

Team 6



In-N-Out Chatbot Application

Product Manager- Sharv Utpat

Architect- Hari Prasath Ramalingam

Developer – Alex Hernandez

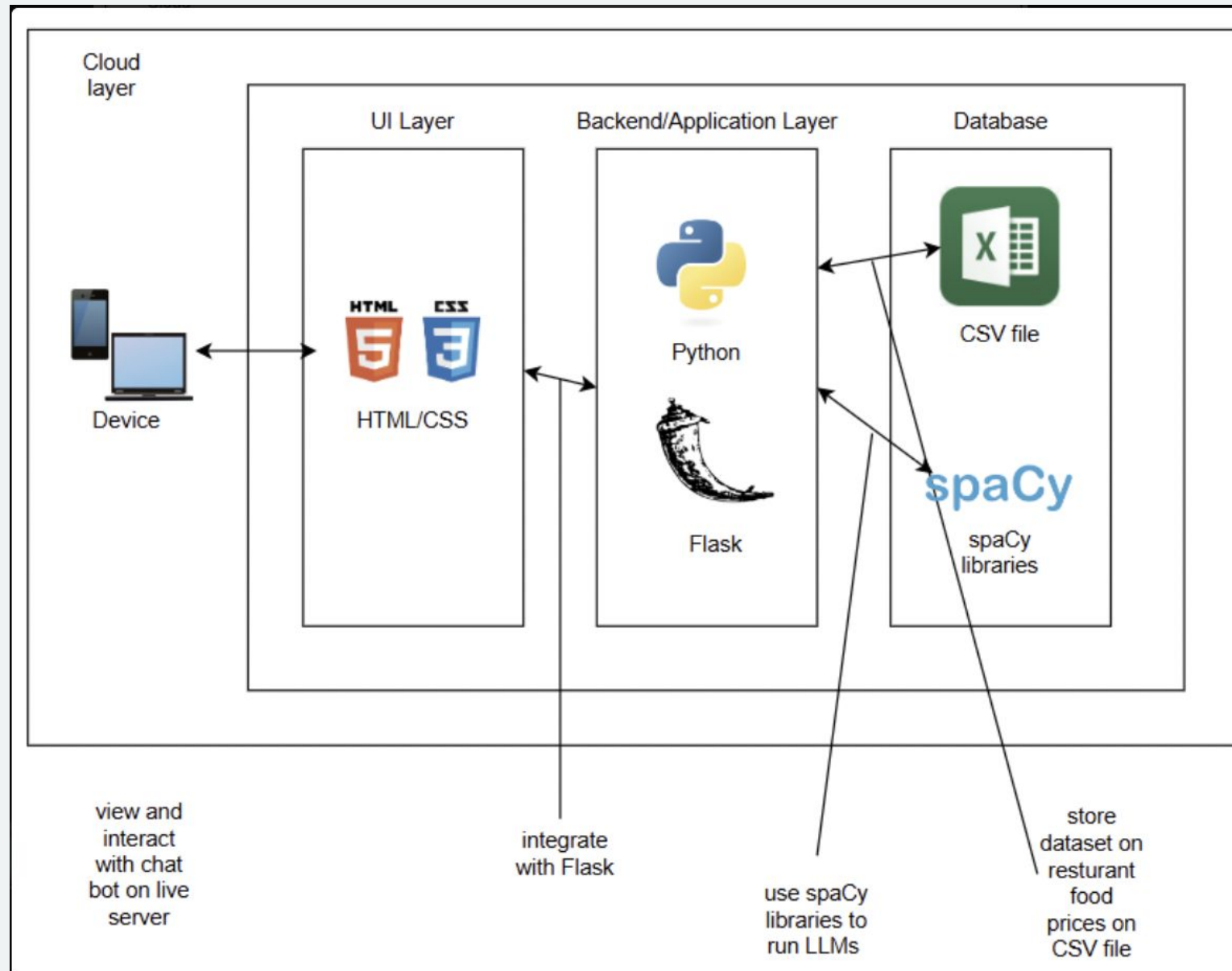
QA/DevOps – Suraj Namburi

<https://innoutchatbox.onrender.com/>

Our Project

- Building a Fast Food Service AI Chat Bot
- Our restaurant was In-N-Out
- We needed an interactive user interface where a customer is able to look at a menu and order the food items that they request
- We built a website to do this effectively
- Our approach started off with simplicity
 - *have a menu and customer must directly say what menu items they want*
 - *menu is to always be visible from the beginning*
 - *list of ordered items must always be visible while ordering*
- We would show the accumulated price accordingly and charge the user at the end
- Later, we added features to try to add or remove ingredients from the menu items

Architecture Diagram



Confluence/JIRA Documentation

2. Menu Browsing

- Display full menu (burgers, fries, drinks, special items).
- Filter and search options (e.g., by category or dietary preferences).

3. Order Customization

- Customization options for each menu item (e.g., burger styles, toppings).
- Option to specify special instructions (e.g., "light on sauce").

4. Order Placement

- Add items to the cart.
- Review and edit cart before finalizing the order.
- Confirmation of the order summary before checkout.

5. Location Services

- Geolocation to find nearby In-N-Out locations.
- Display operating hours and contact information for each location.

6. Order Tracking

- Real-time updates on order status (e.g., preparing, ready for pickup).
- Estimated wait times based on location and order size.

7. Chatbot Interaction

- Natural language processing for understanding user queries.
- Predefined flows for common tasks (placing an order, FAQs).
- Option to escalate to human support if needed.

Non-Functional Requirements

1. Usability

- Intuitive user interface and user experience design.
- Accessibility features for users with disabilities.

2. Performance

- Quick load times for menu and order processing.
- Scalable infrastructure to handle peak times.

3. Security

- Secure user data storage and transaction processing.
- Compliance with relevant data protection regulations (e.g., GDPR, CCPA).

