Nathan Wemmer, Jacob Hanchett, Joseph Cochran

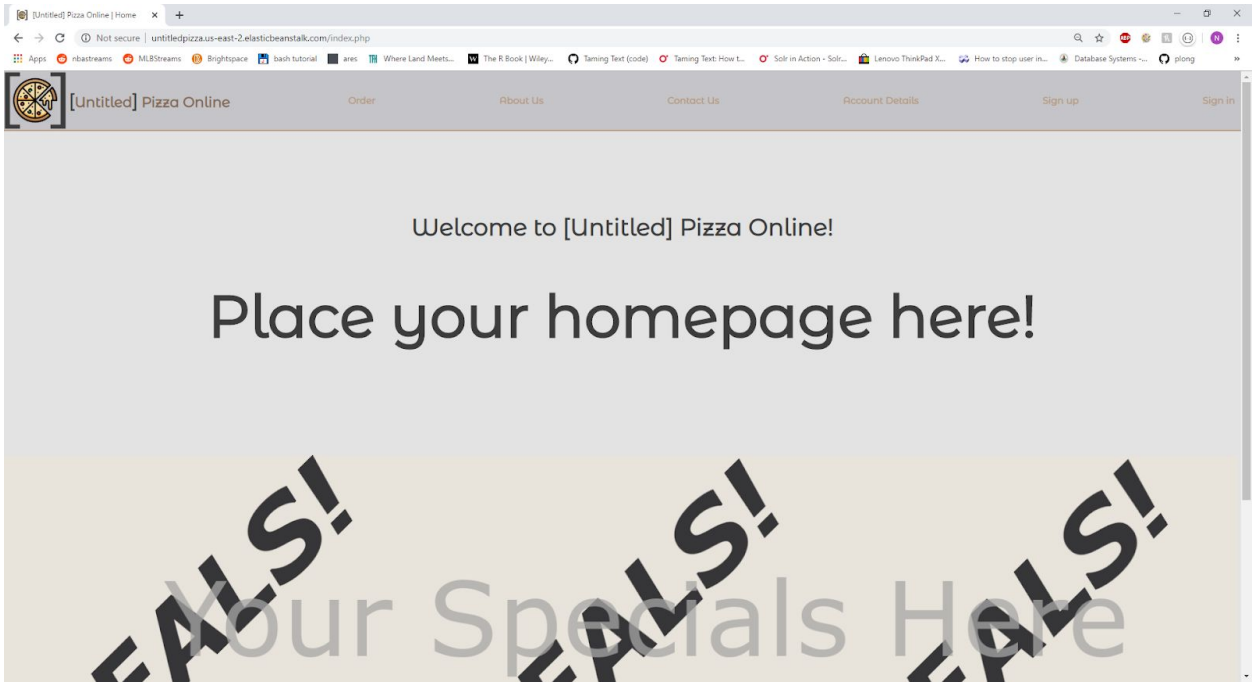
11/20/2019

Internet Systems Programming

Online Pizza Ordering SaaS

About

Untitled Pizza is an attempt at a SaaS pizza ordering application for smaller 'mom and pop' type businesses that do not have as heavy as an internet presence as their big-chain counterparts. The client is able to have a working website with the ability for users/customers to create accounts, log in, view their account details, and view their past orders. There is an about page in the navigation bar that is filled with information about the website. It is very simple but it gets the job done. There is also a contact page, but this is currently under construction and is marked as such on the website.

