Difference between C47 1:1 & C47 1:2	1. Included class dynamics. 2. Added emojis. 3. The activities have been restructured for app development. 4. Removed knowledge base article. 5. Added a detailed explanation to use entities. 6. Added zip file to import the previous chatbot and added steps to import.
Topic	**MULTILINGUAL CHATBOT**
Class Description	**Students create a multilingual chatbot and integrate it with their app.**
Class	**C47**
Class time	**50 mins**
Goal	• Create the chatbot as a multilingual agent. • Create the knowledge base using a CSV file. • Integrate the chatbot app using Thunkable.

Resources Required	• Teacher Resources:
	  ○ Dialogflow
	  ○ laptop with internet connectivity
	  ○ earphones with mic
	  ○ notebook and pen
	
	• Student Resources:
	  ○ Dialogflow
	  ○ laptop with internet connectivity
	  ○ earphones with mic
	  ○ notebook and pen

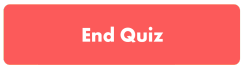
Student Motivation and Engagement	• **Hats-off**: Specific instructions for giving hats-off will be provided in the lesson.

	•  **Concept Magnifier**: Used to highlight new concepts and connect them with real-life examples. •  **Knock-Knock!**: To nudge the students to make sure they are attentive. •  **Thinking Caps**: Used to engage the students for an activity or Q&A. •  **All types of Quizzes**: Includes revision quizzes, riddles, and pop-up quizzes. •  **Candy Boosters**: Used to motivate the students to do better in the activities. •  **Important Points to Remember**: To highlight important concepts.	
Class structure	**Warm-Up** **Teacher-led Activity** **Student-led Activity** **Wrap-Up**	**5 mins** **15 min** **25 min** **5 mins**

CONTEXT

- **Introduce the concept of changing lanes by adding the key condition.**

Class Steps	Teacher Action	Student Action
Step 1: Warm-Up (5 mins)	Hello! Welcome back to your action-packed coding class! To start today's class - let's first quickly review what we did in the last class.	*FYR: If out of the two students, one is an introvert/average learner, and the other is an extrovert/fast learner, then we refer to the introvert/average learner as <Student name 1> and the*

		extrovert/fast learner as <Student name 2>. *The students review the code from the last class.*
	I have an exciting quiz question for you! Are you both ready to answer this question? *Click on the*  *button on the bottom right corner of your screen to start the In-class Quiz.* *A quiz will be visible to both you and the students.* *Encourage the students to answer the quiz question.* *The students may choose the wrong option, help the students to think correctly about the question and then answer again.* *After the students select the correct option, the*  *button will start appearing on your screen.* *Click on End Quiz to close the quiz pop-up and continue the class.*	**ESRs:** Varied.

	 *Refer to Teacher Resources page/slide 1-2 for visual aid.* Q1) Which intents are sent to a presented intent and thus form a flow of conversation? Q2) Which is a simple convenience field that assists in executing logic in your service?  *Give a candy booster to the student who has answered correctly.* Great! Today's class is quite an interesting one, we are going to learn how to publish our chatbot as an app.  Are you both with me so far? Give me a thumbs up, so I know you all are paying attention. *Nudge the students if they do not give a thumbs up.* Great! So let's get started! I will now share my screen.	**ESRs:** The Follow-up intent. **ESRs:** The action field. **ESRs:** Varied.

Teacher Initiates Screen Share		
CHALLENGE • **Understanding the concept of entities.**		
Step 2: Teacher-led Activity (15 mins)	*The teacher opens Teacher Activity 2, and guides the students to open Student Activity 1.* *Note: Guide the students to open the previously created health bot.*	*Students listen*
	In the previous class, we learned to create an intent parameter and an action. *<Student name 2>*, can you tell me what is an entity? Each intent parameter has a type, called the entity type which dictates exactly how data from an end-user expression is extracted. There are two types of entities: • System entity. • Custom entity. *<Student name 1>*, can you name any system entities which we have created? Exactly! Just like system entities, we can also create your own custom entities for custom data.	**ESRs:** Varied. **ESRs:** Date, time.

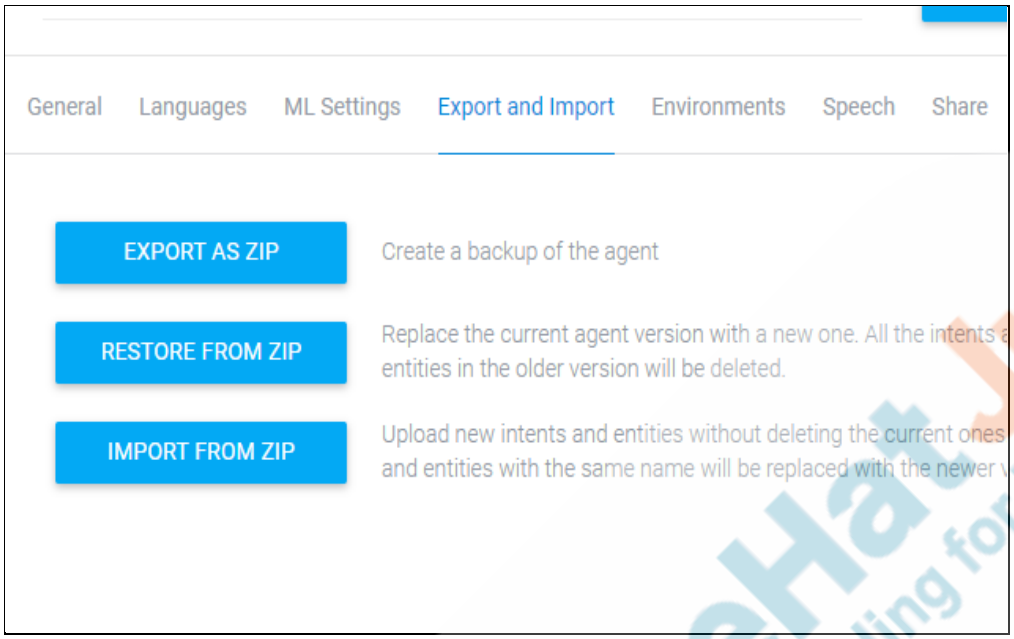
	Exactly! Similarly, we can define a **vegetable** entity that can match the types of vegetables available for purchase with a grocery store agent.	
	We can create different kinds of entities, depending on the entities selected: • Map entity • List entity • Composite entity (*a special kind of list entity*) • Regexp entity Lets learn about List entity.Many system entities are list entities. For example, the @sys.color system entity matches common color inputs like "red" or "blue". In today's class we will create an entity for our health bot app for being rated by the user. Here, the end-user will be asked to choose a rating from 1-5. The teacher can open Teacher Activity 3 and import and save the previous app by clicking on download from GitHub	

*Note: We will be adding ratings to the previous app. The teacher can import Teacher Activity 3 before the class. The name of the app is changed from **HealthBot** to **junior**.*

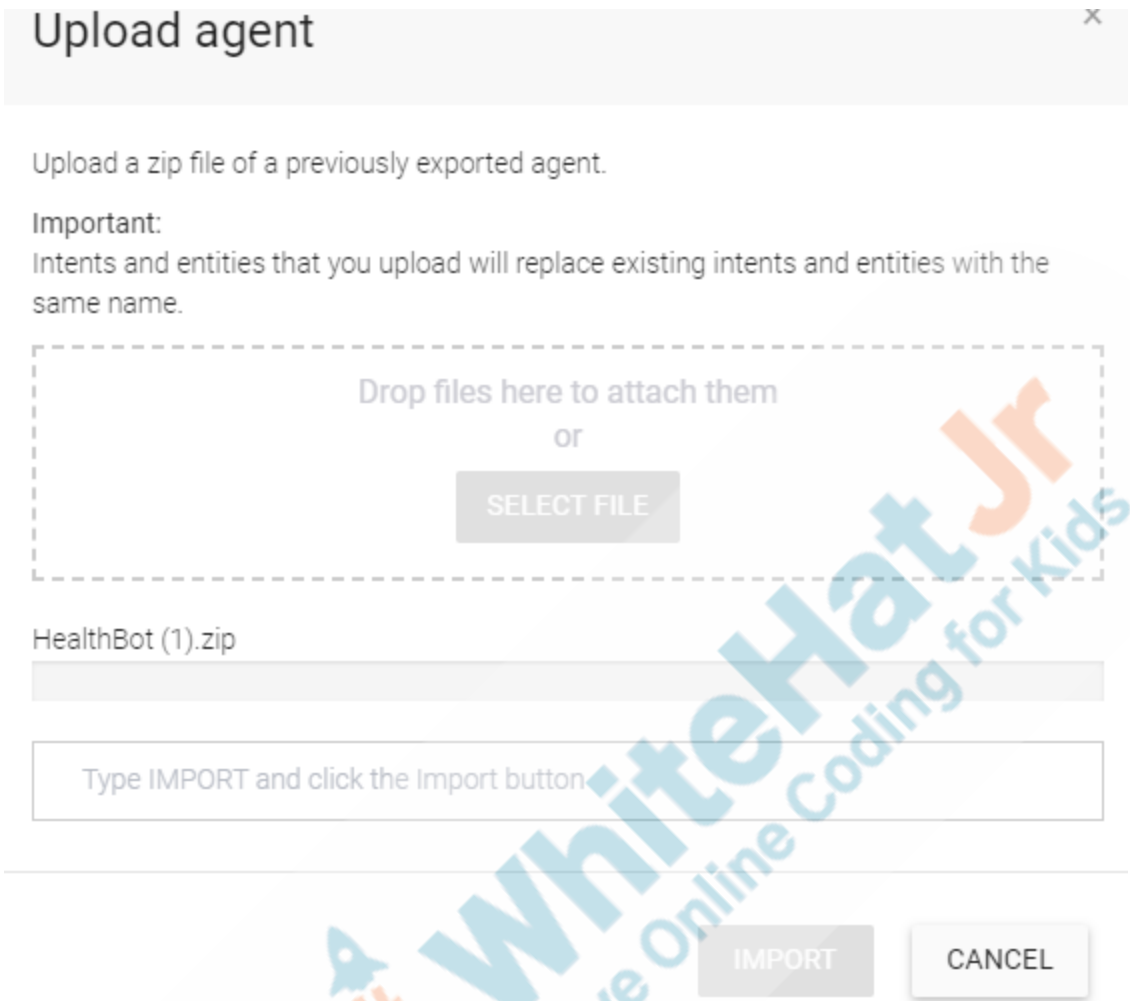
- The teacher needs to download the zip file from Teacher Activity 3.