4. Reliability

- Robust error handling and recovery options.
- Regular updates and maintenance for the app.

5. Compatibility

- Support for multiple devices (iOS, Android, tablets).
- Compatibility with various screen sizes and resolutions.

6. Analytics

- Track user behavior and order trends.
- Monitor chatbot interactions for continuous improvement.

Confluence/JIRA Documentation

Projects / AI-ChatBot

List

 Give feedback

Search list



 Share

 Filter

 Group

 Format

 More

<input type="checkbox"/>	Type	# Key	Summary	Status	Assignee	+
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-6	Github Repository	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-4	Create ChatBot Script	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-1	CSV File for Menu	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-5	Created Confluence/Jira Pages	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-10	Optimize Chatbox Script	IN PROGRESS		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-7	Optimize user ordering option flexibility	IN PROGRESS		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-11	Architecture Diagram	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-12	Requirements/Architecture Upload to Confluence	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-13	Complete Chatbot(Alpha)	DONE		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	AC-15	Basic In-N-Out Menu UI	DONE		

+ Create



Quickstart



Cloud Deployment

The screenshot shows the Render dashboard interface. At the top, there's a navigation bar with 'Render', 'Dashboard', 'Blueprints', and 'Env Groups'. A '+ New' button and a user profile 'Alexander Hernandez's Workspace' are on the right. Below the navigation bar, the main content area shows the 'innoutchatbox' web service. It includes tabs for 'Python 3' and 'Free', an 'Upgrade your instance' link, and buttons for 'Connect' and 'Manual Deploy'. A sidebar on the left lists various settings like 'Events', 'Logs', 'Disks', 'Environment', 'Shell', 'Previews', 'Jobs', 'Metrics', 'Scaling', and 'Settings'. The 'Settings' tab is active, showing a 'General' section with fields for 'Name' (innoutchatbox) and 'Region' (Oregon (US West)). At the bottom of the settings, it shows 'Free' tier with '0.1 CPU' and '512 MB'.

