

Event Hub App

Sharing Medium for Popup Stores / Events

Senior Project Presented To

the Faculty of the Liberal Arts and Engineering Studies Department

California Polytechnic State University, San Luis Obispo

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science in Liberal Arts and Engineering Studies

By

Coby Hong

June 2021

Table of Contents

Introduction	3
Application / Product	3
Background	4
Design	5
Implementation	6
Analysis / Verification	7
Interdisciplinary Connections	7
Related Work	8
Future Work	8
Conclusion	8
Bibliography	9

Introduction

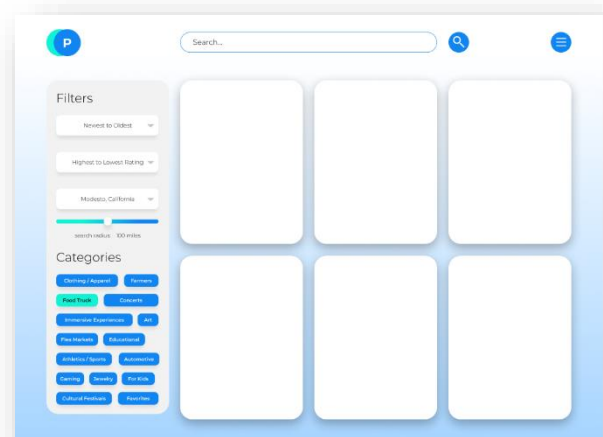
For my senior project I wanted to focus primarily on the technical aspects of Full Stack while also possibly introducing a newer medium for smaller, niche businesses. Looking into the field that my application will target, I found that most popup businesses rely on heavy social media activities to promote awareness of their event with most of these stores performing a “30% increase in social media activities... attributing to 35% of their sales (Leinbach-Reyhle 2014 #1), from their own perceptions. This means that the key to success for most popups seem to be from word of mouth.

Researching the general offerings of platforms, I have found that social media relies on interconnections or pre-followed tags to gain viewability of certain popup events and other more restaurant applications like Yelp focus primarily on permanent / location-based establishments. So, to gain knowledge of a popup event, you must already be within some form of a social circle or research independently to find.

I want to resolve this conflict by creating a central hub / application, completed dedicated to announcing any form of popup event. My solution will provide a more concrete medium for these much smaller businesses to advertise themselves and allow those who actively seek popups have a easier time doing so. I believe my work will be needed as it seems that there has been a general rise in more “mom and pop” shops as prior to 2020, “The global convenience, mom and pop stores market reached a value of nearly \$960.8 billion in 2019, having grown at a compound annual growth rate (CAGR) of 5.5% since 2015, and is expected to grow at a CAGR of 5.7% to nearly \$1,201.3 billion by 2023 (GlobeNewswire 2019 #2). Popup event / stores fall into this category and believe my medium solution would align well with this currently growing market.