- Create a new agent as **junior:**
- Click on the settings icon to select the tab, **Export and Import**. Thereafter, choose

the **IMPORT FROM ZIP** option:



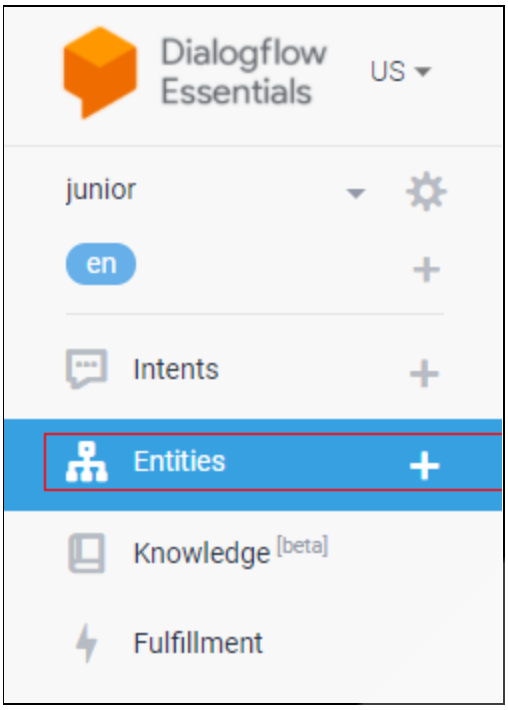
- Add the zip file from folder **HealthBot**:



- Type **IMPORT** within the tab provided and press the **IMPORT** button. Click **Done**:

Note:Teacher can IMPORT and keep the app ready.Now lets learn how to create the Entities

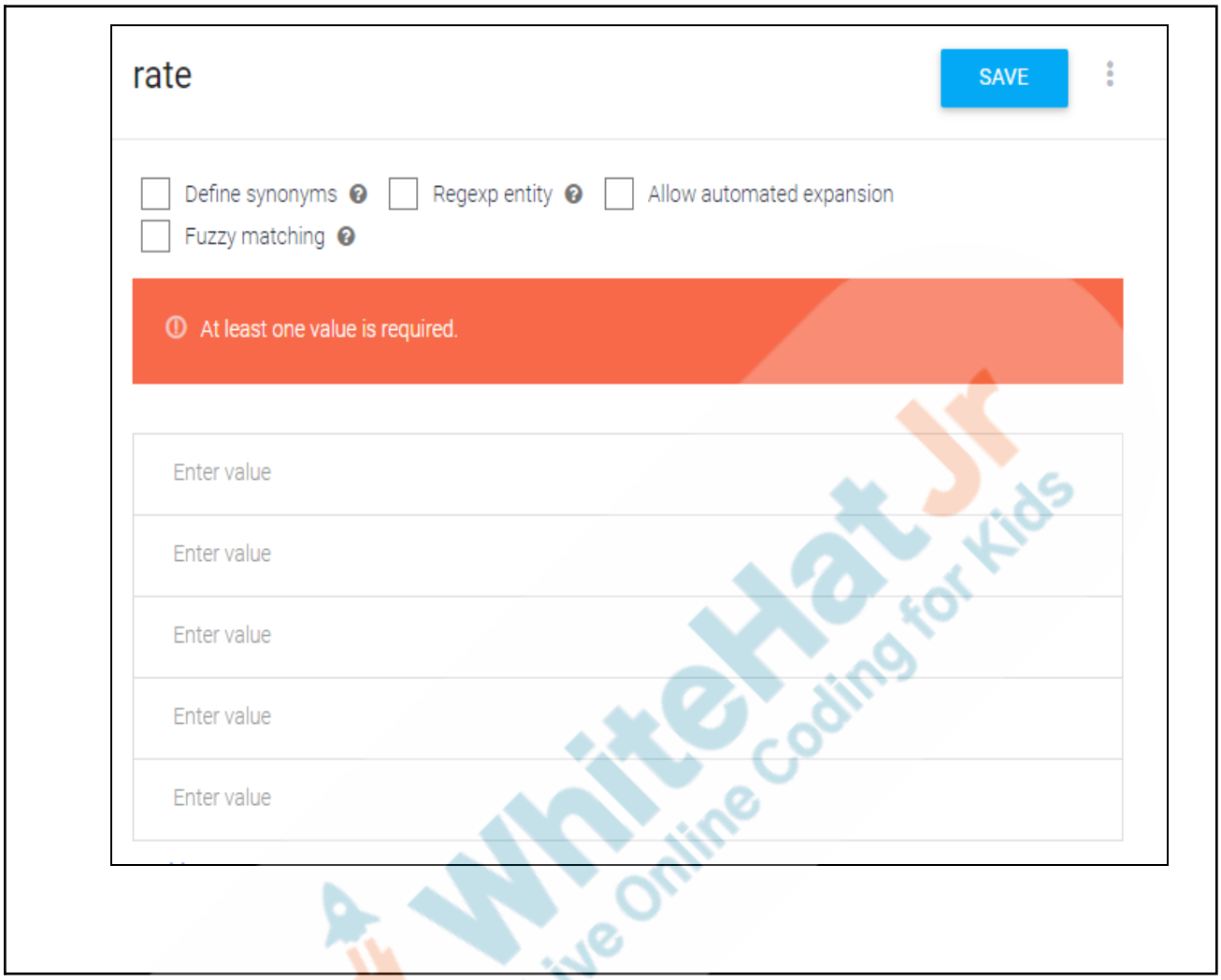
1. Select **Entities** from the left-hand side tab.

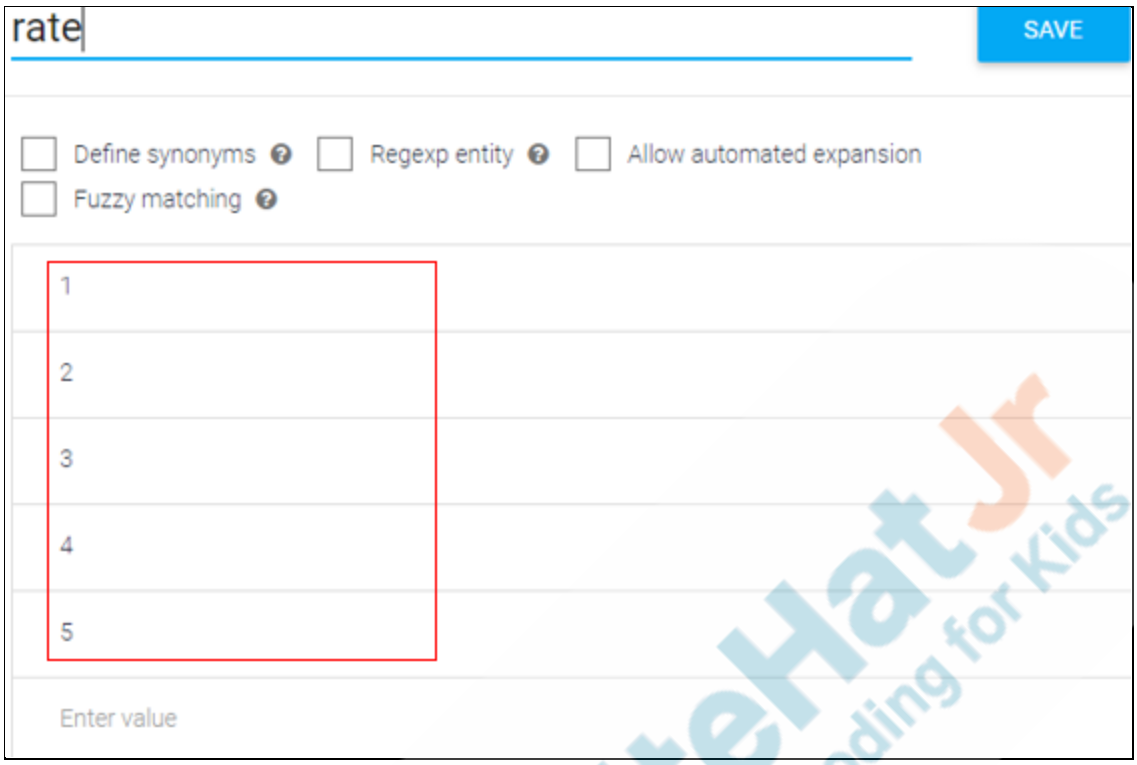


2. Create a new entity by clicking on **CREATE ENTITY**:

*Note: Disable the **Define synonyms.** (This option is enabled by default, hence, we need to disable it.)*

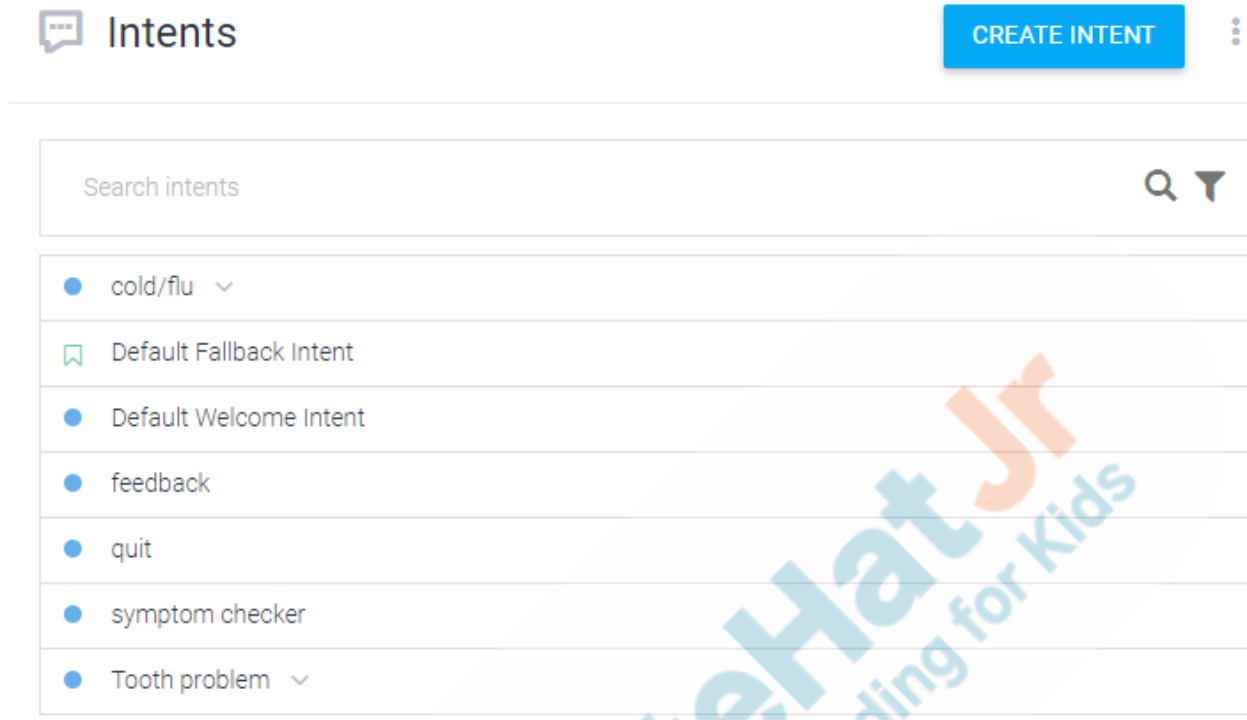
3. Add the name of the entity as **rate** and add the values from 1 to 5 inside the **Enter value** input box and click on **SAVE**:





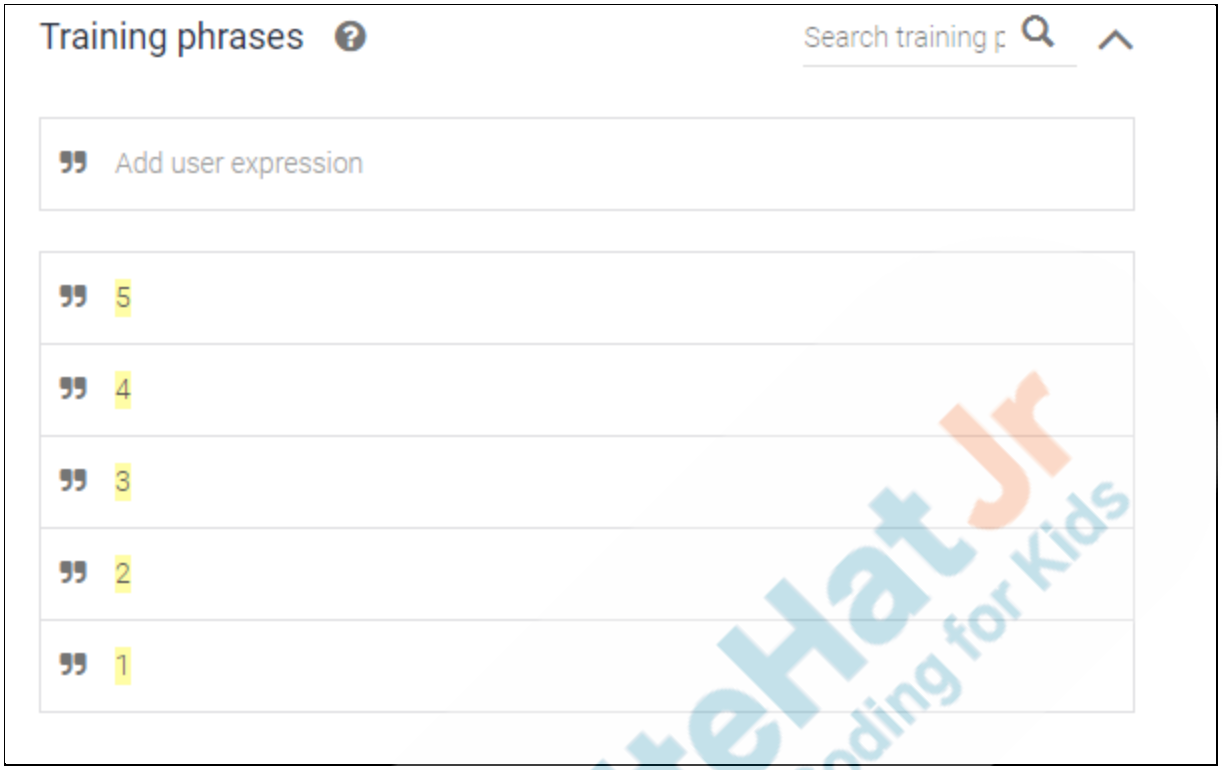
4. Now create a new intent with the name as **feedback**.

   Once you have created this intent, it should be visible in the **Intents** option as shown below:

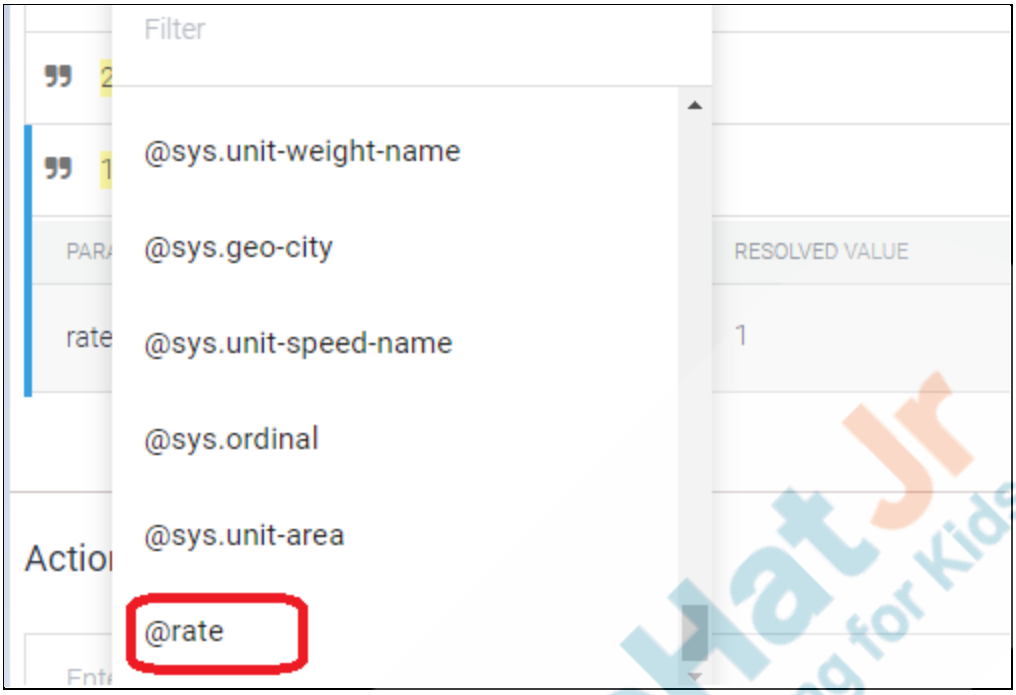


/

5. Now, add the training phrases for the **feedback** intent as shown below for the numbers 1-5 and click on **SAVE**.

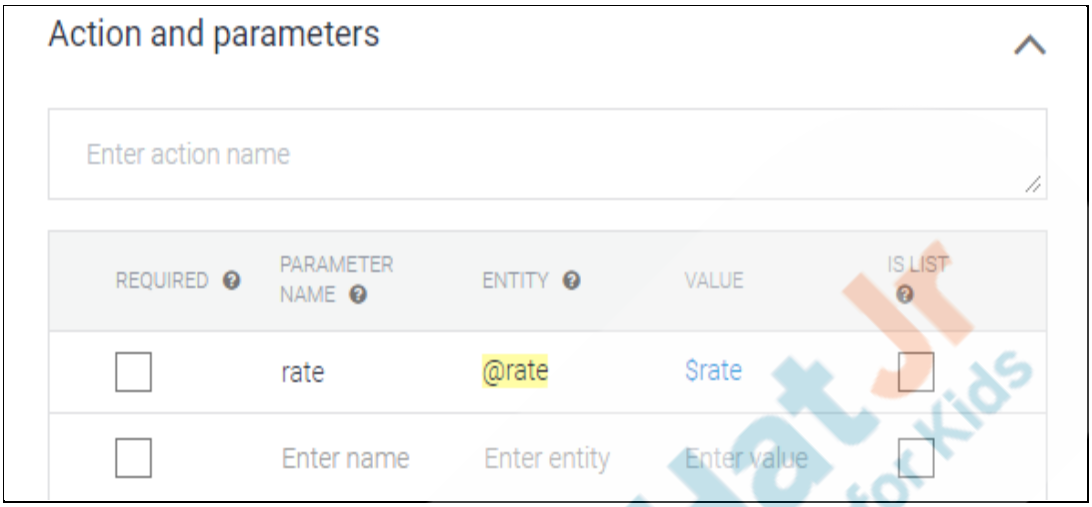


6. Double-click on numbers to add **Action and parameters**, choose the **@rate** entity for all the training phrase number(1-5)

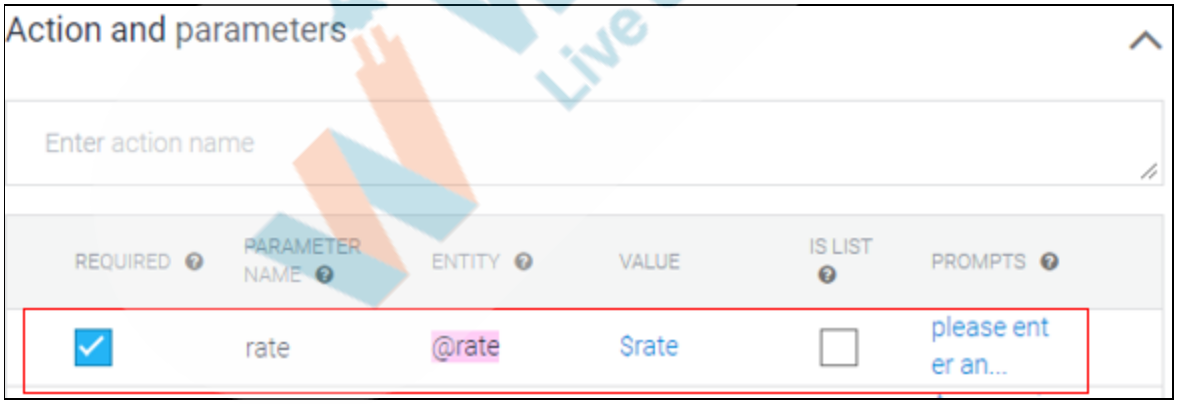


7. Check the below screenshot of @rate added to the training phases

8. Check the **Action and parameters** tab. It will contain **PARAMETER NAME** as **rate**, **ENTITY** as **@rate**, **VALUE** as **$rate**. Keep the **IS LIST** checkbox **unticked**.