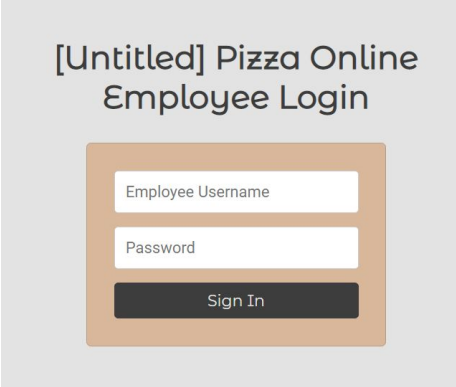


1

Another main feature of the application is the employee login. The employee login has two separate functions. The first and more powerful option is to login as an administrator. After

[1] This is the default homepage of our project website. (http://untitledpizza.us-east-2.elasticbeanstalk.com/index.php)

logging in as an administrator, the site redirects you to a page where you, as a social media representative or owner of a pizza shop, are able to customize the website with one click. You can change several things like the title, logo, name of the website, and the footer code. These elements change across the whole site, by only typing it in once.



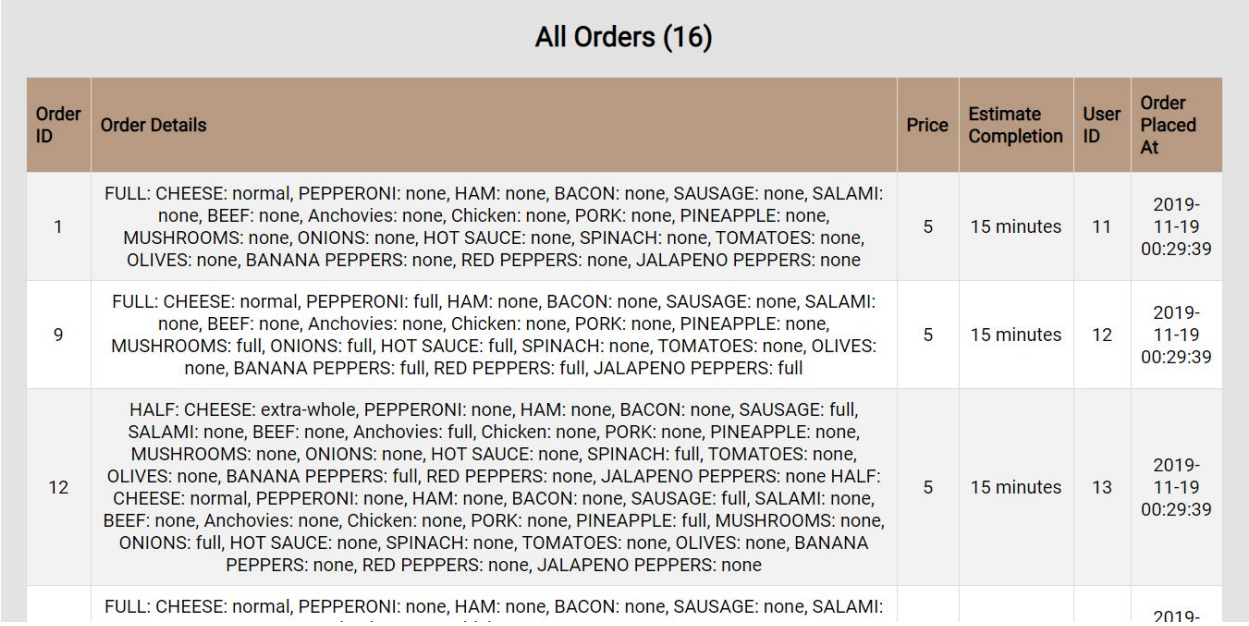
2

3

The second function of the employee login is actual 'employee' login. This allows employees of the businesses to access their pizza order history. After an employee logs in, they are shown a view of all the current orders that have not been served. Our future work in this area will be to

[2] This is the employee login form.
[3] Thi is the employee (admin) feature changing page. This allows you to change everything from the page and have consistent variables across multiple pages.

allow the employee to get rid of an order when it is completed, or mark it as completed in the database, so it's easy for the employee and the user to use the website.