- Explored several hosting options, including Heroku.
- Selected Render despite limitations in the free plan, such as service spin-down during inactivity.
- The free plan provided sufficient memory and CPU resources for our lightweight application.
- Configured the build command to install necessary dependencies from requirements.txt.
- Deployed the Flask application using Gunicorn, ensuring seamless integration with the web server.

This screenshot shows the 'Build Command' configuration panel in the Render dashboard. It has three sections: 'Build Command', 'Pre-Deploy Command', and 'Start Command'. Each section has a text input field and an 'Edit' button. The 'Build Command' field contains '\$ pip install -r requirements.txt'. The 'Pre-Deploy Command' field is empty and marked as 'Optional'. The 'Start Command' field contains '\$ gunicorn app:app'.

Command Type	Command
Build Command	\$ pip install -r requirements.txt
Pre-Deploy Command (Optional)	\$
Start Command	\$ gunicorn app:app

AI & NLP Packages Utilized

spaCy

- **Description:**
Utilized as the primary NLP library for processing and understanding user inputs.
- **Features Leveraged:**
 - **Lemmatization:** Simplifies words to their base forms for consistent matching.
 - **Noun Chunking:** Identifies meaningful phrases within user queries.
 - **PhraseMatcher:** Enables precise matching of menu items within user inputs.

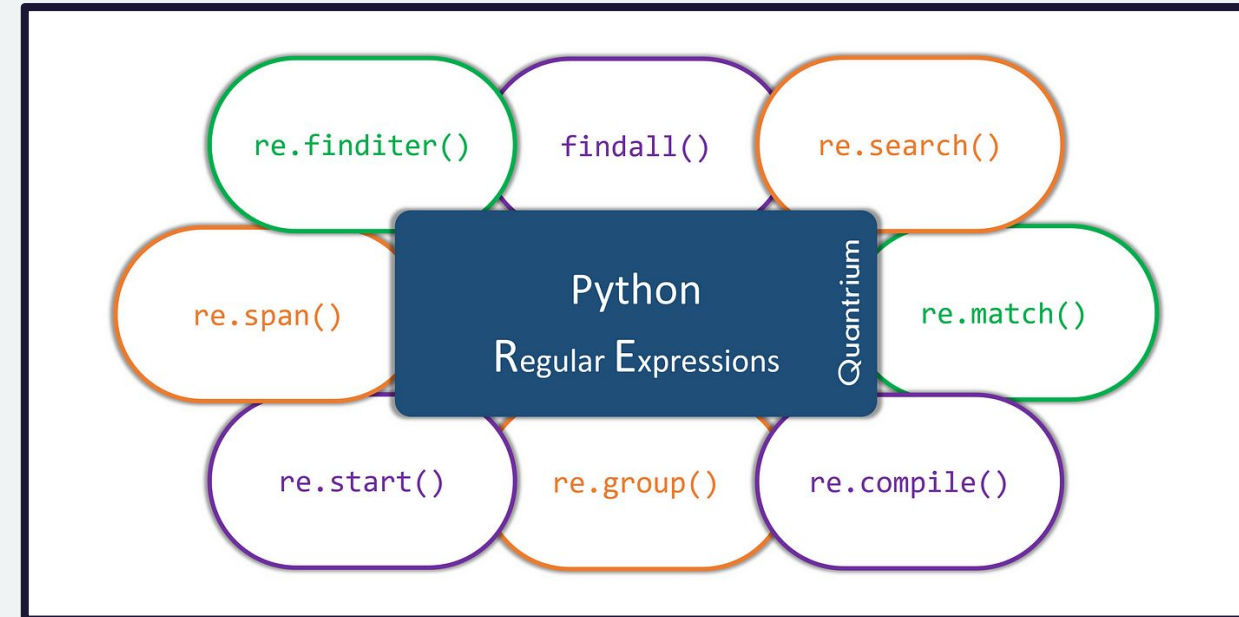
The logo for spaCy, featuring the word "spaCy" in a blue, sans-serif font. The "C" is capitalized and larger than the other letters. The logo is enclosed in a thin black rectangular border.