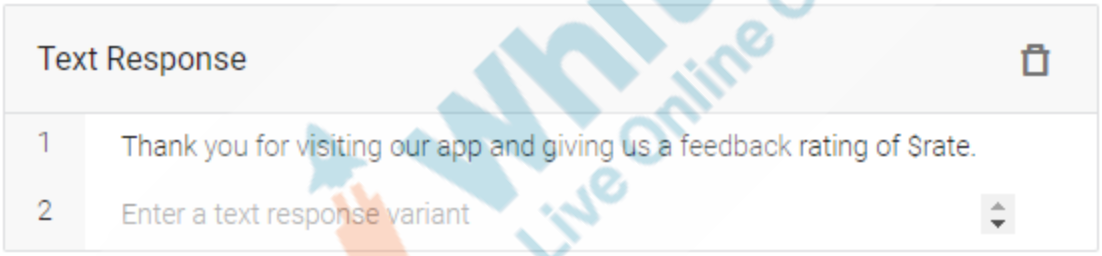


9. Enable the **REQUIRED** tab to get the **PROMPTS** tab:

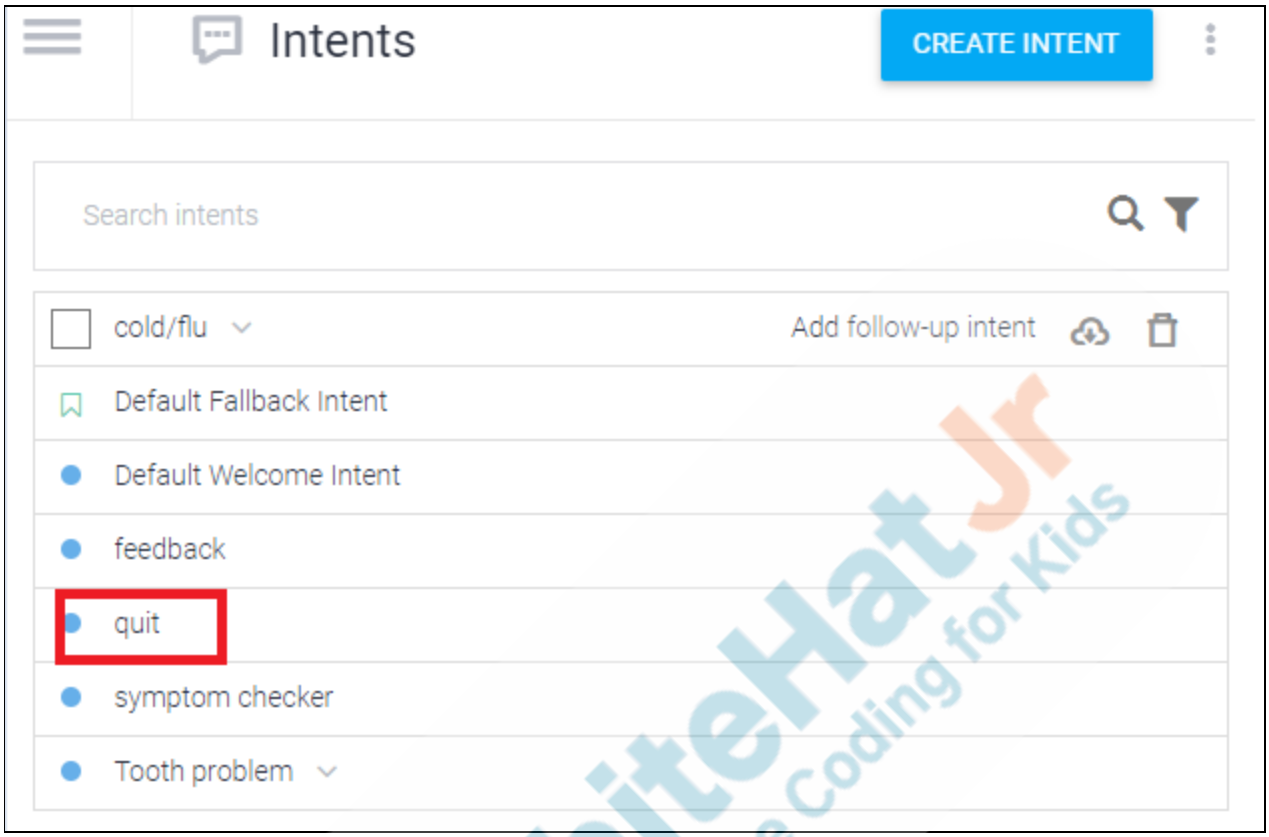


10. Now, add the **PROMPTS** message for the **rate** action and enter, **Please enter day and time:**

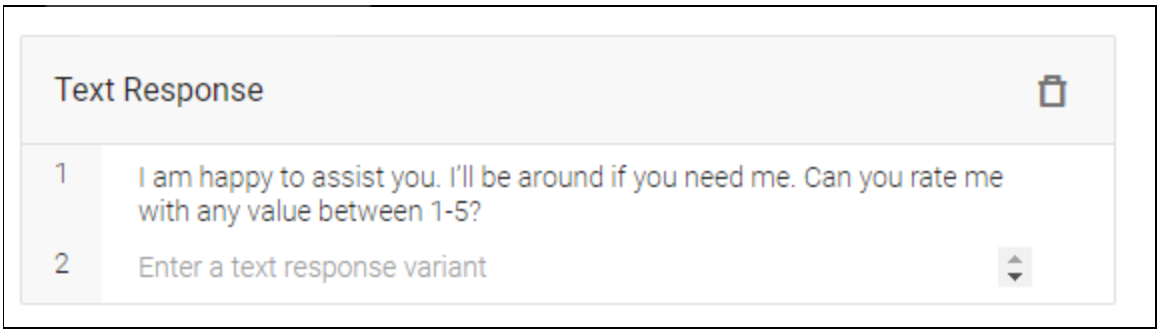
11. Now, include **Responses** to the intent as mentioned in the following image and click on **Add Response** and save the intent



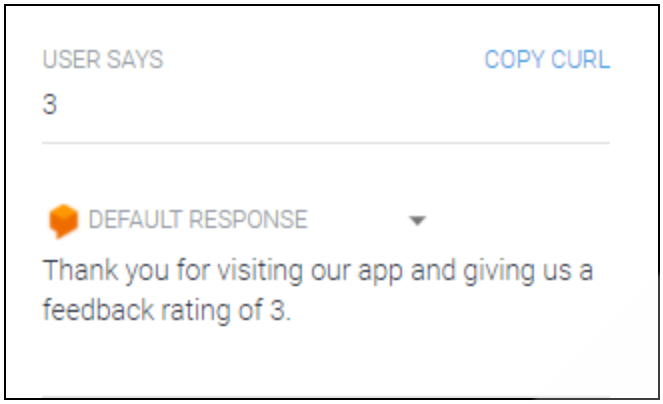
12. Before testing the app, change the response of ending the conversation (**quit**) intent to ask the feedback from the user.



13. Change the response of quit intent as shown in the following screenshot and click on save :



14. Now, test the bot by providing user inputs in the **Try it now** section:



	Great! We have created our own custom entity. We have successfully learned how to train our bot. Now it's your turn. We are going to publish the app in Thunkable. Please share your screen with me.	

- **Ask the students to press the ESC key to come back to the panel.**
- **Guide the students to start a screen share.**
- **The student gets into fullscreen.**

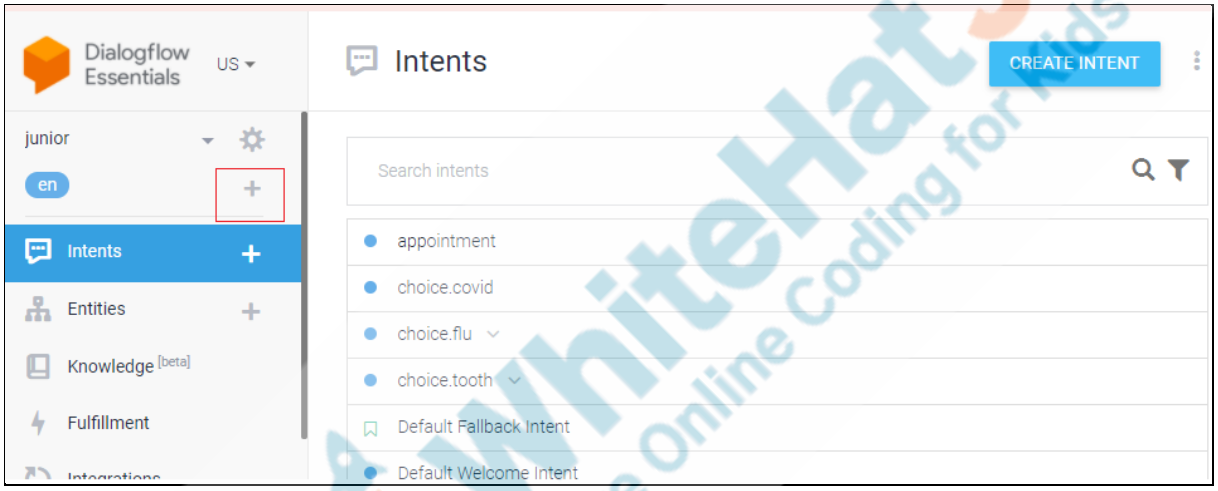
ACTIVITY

- Integrate a chatbot in the app using Thunkable.
- Create a multilingual agent.

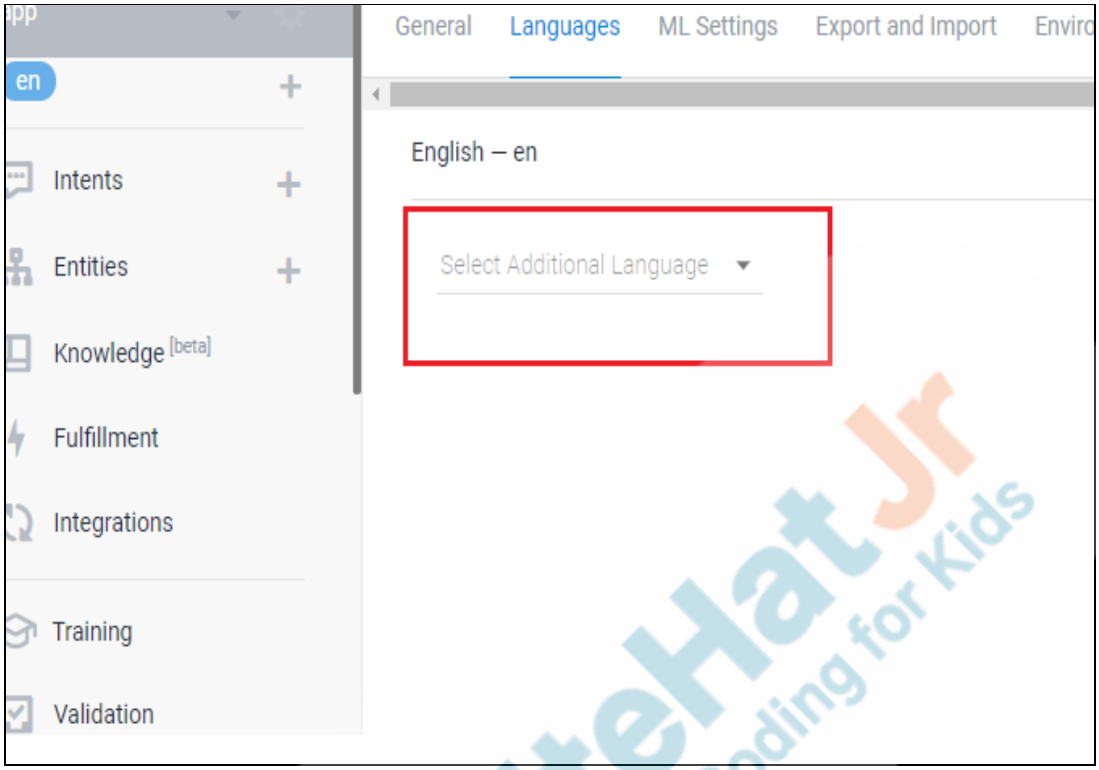
Step 3: Student-Led Activity (25 mins)	*Guide the student to open Student Activity 1.* *Note: Guide the students to recreate the teacher activity for the health bot.* Now, let's learn how to add the multilingual agent. *<Student name 1>*, do you know what is multilingual? Multilingual means many languages are supported by Dialogflow. The language you choose when creating an agent is set as the default language, and you can add additional languages. *<Student name 2>*, do we have to create a separate intent based on language? For intent and entity data, some data would be common for all languages for a multilingual agent, and some data would be language-specific. Language-generic intent and entity data are common for all languages. The following data is language-generic. For example, welcome intent and fallback intent.	*The students open Student Activity 1.* *Students are expected to build the entire bot on their own with the teacher's guidance to make sure the students follow the steps shown above.* **ESRs:** Varied. . **ESRs:** Varied. *Students listen.*

	Language-specific intent and entity data are unique for each language supported by an agent. You must supply this data separately for each language. Let's check the multilingual feature for our welcome intent.	