Application / Product

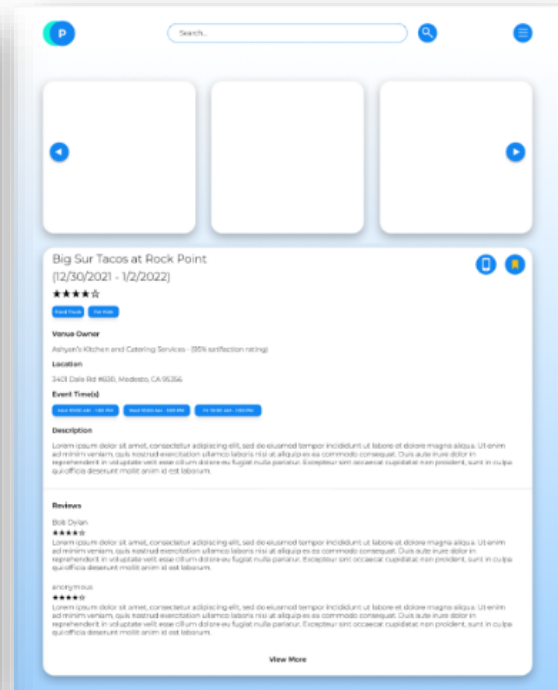
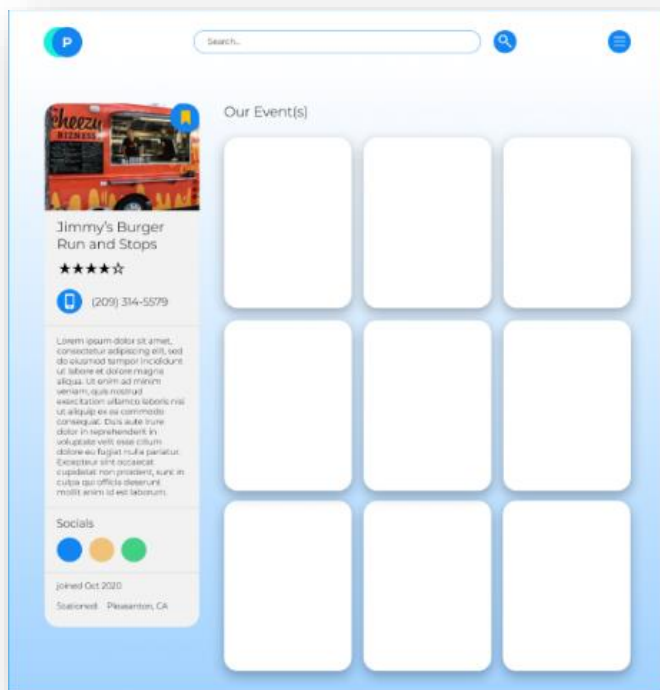
My application will be like a Facebook feed, but dedicated entirely to popup events / stores. Users will be able to view listed events and create their own. Created event cards will have attributes of location, rating, description, images, owner, reviews, and occurrence. Cards will only be available on news feed until their finish date becomes greater than current date. Basically, the popup event cards will delete themselves once expired. Users will be able to filter through popup event cards given search and filter functionalities.



Users will be able to login through Google. Through accounts, users will have the simple ability to save their favorite events and keep track of events they have posted themselves. I planned on adding a commenting system, but this did not come into fruition. A way to think of this user structure is like the site craigslist. Instead of posting items for sale, users post their popup events. Other users can view the popup events through the news feed and offer up either reviews or ratings. Their critique leading to the overall review of the user who posted and a determinant for future filter applications from others using the application. In short, a newsfeed representing a list of popup events.



The listings are represented on individual cards detailing date duration, location, rating, title, and tags. Expanding / clicking on the card will lead to a more detailed listing of the event. Users will have the ability to filter these cards as well as review and rate them. These rating and reviews ultimately influencing the newsfeed filter process as well as the event owner's profile record.



Background

To better understand how to get into creating the application, I investigated preferable stacks that would work well for sites that focused on listing. I as well considered budget as I am working with a nonexistent one. I found most preferably that any variation of the MERN stack (MongoDB and React) would suffice as React is a popular and well documented front-end framework to build scalable web applications off. MongoDB is a json-based database that is relatively easy to setup and handles the information quite easily since most data is just strings / text. Most of my knowledge of the stack came from taking CSC-307 as an elective which gives you general understanding of software engineering workflow and basic usage of MERN-type stacks.

I gained greater understanding of the technologies through YouTube videos primarily from the channel “Web Dev Simplified” which helped with setting up both my front and back end. In general, the best approach I found to understanding the processes is through simply googling and asking on help forums such as Programming discord. I good understanding for web layout came from this article: (<https://css-tricks.com/snippets/css/complete-guide-grid/>) which provides introductory information on how to plan the landscape of your web application. Essentially, creating grids on a page and putting components into those grids. For understanding styling, I attained most of my understanding through examples on (https://www.w3schools.com/html/html_css.asp). In addition to custom styling, I used packages from bootstrap-react to expedite the beautifying process.

Most of the backend information and understanding I attained from my coursework in CSC-307 and various YouTube tutorials. I would advise looking at the documentation on the MongoDB website for connecting to their atlas service. I good resource for initial design would be Figma, a free mock-up maker. There are services to convert Figma designs into React code, but that requires a paid service and I found it would be more beneficial to myself to learn how most people in the industry do it.