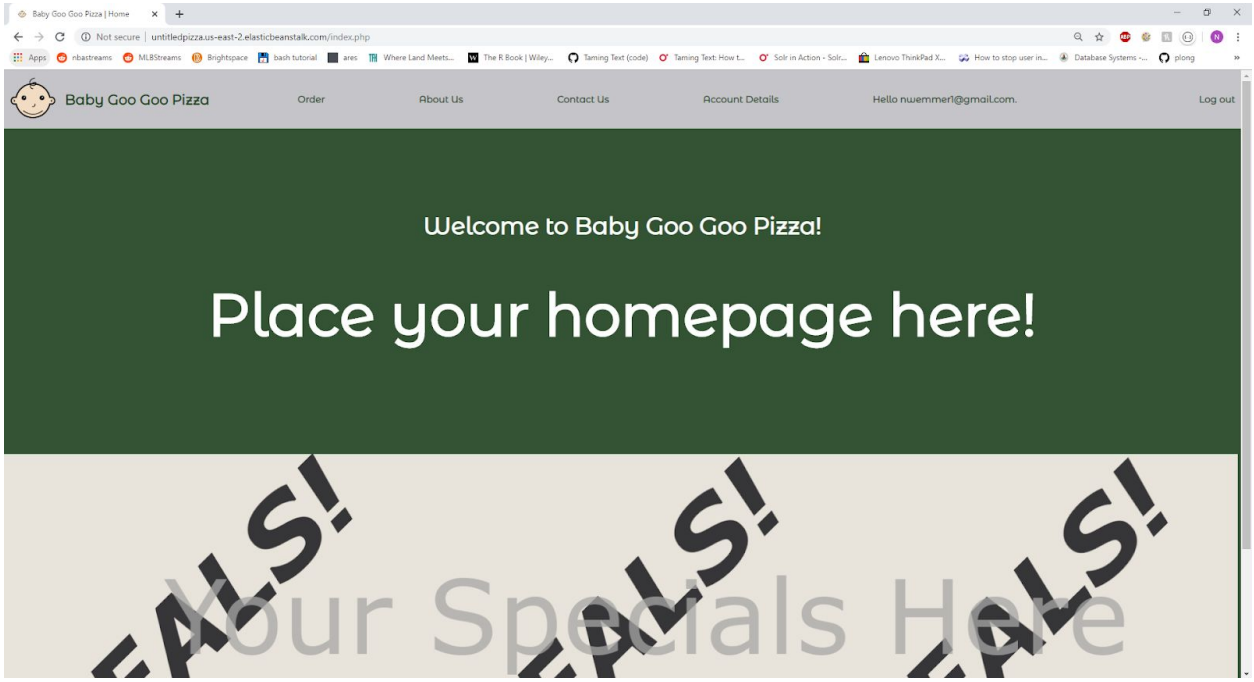
All Orders (16)

Order ID	Order Details	Price	Estimate Completion	User ID	Order Placed At
1	FULL: CHEESE: normal, PEPPERONI: none, HAM: none, BACON: none, SAUSAGE: none, SALAMI: none, BEEF: none, Anchovies: none, Chicken: none, PORK: none, PINEAPPLE: none, MUSHROOMS: none, ONIONS: none, HOT SAUCE: none, SPINACH: none, TOMATOES: none, OLIVES: none, BANANA PEPPERS: none, RED PEPPERS: none, JALAPENO PEPPERS: none	5	15 minutes	11	2019-11-19 00:29:39
9	FULL: CHEESE: normal, PEPPERONI: full, HAM: none, BACON: none, SAUSAGE: none, SALAMI: none, BEEF: none, Anchovies: none, Chicken: none, PORK: none, PINEAPPLE: none, MUSHROOMS: full, ONIONS: full, HOT SAUCE: full, SPINACH: none, TOMATOES: none, OLIVES: none, BANANA PEPPERS: full, RED PEPPERS: full, JALAPENO PEPPERS: full	5	15 minutes	12	2019-11-19 00:29:39
12	HALF: CHEESE: extra-whole, PEPPERONI: none, HAM: none, BACON: none, SAUSAGE: full, SALAMI: none, BEEF: none, Anchovies: full, Chicken: none, PORK: none, PINEAPPLE: none, MUSHROOMS: none, ONIONS: none, HOT SAUCE: none, SPINACH: full, TOMATOES: none, OLIVES: none, BANANA PEPPERS: full, RED PEPPERS: none, JALAPENO PEPPERS: none HALF: CHEESE: normal, PEPPERONI: none, HAM: none, BACON: none, SAUSAGE: full, SALAMI: none, BEEF: none, Anchovies: none, Chicken: none, PORK: none, PINEAPPLE: full, MUSHROOMS: none, ONIONS: full, HOT SAUCE: none, SPINACH: none, TOMATOES: none, OLIVES: none, BANANA PEPPERS: none, RED PEPPERS: none, JALAPENO PEPPERS: none	5	15 minutes	13	2019-11-19 00:29:39
	FULL: CHEESE: normal, PEPPERONI: none, HAM: none, BACON: none, SAUSAGE: none, SALAMI:				2019-