1. On the left-hand sidebar menu, click on **+** symbol, next to the existing language(s):



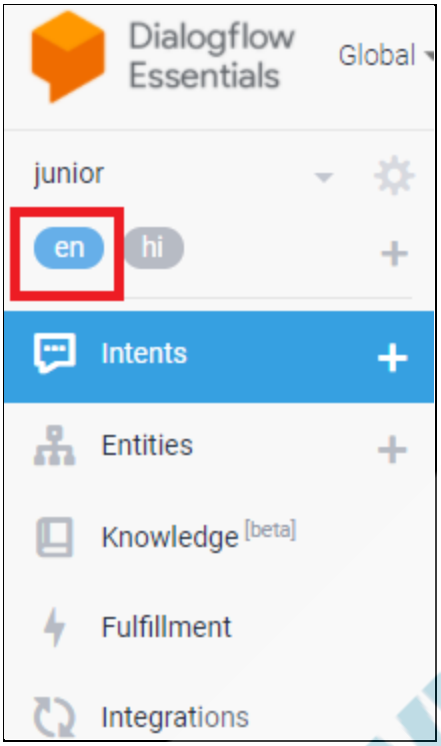
2. Choose a language from the **Select Additional Language** dropdown menu.



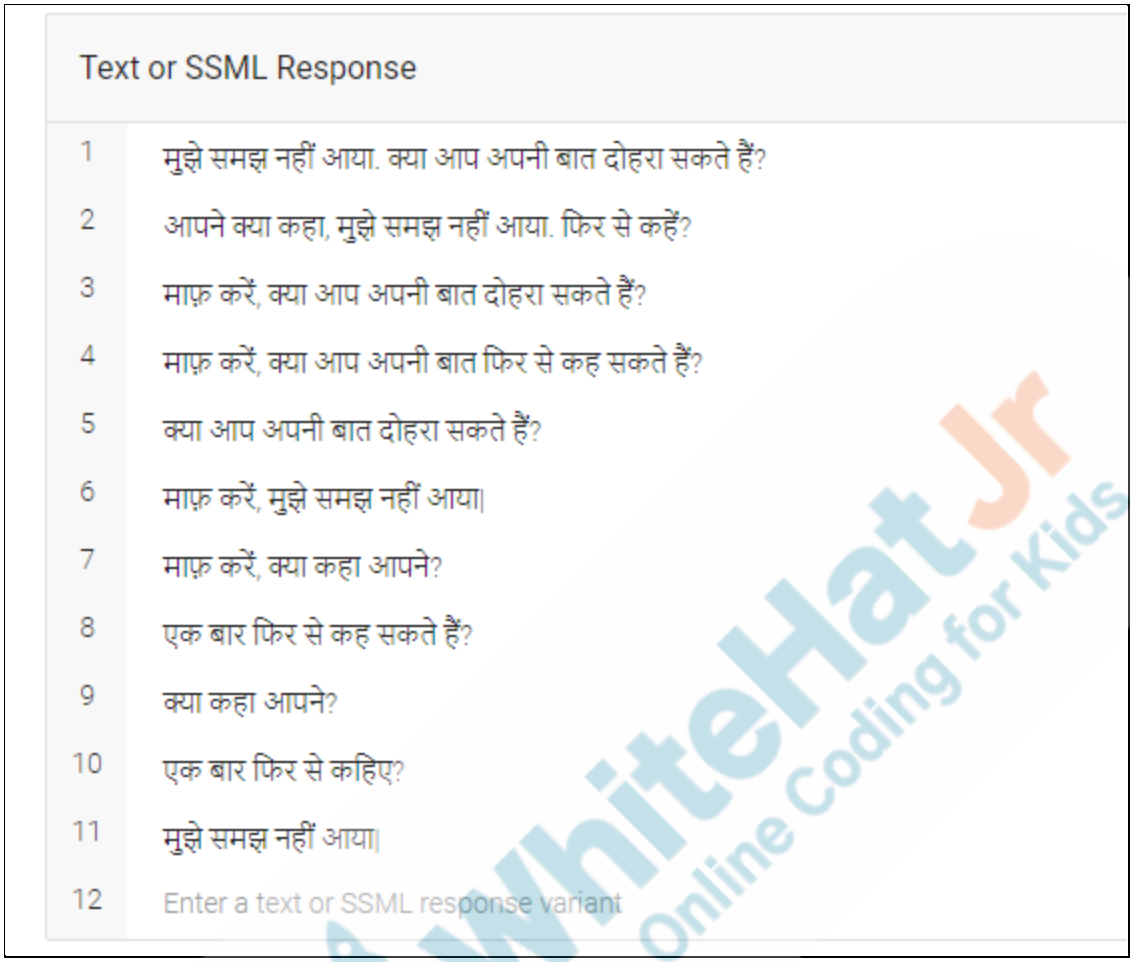
3. Add the language and click on the **SAVE** button.

Note: You are free to choose any language of your choice.

4. To access language-specific data with the console, click the language (**hi**) button under the agent name:



5. Our welcome intent response is automatically converted into Hindi from English.

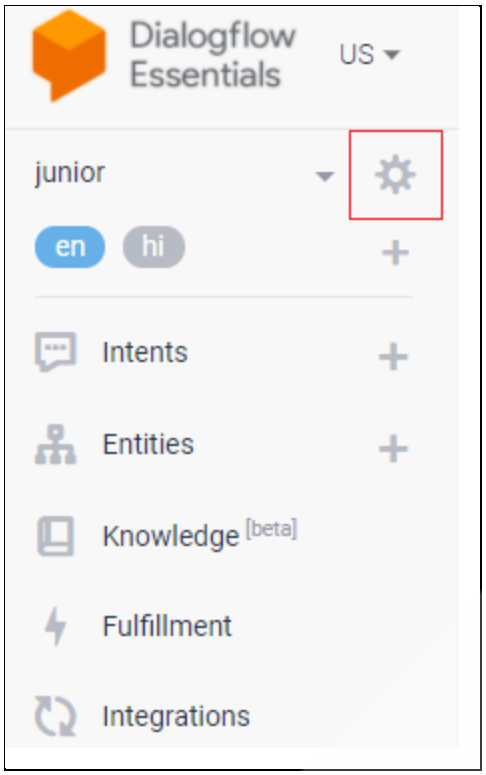


Note:

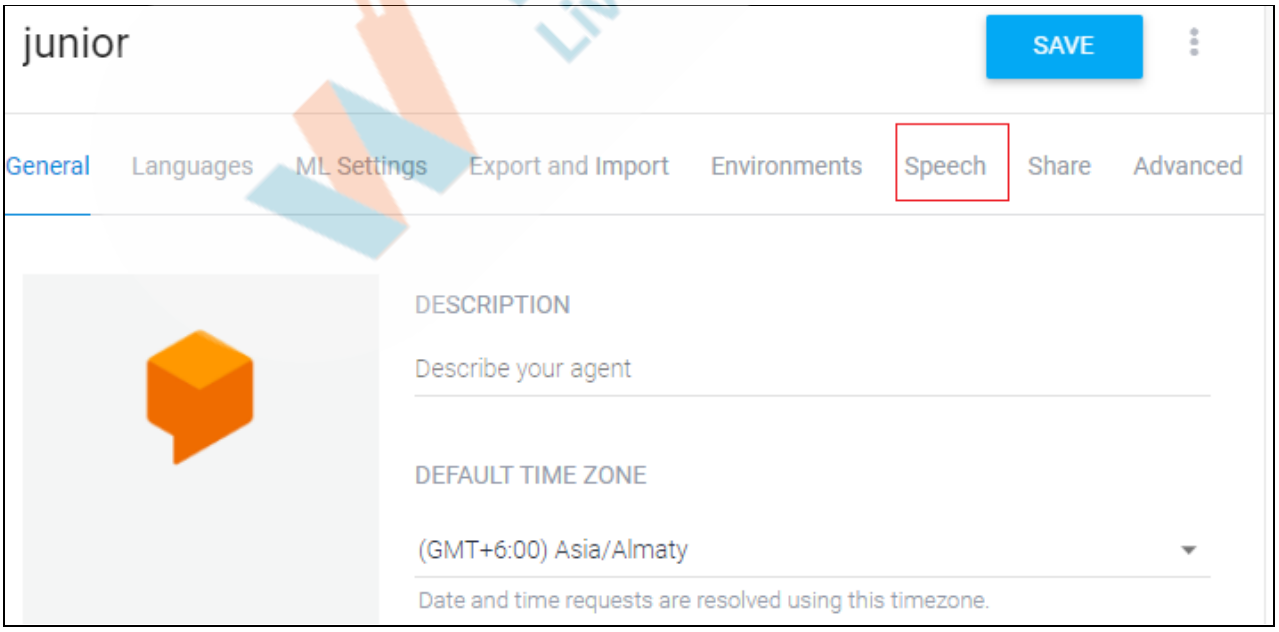
- *We can toggle the language button to use another languages.*
- *We need to add data separately for each language for the language-specific intent.*

	Great! Our chatbot is now multilingual, we can surely add more languages. In addition to the language, won't it be great if we can change the text responses as speech in our conversations with the bot?	*Students listen.*

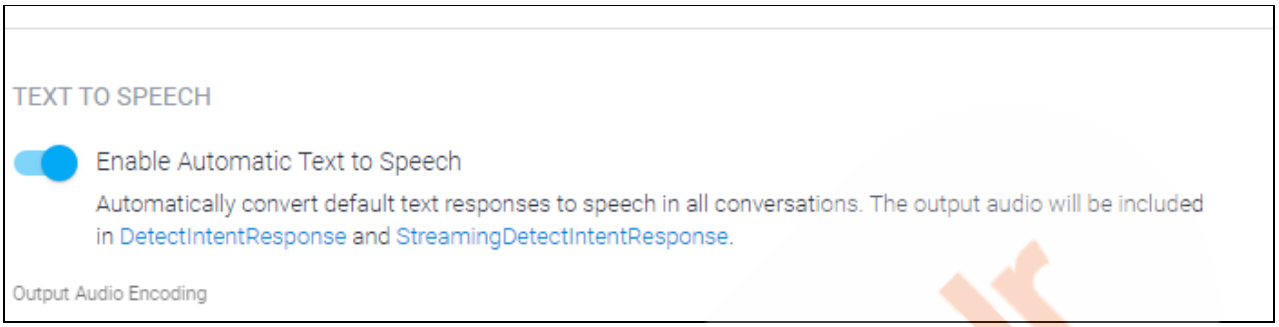
6. Choose the **setting tab** near the agent (**junior**).



