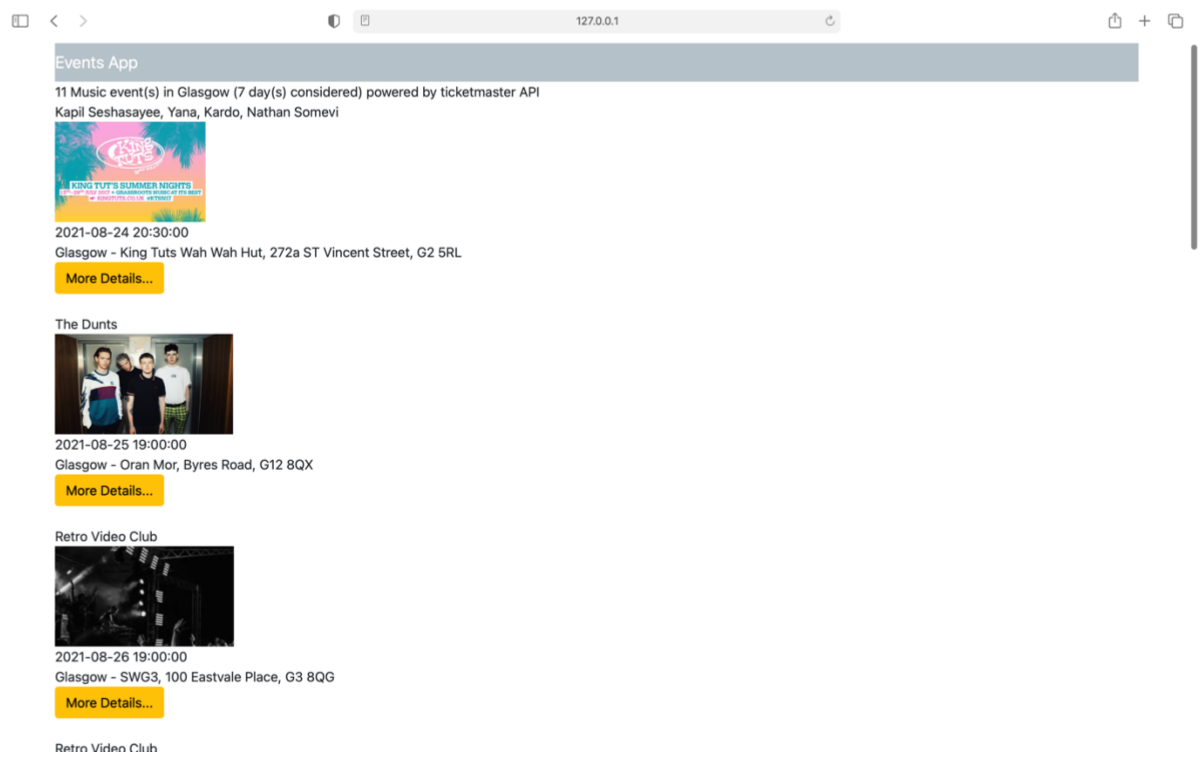
```
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2088, in __call__
  return self.wsgi_app(environ, start_response)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2073, in wsgi_app
  response = self.handle_exception(e)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 2070, in wsgi_app
  response = self.full_dispatch_request()
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1515, in full_dispatch_request
  rv = self.handle_user_exception(e)
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1513, in full_dispatch_request
  rv = self.dispatch_request()
File "/Users/patriciaperez/events-app/venv/lib/python3.9/site-packages/flask/app.py", line 1499, in dispatch_request
  return self.ensure_sync(self.view_functions[rule.endpoint])(**req.view_args)
File "/Users/patriciaperez/events-app/app.py", line 54, in results_page
  events = perform_event_ticketmaster_api_call(url)
File "/Users/patriciaperez/events-app/app.py", line 93, in perform_event_ticketmaster_api_call
  event = event_converter.ticketmaster_event_to_app_event(raw_event)
File "/Users/patriciaperez/events-app/event_converter.py", line 35, in ticketmaster_event_to_app_event
  latitude = venue1["location"]["latitude"]
KeyError: 'location'
```

The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

Following the exception handling:



127.0.0.1

Events App

11 Music event(s) in Glasgow (7 day(s) considered) powered by ticketmaster API

Kapil Seshasayee, Yana, Kardo, Nathan Somevi

2021-08-24 20:30:00

Glasgow - King Tuts Wah Wah Hut, 272a ST Vincent Street, G2 5RL

The Dunts

2021-08-25 19:00:00

Glasgow - Oran Mor, Byres Road, G12 8QX

Retro Video Club

2021-08-26 19:00:00

Glasgow - SWG3, 100 Eastvale Place, G3 8QG

Retro Video Club

System limitation

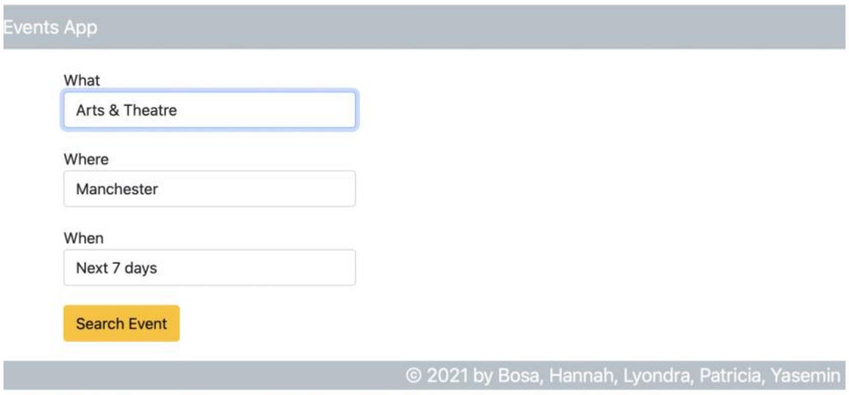
We all used Python version 3.6+

Conclusion

We had a lot of exciting ideas to begin with and it took us a little while to agree/finalise exactly what we were trying to aim for with the final project, which meant we ended up having less time to work on the code for the project so we weren't able to implement everything we would have liked. Additionally, the API didn't have all the parameters we initially thought it did to allow users to filter by price so we had to adapt and filtered by date instead. We needed some guidance from the instructor based on our project homework submission in order to get a clearer idea of how to actually start working on the project and divide up the steps required to try and make our idea a reality. It's been an uphill battle but given the limited time we had in the end, a lot has been achieved and it does the core of what we set out to do in addition to connecting to the Google Maps API which was a bonus! In future, having this project experience would be really useful in terms of knowing what steps to follow to start off and deciding how to divide up the tasks for completion. With different availabilities of team members to work on the project and to meet, the end result is something to be proud of and a huge achievement!

You can view the end result of our web app below and see that it is not too dissimilar from the wireframes we originally envisioned:

Main/search page:



Events App

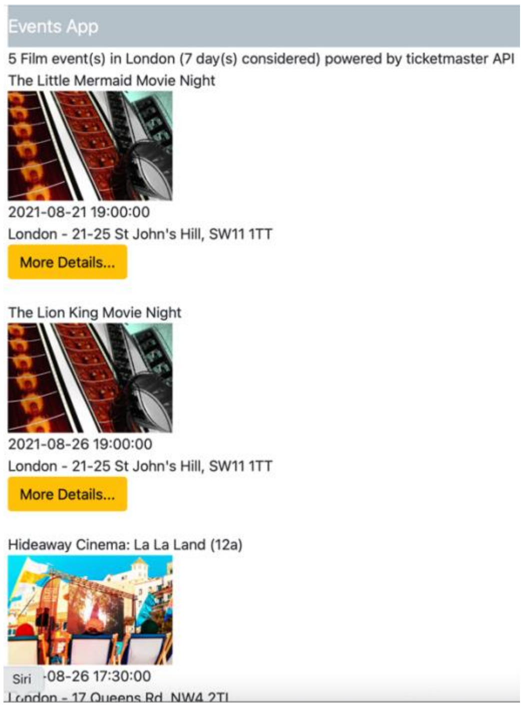
What: Arts & Theatre

Where: Manchester

When: Next 7 days

© 2021 by Bosa, Hannah, Lyondra, Patricia, Yasemin

Results page:



Events App

5 Film event(s) in London (7 day(s) considered) powered by ticketmaster API

The Little Mermaid Movie Night

2021-08-21 19:00:00

London - 21-25 St John's Hill, SW11 1TT

The Lion King Movie Night

2021-08-26 19:00:00

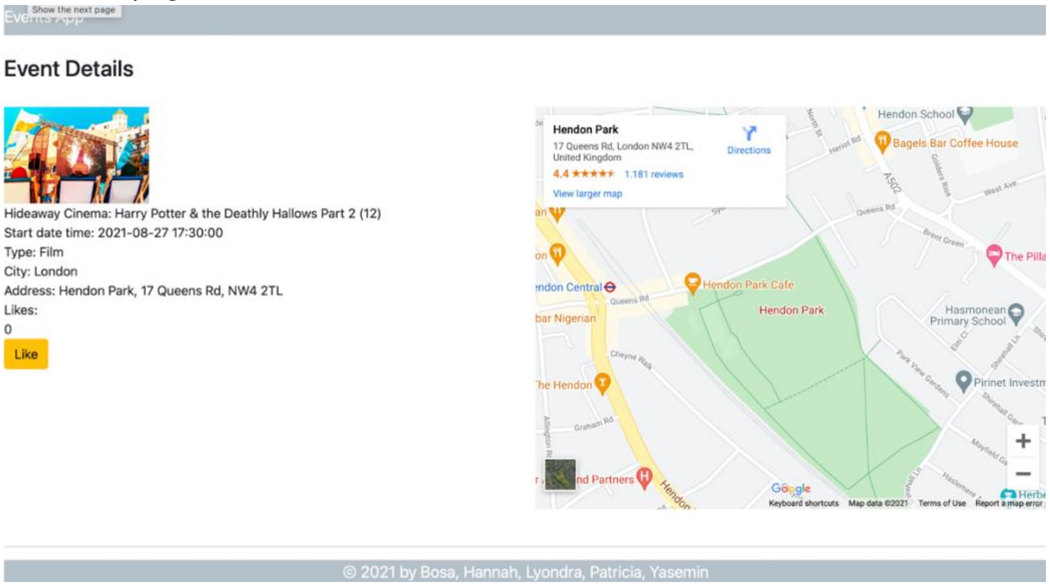
London - 21-25 St John's Hill, SW11 1TT

Hideaway Cinema: La La Land (12a)

-08-26 17:30:00

London - 17 Queens Rd, NW4 2TL

Event details page:



Events App

Show the next page

Event Details

Hideaway Cinema: Harry Potter & the Deathly Hallows Part 2 (12)
Start date time: 2021-08-27 17:30:00
Type: Film
City: London
Address: Hendon Park, 17 Queens Rd, NW4 2TL
Likes:
0

Like

Hendon Park
17 Queens Rd, London NW4 2TL, United Kingdom
4.4 ★★★★★ 1,181 reviews
View larger map
Directions

© 2021 by Bosa, Hannah, Lyondra, Patricia, Yasemin