[4]

The appeal to the client is that is a unique alternative to something like GrubHub or UberEats, that allows them to customize their website, and not just having a section of someone else's website such as grubhub, foodtecsolutions, or hungerrush. Sometimes these items aren't listed correctly, or the order is received correctly. These website's URLs are hard to remember, and aren't as easy to just type in and go. We would allow our user to have their own simple URL unique to their name, instead of an extension to a bigger website.

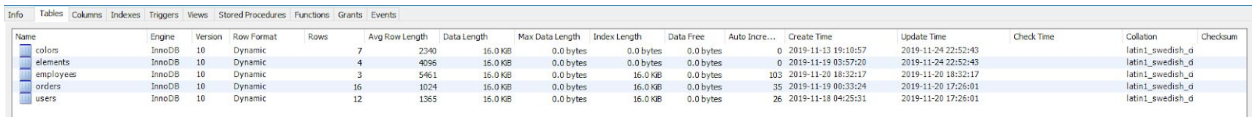
[4] This is the view of the employee, when they are logged in. This is just a view of all of the orders that are currently out.



5

Implementation Details

We used several software platforms to make this project. We started off with Amazon Web Services (AWS) to host the website and the MySQL (RDS) Database. In the database, there are 5 tables that have information about the website and the users. The computing platform we used to run the website code is Elastic Beanstalk. Elastic Beanstalk is nice for us to use because all of the features scale up with demand made by traffic on the website. This is useful for smaller projects because it allows the host computer to use only the necessary resources to host, and not wasting any. We used MySQL and PHP for the backend of the website and HTML/CSS/Javascript for the front end.