Pattern Matching with spaCy's PhraseMatcher

- **Description:**
Employed to accurately detect and prioritize menu items in user messages.
- **Benefits:**
 - **Accuracy:** Reduces false positives by matching exact phrases.
 - **Efficiency:** Enhances the chatbot's ability to handle complex orders involving multiple items.

Regular Expressions (re)

- **Description:**
Used for preprocessing tasks such as replacing numerals with words.
- **Applications:**
 - **Numeral Replacement:** Converts digits (e.g., "2") to words (e.g., "two") for uniformity in processing.
 - **Text Cleaning:** Removes unwanted characters and standardizes input format.



Supporting Libraries & Data Management

Flask

- **Description:**
Serves as the web framework for building the chatbot interface.
- **Key Uses:**
 - **Routing:** Manages URL endpoints for user interactions.
 - **Session Management:** Maintains user-specific order data throughout the chat session.

pandas

- **Description:**
Utilized for data manipulation and handling the menu data.
- **Key Uses:**
 - **CSV Handling:** Reads and processes the `In N Out Menu.csv` file.
 - **Data Normalization:** Cleans and standardizes menu item names for accurate matching.

Regular Expressions (re)

- **Description:**
Supports text preprocessing and pattern matching tasks.
- **Key Uses:**
 - **Numeral Conversion:** Transforms digits to words.
 - **Input Sanitization:** Cleans user input by removing punctuation and unwanted characters.

Session-Based Storage

- **Description:**
Manages user orders without the need for an external database.
- **Advantages:**
 - **Simplicity:** Eliminates the complexity of integrating a separate database.
 - **Efficiency:** Stores order data temporarily, ensuring privacy and quick access.



Key Modules & Custom Functionalities

Order Parsing Module

- **Function:** `parse_order(user_input)`
- **Description:**
Extracts ordered items and their quantities from user messages.
- **Key Features:**
 - **Lemmatization:** Converts words to their base forms for consistent matching.
 - **Prioritized Matching:** Sorts menu items by length to prioritize longer, more specific names, preventing partial matches.

Removal Parsing Module

- **Function:** `parse_removal(user_input)`
- **Description:**
Identifies items and quantities the user wishes to remove from their order.
- **Key Features:**
 - **Quantity Extraction:** Determines the number of items to remove.
 - **Item Validation:** Ensures that only existing order items can be removed.

Modification Parsing Module

- **Function:** `parse_modifications(user_input, parsed_order)`
- **Description:**
Handles ingredient-level modifications such as adding or removing specific components (e.g., "no onions", "extra cheese").
- **Key Features:**
 - **Action Identification:** Detects modification intents like 'add', 'remove', 'without'.
 - **Ingredient Association:** Associates modifications with the correct ordered item.

Multiple Actions Parsing Module

- **Function:** `parse_multiple_actions(user_input)`
- **Description:**
Parses user inputs that include both additions and removals in a single message.
- **Key Features:**
 - **Simultaneous Handling:** Processes adding and removing actions without conflict.
 - **Sequential Processing:** Adds items before removing to maintain order integrity.

Session Management & Order Handling

- **Functions:** `handle_addition()`, `handle_removal()`, `handle_modifications()`, `handle_cancel_order()`
- **Description:**
Manages the state of user orders, applying additions, removals, and modifications as per user requests.
- **Key Features:**
 - **Dynamic Updates:** Reflects real-time changes to the order based on user interactions.
 - **Error Handling:** Provides feedback when invalid actions are attempted (e.g., removing an item not in the order).

Chat Interface & Response Generation

- **Functionality:**
Renders the chat interface, handles user inputs, and generates appropriate bot responses based on parsed data.
- **Key Features:**
 - **User-Friendly UI:** Clean and intuitive interface for seamless interactions.
 - **Automated Responses:** Delivers real-time updates and confirmations to enhance user experience.