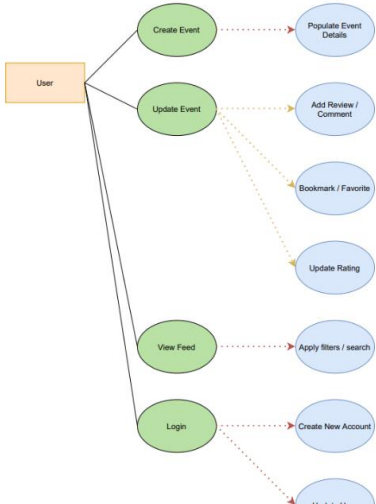
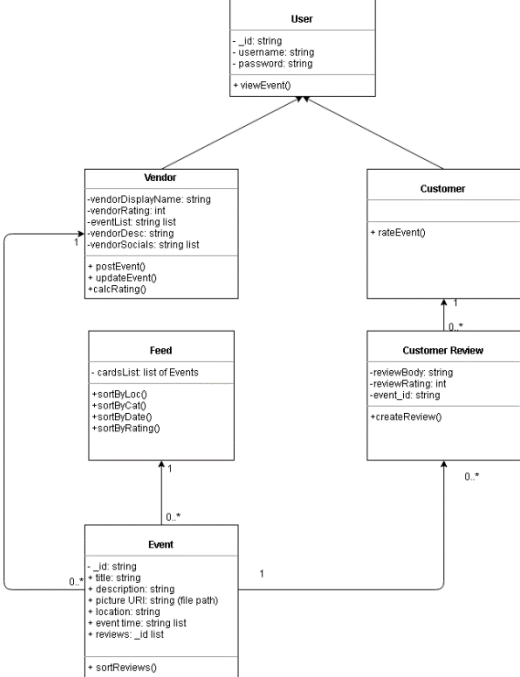
For later development of user accounts, I garnered most of my knowledge from the NPM documentation site: (<https://www.npmjs.com/package/react-google-login>). This site mainly detailing that to use Google Login services, one must have the NPM package installed and an API-key. The API-key acquired through creating a new project in Google API Cloud. Base code is provided to get started within the package’s documentation pages. Essentially, using their React component will return a user token unique to that individual.

Design

The first steps entailed mock-ups in Figma to understand what I want visually, as well as components that would be needed. I essentially knew that the foundation of my project will be on the application’s ability to display, post, remove, and change a list of cards. I wanted the design to be easy to follow from a user standpoint so followed pre-existing design components which were cards. Cards being widely used on Facebook, OfferUp, and Yelp.

Essentially, most of the design will follow on making sure the major components of the project work and building additional functionality off it such as user login and review system. My process has essentially followed initial Figma mockups, after the mockups doing a general UML diagrams to know what types of data structures and functions, I would essentially need for this application to work. Luckily, my project mirrors the skeletal structure of a simple “ToDo” application, so the structure is not complicated.

From there, I generated the backend with very basic front-end components for testing and later styled them appropriately. I made sure to follow basic UI/UX principles that would make identifying and using components simple to users hence familiar card-styled listings and general filter functionality. The Card components working best as well for transitioning into a mobile platform / view.



Implementation

For the backend, I am using MongoDB atlas. This is a json-type database stored on MongoDB's cloud server. This server works by essentially creating "clusters" which are data tables in your online database. Accessing the database requires essentially a URL key that you input into your mongo client manager. You can index to whatever table you want to access like a dictionary key in Python.

To connect to the MongoDB database, I am using a REST API. I installed flask which is essentially just a python framework that allows for http requests. From there, I installed the necessary pymongo packages to allow for interaction with MongoDB services. Connecting to the client simply being to use the built-in pymongo connectors with your given MongoDB URL key. From there, I establish basic http request / API calls that my front end will use.