7. Choose the **Speech** tab as shown in the following screenshot:

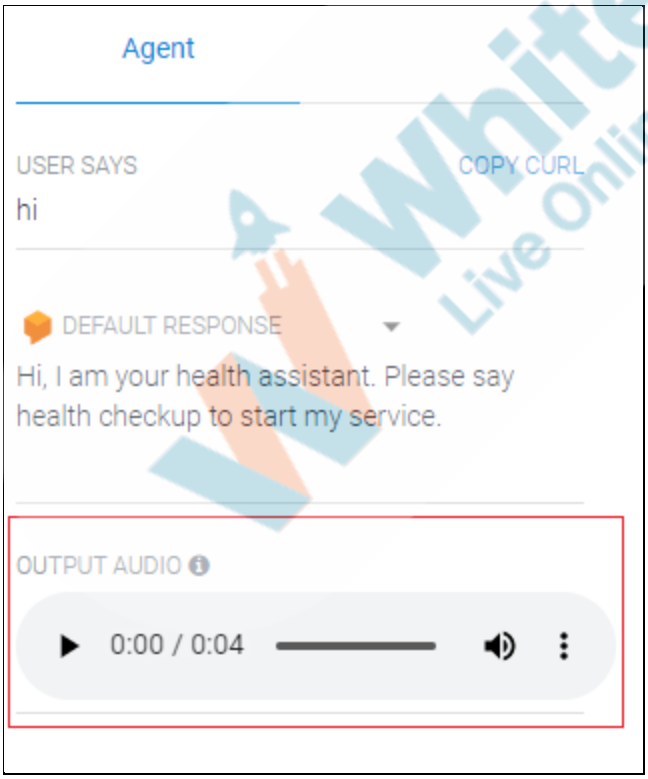


8. Enable the **Enable Automatic Text to Speech** tab. This will automatically convert output responses to speech. Finally, click on the **SAVE** button:



Note: To Enable input response, we need the enterprise edition(paid).

Now test the audio response for the default response by giving the intent on the right side:

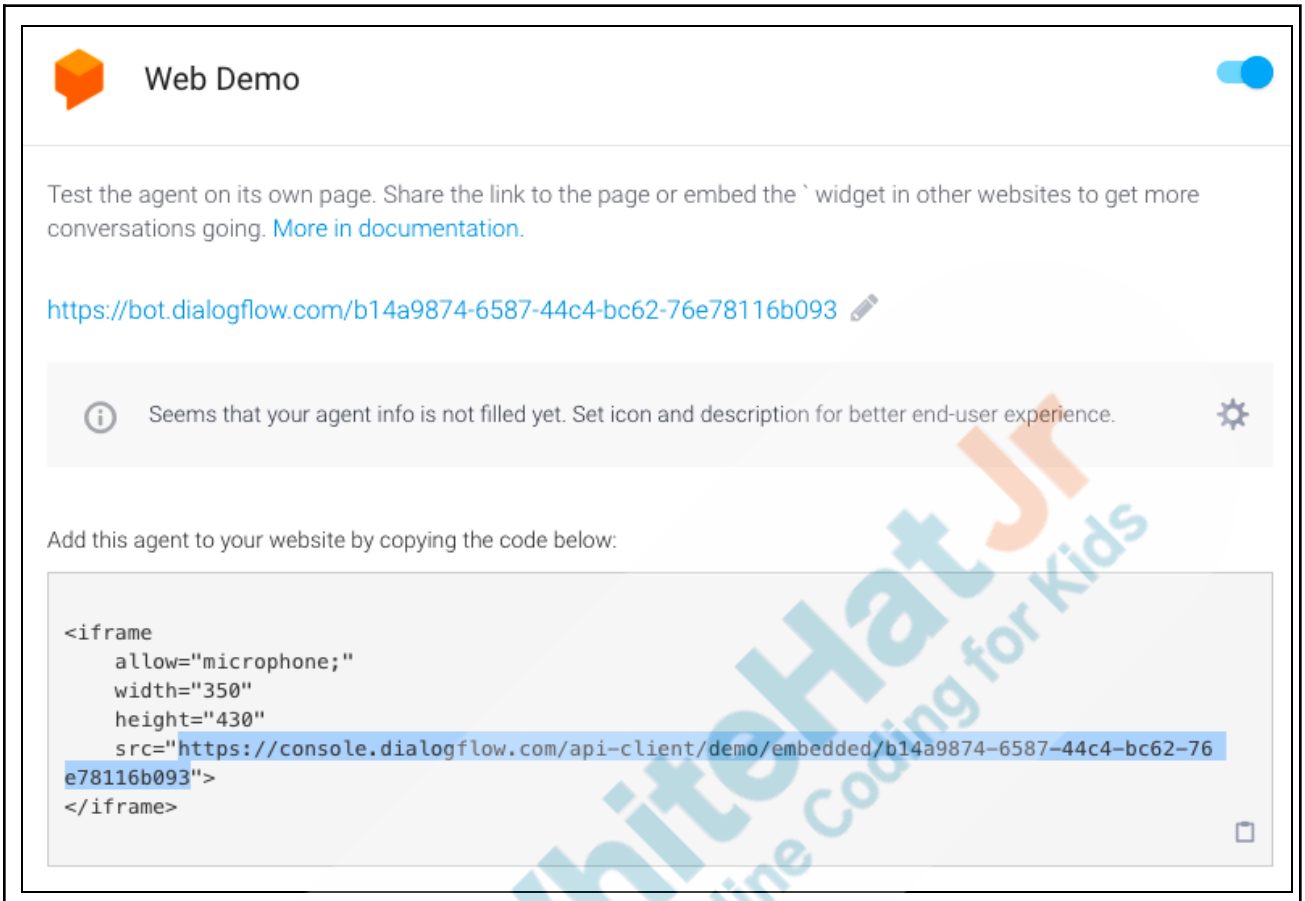


	Our chatbot is ready, we need to now integrate the chatbot to Thunkable.	

	Do you remember how to get started?	**ESRs:** Varied.
	Before publishing it in Thunkable. Let's integrate the app and get the link for publishing. *The teacher guides the student to integrate the bot.*	*Students integrate the bot by following the below step.*

10. Enable **Web Demo** by clicking on the **Integrations** button on the left-hand side tab.

11. You will get a popup showing the link to our bot:



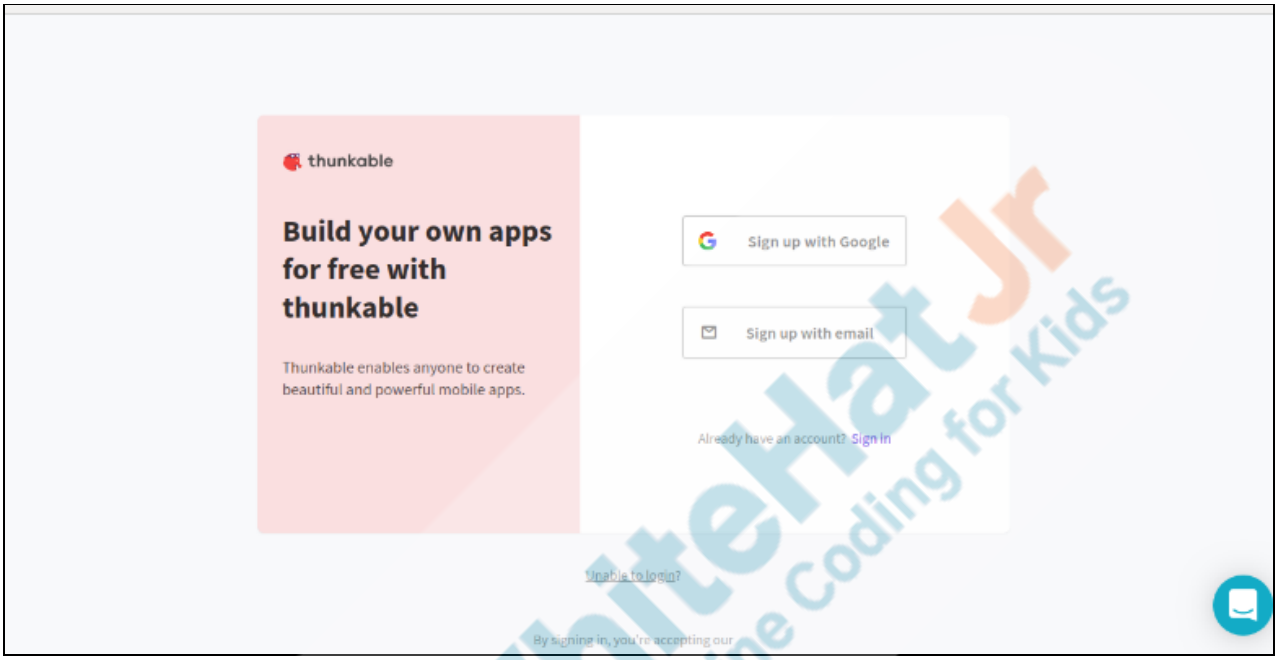
The link mentioned in the **src** or **source** is the bot link.

12.Copy this link, paste it into Thunkable.