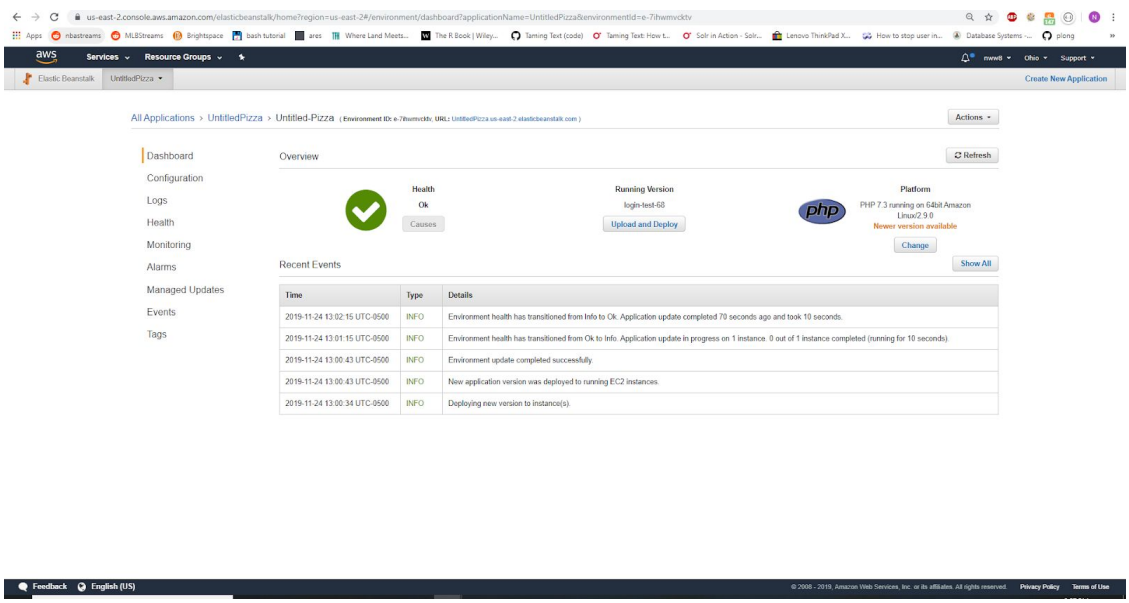
6

[5] This is the website, in the view of a customer, that has changed the website to their own pizza website, 'Baby Goo Goo Pizza'.
[6] This is a view of all the tables in MySQL

Info | Tables | Columns | Indexes | Triggers | Views | Stored Procedures | Functions | Grants | Events

Table	Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments
colors	color_name	varchar(30)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
colors	color_hex	varchar(20)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
elements	name	varchar(30)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
elements	contents	varchar(2500)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
employees	user_id	int(11)		NO			select,insert,update,references	auto_increment	
employees	username	varchar(64)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
employees	password	varchar(32)		YES	latin1	latin1_swedish_ci	select,insert,update,references		
employees	privileges	varchar(1)	0	NO	latin1	latin1_swedish_ci	select,insert,update,references		
orders	order_id	int(11)		NO			select,insert,update,references	auto_increment	
orders	details	varchar(1000)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
orders	price	int(11)		NO			select,insert,update,references		
orders	estimated_completion	varchar(20)		YES	latin1	latin1_swedish_ci	select,insert,update,references		
orders	user_id	int(11)		NO			select,insert,update,references		
orders	order_time	datetime	CURRENT_TIMESTA...	YES			select,insert,update,references	on update CURR...	
users	user_id	int(11)		NO			select,insert,update,references	auto_increment	
users	email	varchar(80)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
users	password	varchar(32)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
users	phone_number	varchar(20)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
users	address	varchar(60)		YES	latin1	latin1_swedish_ci	select,insert,update,references		
users	full_name	varchar(80)		NO	latin1	latin1_swedish_ci	select,insert,update,references		
users	past_orders	varchar(2000)		YES	latin1	latin1_swedish_ci	select,insert,update,references		
users	created_at	datetime	CURRENT_TIMESTA...	YES			select,insert,update,references		

7



8

Possible Future Work / Lessons Learned

Possible future work could include many things. First, the website could be scaled to be tailored to a very large amount of pizza shops, and their many customizations. To add to this, work could be done to the order page to give the current admin more control/capability to add what they want to the order system. One possible future work could be giving an admin the power to create new pages using drag and drop capabilities. This would definitely give the admin more freedom in terms of what they want their website to look like to the user.

---

[7] This is a view of how each table is laid out, with all of the attributes for each table.
[8] This is a view of the AWS Elastic Beanstalk Management Console. From here, you can deploy a new version or configure settings.

Some lessons learned during this project were the usefulness/complexity of Amazon Web Services and how it can be used to host a whole n-tier application. Another lesson learned is the possibilities of dynamic, server-side languages like PHP. We also learned a lot about the different uses of a database and how you can use it to store user created values to be outputted all over the site. This is a lot different from how we were using it before, to store user data. Using it to store elements of the page and color values for the CSS was very clever and allowed extreme flexibility when designing the website and keeping a changeable consistent theme from the beginning sign-up forms, to the order page. Allowing each page to be a PHP file but still allowing HTML and javascript in these files was a huge help.