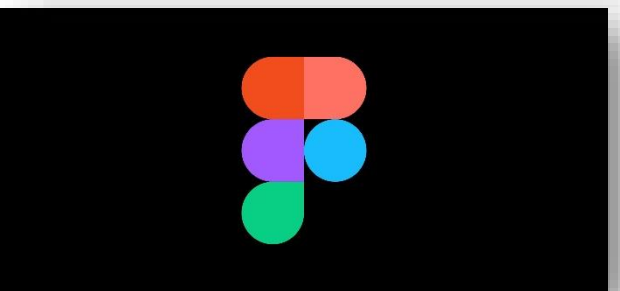
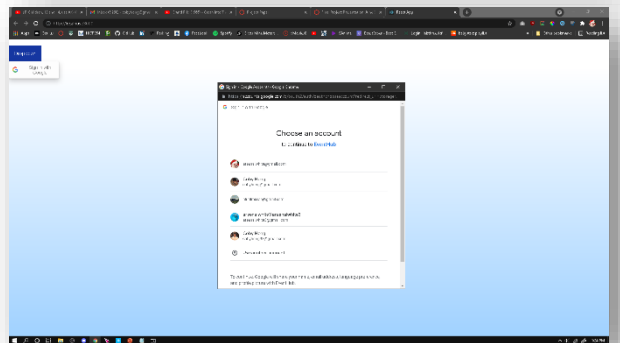
For the front end, I am simply using React. Any actions towards the database will make calls to my REST API http calls with the axios library. These calls then performing the basic alterations of the MongoDB database. User account login was done using the Google Login Service API as described previously.

For UI/UX planning / design, I am using Figma and just doing my layouts there. I use various websites and boilerplates to transition my design into actual React code.

For more detailed explanation, I currently have a cluster in my MongoDB database called cards. The cards cluster hold json objects. From the front-end, the listings are represented as a list / array of objects. When I want to add an object to the database, I make a POST call that grabs

```
makePostCall(character){
  return axios.post('http://localhost:5000/users', character)
  .then(function (response) {
    console.log(response);
    return (response.status === 201);
  })
  .catch(function (error) {
    console.log(error);
    return false;
  });
}
```

```
@app.route('/users', methods=['GET', 'POST'])
def get_users():
    if request.method == 'GET':
        #http://127.0.0.1:5000/users?name=Test&job=asdf
        search_username = request.args.get("name")
        search_job = request.args.get("job")
        if search_username and search_job:
            # TODO: Replace with database access
            #result = find_users_by_name_job(search_username, search_job)
            result = User().find_by_name_and_job(search_username, search_job)
        elif search_username:
            # TODO: Replace with database access
            #result = [user for user in users['users_list'] if user['name'] == search_username]
            result = User().find_by_name(search_username)
        else:
            result = User().find_all()
            return ("users list": result)
    elif request.method == 'POST':
        userToAdd = request.get_json() # no need to generate an id ourselves
        newUser = User(userToAdd)
        newUser.save() # pymongo gives the record an "id" field automatically
        resp = jsonify(newUser), 201
        return resp
```

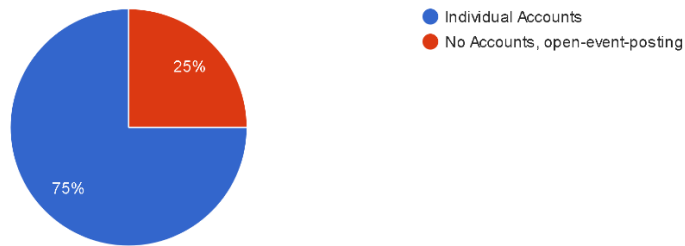


that object, and makes the following REST API call. That REST API call will take that object, turn it into a json object, and send it to the database.

Analysis / Verification

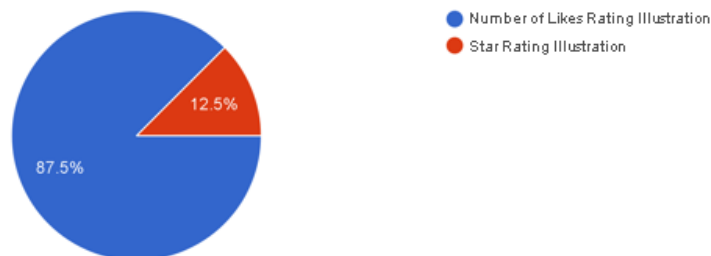
I performed a general survey questioning the design and functionality of the app to various forums. This would include sites like Reddit, The UI/UX Club Slack, and Web Development Discord I am a part of. I used input from these responses to further tweak my web application. My continuation of user accounts due to a 75% backing of the implementation. Generally, useability illustrations desired some form of hover or growth effect found pleasing and affirmative to the user. There was a desire from users for Google Login accounts as well.