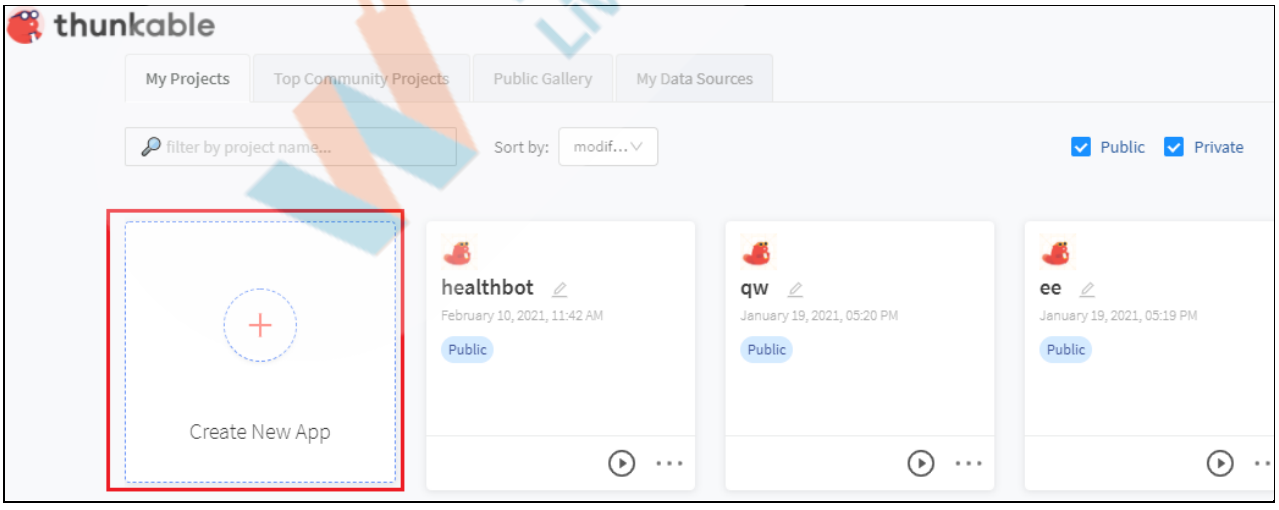
	Do you both remember the Thunkable we have learned for publishing an app for our Trex game? Let's open **Student Activity 2** and the teacher can open Teacher Activity 4 (Thunkable.com). Now, quickly sign up and then log in and create a new project with the name Health bot.	**ESRs**: Varied *The students open **Student Activity 2** and log in to Thunkable.com (using google).*

	Guide the students to sign up and log in to Thunkable.com.	

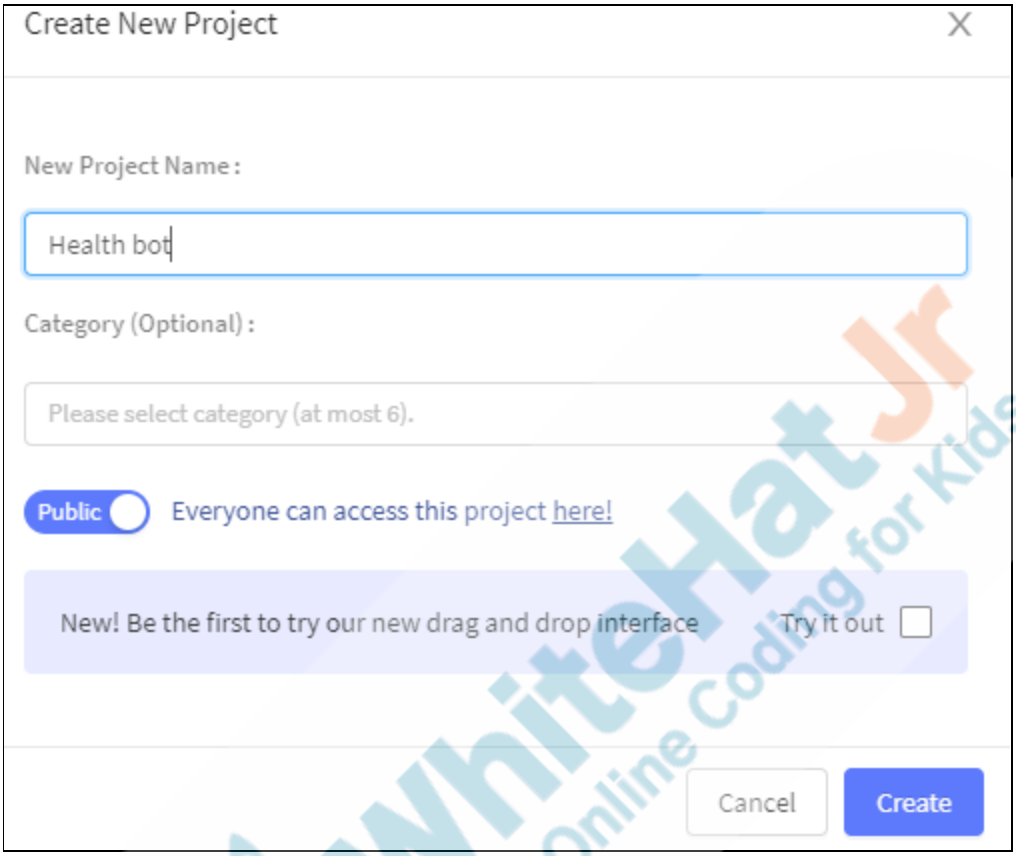
13. Login to your Thunkable account:



14.Choose to **Create New App**:



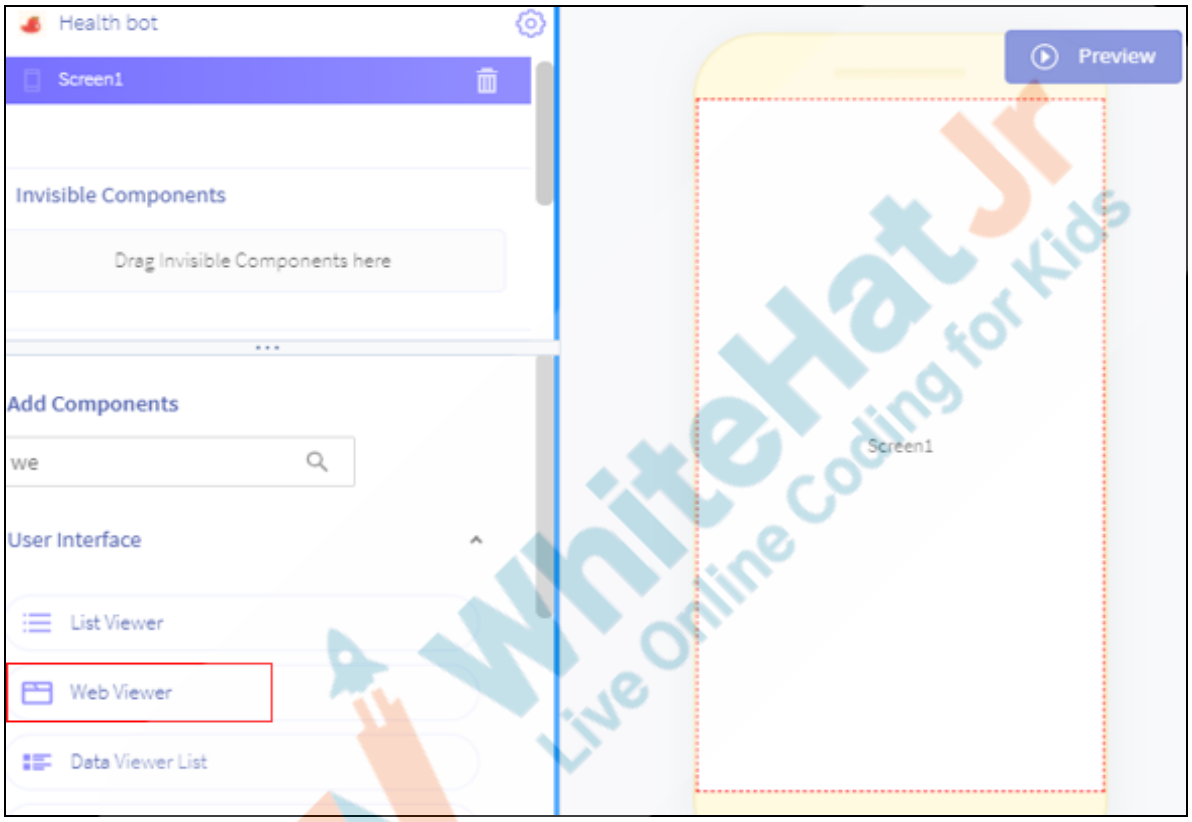
15.Add the project name and click on **Create**:



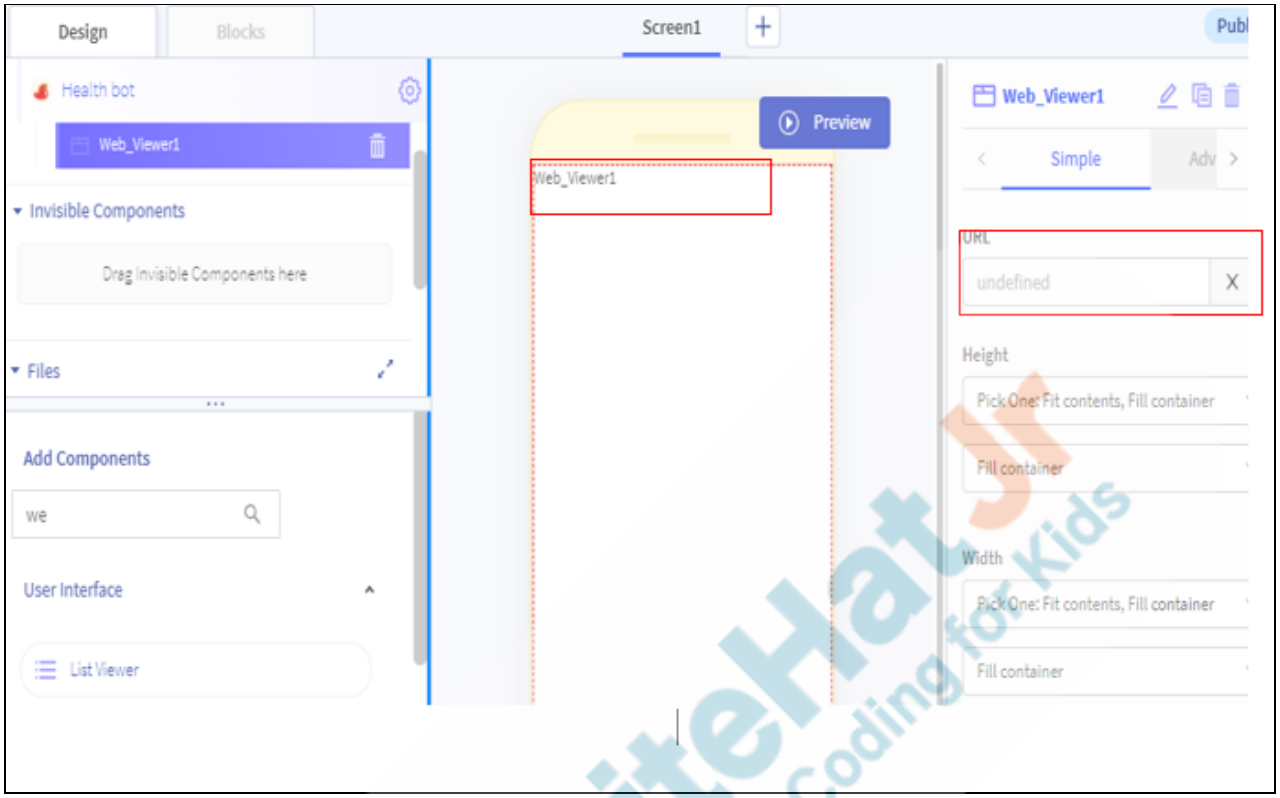
	You will see two tabs - **Design** and **Blocks**. Under the **Design** tab, locate the **Web Viewer** option and drag it onto **Screen 1**, which you see on the right-hand side. You will see an additional web viewer option to the extreme right. *Add the Dialogflow web demo URL here (Student Activity3). The teacher can use Teacher Activity 5.*	*The students follow the instructions to add a Web View component in Thunkable.*