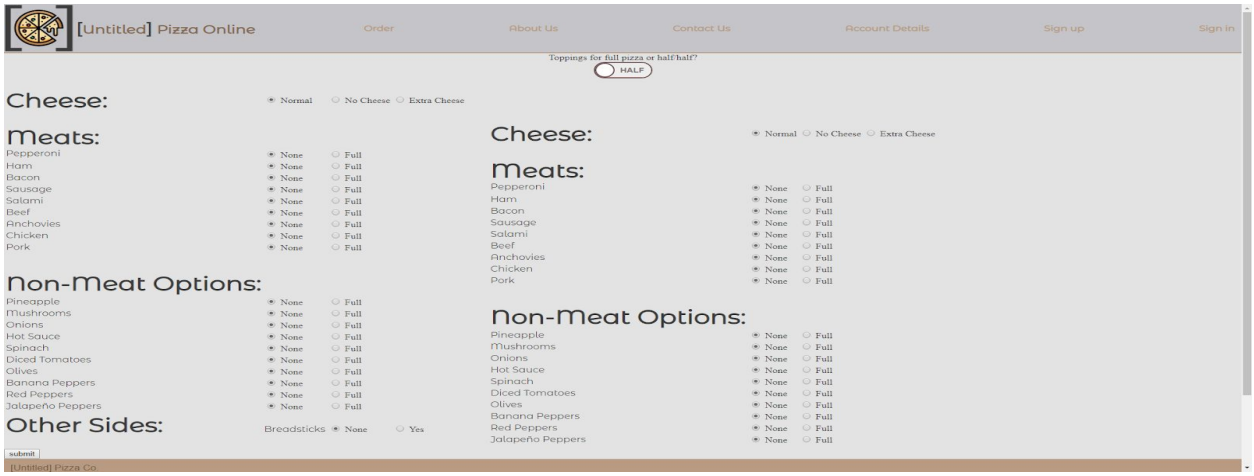
Contributions

Joseph contributed by creating the PHP/HTML/CSS/javascript for the order page where a user is able to enter their order. The format starts with a switch that determines if the following pizza's toppings will be for all or half of the pizza. If full, the user inputs the toppings and presses submit. If half, the table with the toppings split and the user can enter the toppings he/she wants on both halves. Once submitted, a PHP string is created containing all of the order details. If "none" is selected for a topping, that is not appended to the string. Once the PHP variable is created, it is put into the database, in the orders table under the order details column. Also, that user's id is put into the other table so record that he/she ordered something from the order page.

Nathan contributed by creating and maintaining the MySQL database, hosted by Amazon Web Services (AWS). From here, he created the tables and their attributes, established connection code for each of the pages across the site, and displayed and used variables or data from the database for the user to see information. Also, Nathan learned how to use Amazon Web Services to fully deploy the n-tier web application. Here is where Elastic Beanstalk was used, along with the RDS (MySQL) database. A lot of the features that had to do with logging in, or inserting, deleting, or updating data in the database was also done by Nathan. PHP code was written, alongside CSS/Javascript/HTML to create different pages for the website. MySQL scripts were written to put employee data in, because currently, you do not want just anybody

to input a new employee or administrator user into the database. So you have to manually connect to the database through the Workbench or some similar program and add data to allow you to login. This is fairly simple and only takes seconds to complete.

Jacob contributed by creating the administrator page that uses database tables to store the colors of the site and HTML elements and update them. Since the administrator page, controls the color scheme and HTML elements all pages had to be made so they would fetch the correct things from the database, then be put into the HTML code to be seen by the user. This was made easier by creating php code snippets for things like the header and footer. Jacob also contributed by formatting much of the look of the website. This includes the format of the header, footer, sign in forms, and more. For features that were competed, but had no consistent CSS, Jacob would take the code that was there and add CSS code and some HTML, mostly divs, to make the page pleasing to the eyes without breaking functionality.



9

[9] This is a view of the order page when the user selects that they want different toppings on the two halves of the pizza.