A current discussion is whether their should be individual accounts or none. Individual accounts possibly leading to issues of vendor monopolies w...see reference below for account profile mockup).
8 responses



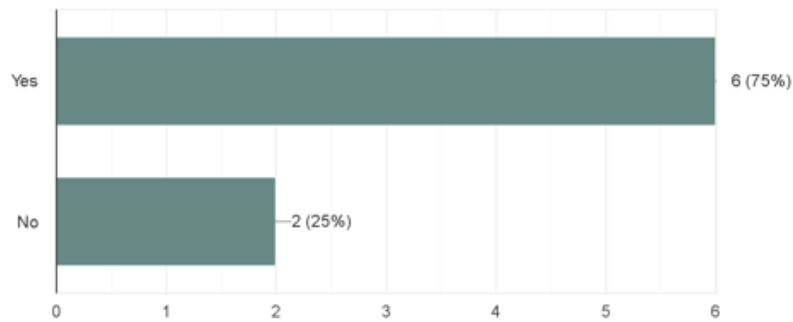
85.7% of responses found that the overall design of the application was representative of its cause. The given filter abilities of location, popularity, age, categories, and search name were seen to be sufficient by 75% of users. A surprising response was that users would rather see a “number of likes” rating over a star rating system.

A current discussion is whether the rating system for each Card “popup” event should be displayed / illustrated by number of likes, or by a ...rating. Which do you prefer? (See reference below).
8 responses



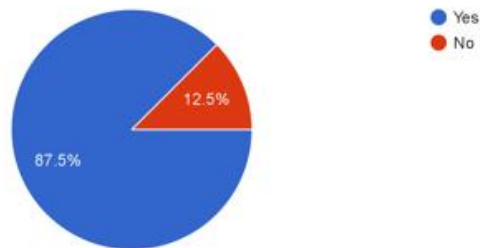
Do believe the current filter options are sufficient for the given application? (See above reference #1).

8 responses



EventHub App focuses on providing a digital medium / platform where small "popup" businesses and events can expose their activities to a populati...achieves this initiative? (See above reference #2).

8 responses



I as well contacted personal associates who have graduated and work in the field. My contacts at Twitter and Adobe who are both Full Stack developers guiding my design process and providing feedback on the effects they found most appealing with user navigation and satisfies general UI principles. They as well provided insight on the optimization of some of my functions and API requests. Testing on my own part confirms that the backend works properly with sending, receiving, and updating database information. The frontend very closely mirroring the Figma mockups. Below showcasing my mentors and other external help. The left side containing channels icons for Discord Programming and UX SLO Fest. The two photos on the right being my mentors.