Application User Interface Successes

👋 Welcome to In-N-Out Ordering Chatbot! How can I assist you today?

Please display the menu

🍔 In-N-Out Menu:

Cheeseburger: \$3.70
Hamburger: \$3.30
French Fries: \$2.15
Shake: \$2.90
Small Drink: \$1.95
Medium Drink: \$2.10
Large Drink: \$2.30
X Large Drink: \$2.50
Milk: \$0.99
Hot Cocoa: \$2.10
Coffee: \$1.35
Number One Meal: \$9.55
Number Two Meal: \$7.95
Number Three Meal: \$7.55

👋 Welcome to In-N-Out Ordering Chatbot! How can I assist you today?

I would like french fries and a cheeseburger and a medium drink

🛒 **Item(s) added to your order.**

Your current order:

1 x French Fries - \$2.15
1 x Cheeseburger - \$3.70
1 x Medium Drink - \$2.10

Total: \$7.95

🛒 **Item(s) added to your order.**

Your current order:

1 x French Fries - \$2.15
1 x Cheeseburger - \$3.70
1 x Medium Drink - \$2.10

Total: \$7.95

Remove cheeseburger

❌ Removed 1 x Cheeseburger from your order.

Your current order:

1 x French Fries - \$2.15
1 x Medium Drink - \$2.10

Total: \$4.25

✖ Removed 1 x Cheeseburger from your order.

Your current order:

1 x French Fries - \$2.15
1 x Medium Drink - \$2.10

Total: \$4.25

+ Added Onion to your Cheeseburger.

Your current order:

1 x French Fries - \$2.15
1 x Medium Drink - \$2.10
1 x Cheeseburger (Add: Onion) - \$3.70

Total: \$7.95

Add a cheeseburger with extra onions

1 x French Fries - \$2.15
1 x Medium Drink - \$2.10

Total: \$4.25

+ Added Onion to your Cheeseburger.

Your current order:

1 x French Fries - \$2.15
1 x Medium Drink - \$2.10
1 x Cheeseburger (Add: Onion) - \$3.70

Total: \$7.95

What does french fries contain?

📖 The French Fries contains the following ingredients: Potato, Salt.

What does cheeseburger contain?

📖 The Cheeseburger contains the following ingredients: Beef Patty, Cheese, Lettuce, Onion, Tomato, Spread, Pickle.

Add a cheeseburger with extra onions

Cancel my order

🗑 Your entire order has been canceled.

Potential Drawbacks

I would like a hamburger without onions and a shake

 Shake doesn't contain Onion.

Your current order:

1 x Hamburger - \$3.30
1 x Shake - \$2.90

Total: \$6.20



Customized
Items Must Be
at End of List
in Order

I would like a shake and hamburger without onions


— Removed Onion from your Hamburger.

Your current order:

1 x Shake - \$2.90
1 x Hamburger (Remove: Onion) - \$3.30

Total: \$6.20



 **Item(s) added to your order.**

Your current order:

1 x Cheeseburger - \$3.70
1 x French Fries - \$2.15

Total: \$5.85

Add a cheeseburger and french fries please

Cannot add or remove items in the same line as that causes confusion

Remove the cheeseburger and add a hamburger

 You don't have any Hamburger in your order to remove.

Your current order:

1 x Cheeseburger - \$3.70
1 x French Fries - \$2.15

Total: \$5.85

Add a cheeseburger and french fries please

 **Item(s) added to your order.**

Your current order:

1 x Cheeseburger - \$3.70
1 x French Fries - \$2.15

Total: \$5.85

Add a hamburger and remove the cheeseburger

 You don't have any Hamburger in your order to remove.

Your current order:

1 x Cheeseburger - \$3.70
1 x French Fries - \$2.15

Total: \$5.85