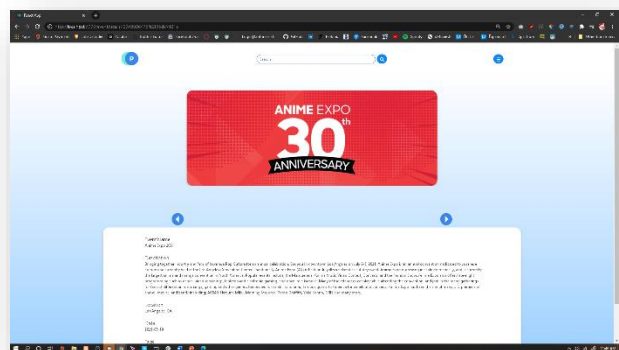
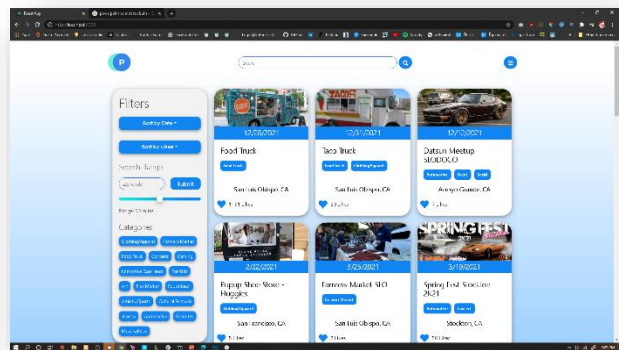
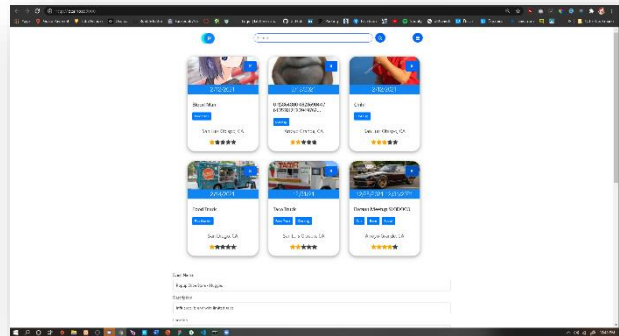


Interdisciplinary Connections

My work combines both the arts and computer science through the production process of a full stack engineer. As a full stack, I get to get into the design aspects of an application both visually and internally which work well with my GRC curriculum. My CS courses have given me the fundamentals of planning out the execution of a program as well as the general programming knowledge needed to create my works.

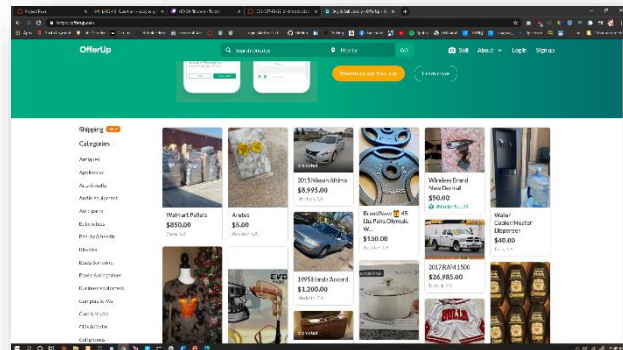
I found that GRC-201 played a key role in the design aspects of this project as I have a better notion for color palettes and element placement. I have gained graphic skills that have helped me with creating my own logos and supporting graphics to beautify this project. I as well got to gain knowledge through the UI/UX club targeted towards LAES students which has helped me understand the flow of user interfaces and good design practices. Full stack basically gets to go through the full process of a software application. I have gotten to research technologies that are prominent in the software engineering field, and practice workflows such as sprints and waterfalls to better manage my workload.

A key summary of this work can be represented by the LAES major itself. My graphic communications concentration has allowed me to visual express and illustrate my design ideas and gain input on what changes would benefit those given works. My Computer graphics concentration has given me the key fundamentals to programming to help create those visuals. I think the most essential thing for this LAES project stemming to the notion that a platform is essential for any group to get their ideas out.



Related Work

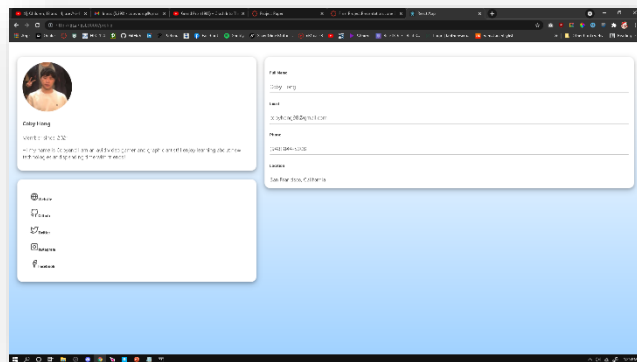
A heavy influence for my project comes from the site OfferUp. OfferUp is a platform for selling goods locally. It is like Craigslist, but with a much more friendly user-oriented design. I took most of my design patterns / user interface inspiration from their platform. However, unlike OfferUp, I will be transforming their product into a platform aimed at listing popup events, rather than actual objects for sale.



An inspiration for this was Facebook events. I liked the idea of being able to see local events that friends are attending. However, this was more of a side product from Facebook and not their main selling point. My application will focus purely on events that would fall into this category and nothing else. As well, these events will not be limited to a social circle but be available on a public medium news feed.

Future Work

I feel that I have gotten most of the functionality completed for this application though some aspects remain uncompleted. I still need to fully integrate the login system into the main web application. After this, I would like to allow logged users to bookmark and save their events. A possible future endeavor would be to add Google location maps into the expanded card view.



Conclusion

I feel my project has given me good understanding of the technologies needed for this tech industry. I have learned to approach design from the user's perspective and have learned that the progression of a project is never a fixed discipline. It is ever-changing with adapting ideologies and fixes to better represent the people who would use the product. Therefore, I got to adapt a more fluid workflow while also seeing how well I could implement those adaptations.

Bibliography

12, Samson Omojola onMar, et al. "CSS-Tricks." *CSS*, css-tricks.com/.

Haas, Sandra. "What Drives the Success of Pop-Up Stores?"

Https://Www.researchgate.net/Publication/301324724_What_Drives_the_Success_of_Pop-Up_Stores, Lisa Schmidt, Mar. 2016,

www.researchgate.net/publication/301324724_What_Drives_the_Success_of_Pop-Up_Stores.

Markets, Research and. "Global Market Opportunities in the Convenience, Mom and Pop Stores

Market to 2030: COVID-19 Impact and Recovery." *GlobeNewswire News Room*,

"GlobeNewswire", 28 Dec. 2020, [www.globenewswire.com/news-](http://www.globenewswire.com/news-release/2020/12/28/2150971/0/en/Global-Market-Opportunities-in-the-Convenience-Mom-and-Pop-Stores-Market-to-2030-COVID-19-Impact-and-Recovery.html#:~:text=The%20global%20convenience%2C%20mom%20and,nearly%20%241%2C201.3%20billion%20by%202023)

[release/2020/12/28/2150971/0/en/Global-Market-Opportunities-in-the-Convenience-](http://www.globenewswire.com/news-release/2020/12/28/2150971/0/en/Global-Market-Opportunities-in-the-Convenience-Mom-and-Pop-Stores-Market-to-2030-COVID-19-Impact-and-Recovery.html#:~:text=The%20global%20convenience%2C%20mom%20and,nearly%20%241%2C201.3%20billion%20by%202023)

[Mom-and-Pop-Stores-Market-to-2030-COVID-19-Impact-and-](http://www.globenewswire.com/news-release/2020/12/28/2150971/0/en/Global-Market-Opportunities-in-the-Convenience-Mom-and-Pop-Stores-Market-to-2030-COVID-19-Impact-and-Recovery.html#:~:text=The%20global%20convenience%2C%20mom%20and,nearly%20%241%2C201.3%20billion%20by%202023)

[Recovery.html#:~:text=The%20global%20convenience%2C%20mom%20and,nearly%2](http://www.globenewswire.com/news-release/2020/12/28/2150971/0/en/Global-Market-Opportunities-in-the-Convenience-Mom-and-Pop-Stores-Market-to-2030-COVID-19-Impact-and-Recovery.html#:~:text=The%20global%20convenience%2C%20mom%20and,nearly%20%241%2C201.3%20billion%20by%202023)

[0%241%2C201.3%20billion%20by%202023](http://www.globenewswire.com/news-release/2020/12/28/2150971/0/en/Global-Market-Opportunities-in-the-Convenience-Mom-and-Pop-Stores-Market-to-2030-COVID-19-Impact-and-Recovery.html#:~:text=The%20global%20convenience%2C%20mom%20and,nearly%20%241%2C201.3%20billion%20by%202023).

W3Schools Online Web Tutorials, www.w3schools.com/.

"Web Dev Simplified." *YouTube*, YouTube,

www.youtube.com/channel/UCFbNIlppjAuEX4znoulh0Cw.

"Where Developers Learn, Share, & Build Careers." *Stack*