	Note: Students can use their web demo URL or make use of the **Student Activity 3** *link.*	

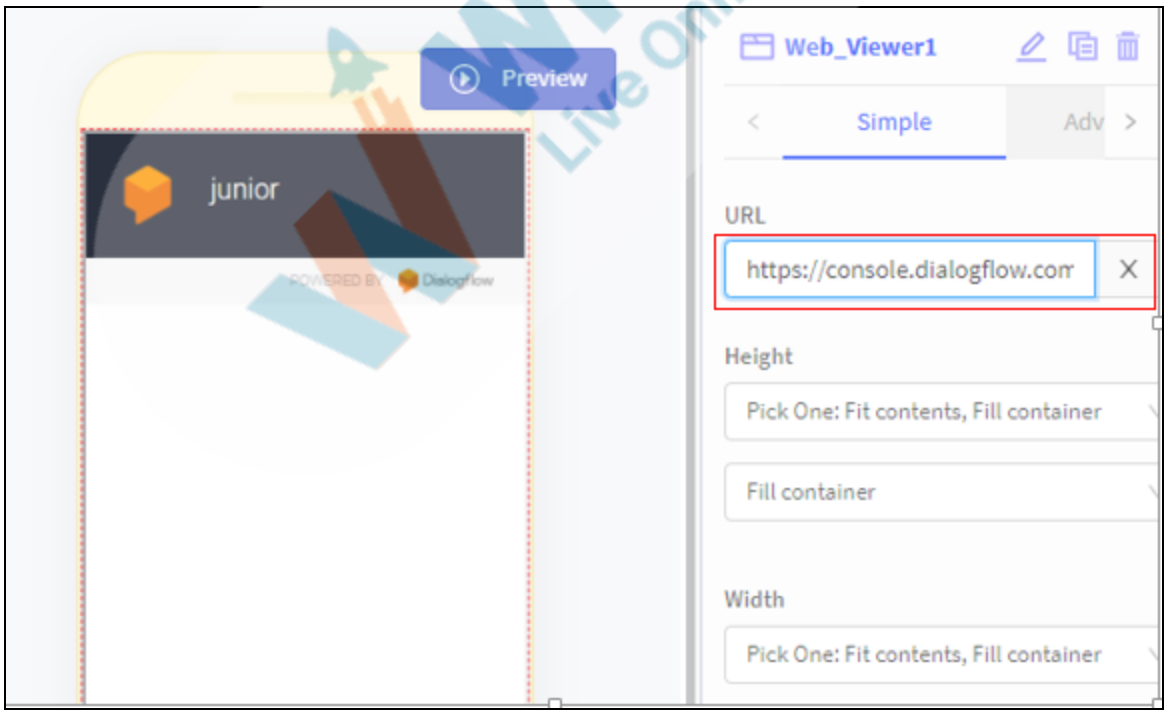
16. Drag and drop the **Web Viewer** option into **Screen1**.

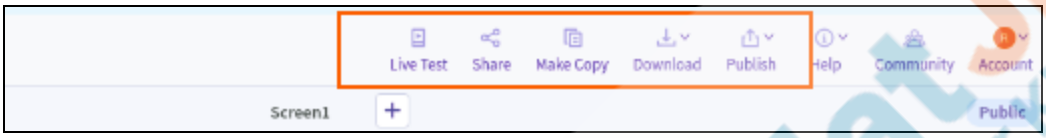


17. Specify the Dialogflow integration URL in the **URL** tab as highlighted in the following screenshot.

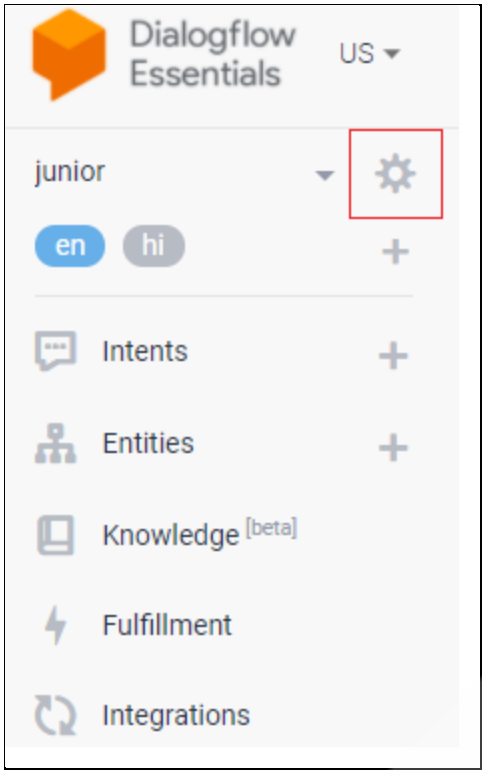


18. Click on **Preview** to view the result:

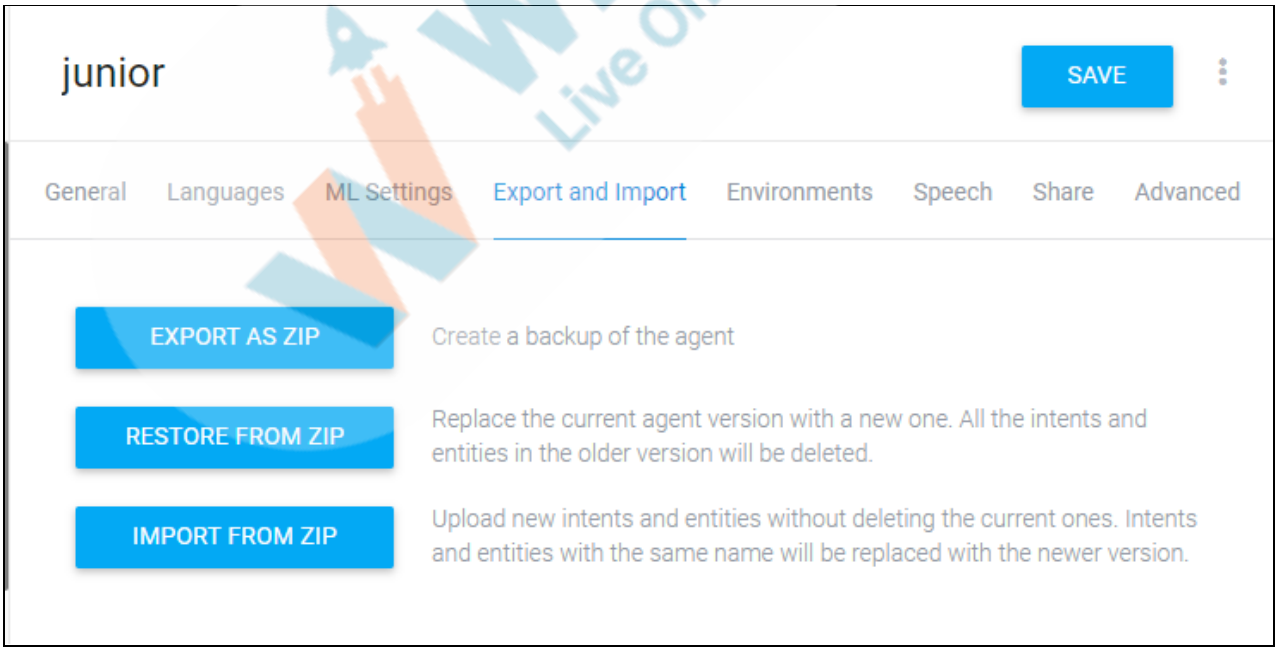


	Amazing! You hit **Preview** and can see the live preview version chatbot on the Thunkable site itself. You can download the Thunkable app on your Android/iOS smartphone, sign up and live test it.	*The students test the app live on the phone if they have access to a smartphone.*
		
	You can use the download button to download the APK file, which can be installed on Android, or the IPA file which can be installed on iOS. *(Note: The students need to wait for some time after pressing the **Download** button for the download action to complete.)* The APK file needs to be transferred and installed on your mobile phone. *The students also have to allow the installation of the file from **unauthorized sources** in the mobile settings if it is not installed.*	*The students use the download option to test the download and installation of the file.*
	Wow! Great! Now we have learned how to create a chatbot API, and we also learned how to publish the app.	

	Good work!	
	<Student name1>, do you wish to share the source file of the bot you have created with your friends? We can do that by clicking the **Export and Import** tab in the settings option in Dialogflow. The following tab options are available: • **EXPORT AS ZIP**: We can export our chatbot and share it with our friends. • **RESTORE FROM ZIP**: We can overwrite the current agent. • **IMPORT FROM ZIP**: We can import another chatbot and add intent or entities to that.	**ESRs:** Yes!
19. Choose the setting options:		



20. Choose the **Export and Import** as per the requirements:



Teacher Guides Student to Stop Screen Share		
FEEDBACK • **Encourage the students to make reflection notes in the markdown format.** • **Compliment the students for their effort in the class.** • **Review the content of the lesson.**		
Step 4: Wrap-Up (5 min)	Did you learn to create additional features for our chatbot and publish it? Let us quickly recollect what we learned and did in today's class. I am going to present some questions on my screen and also read them out to you. You have to tell me the answers.	**ESRs:** Yes!
	<Ask both the students alternatively.>  *Refer to Teacher Resources page/slides 3-5 for the visual aids.* Q1) What are custom entities? Q2) Define map entities.	**ESRs:** Custom entities are used to match data specific to your agent. **ESRs:** Map entities provide a map from reference values to synonyms for each reference value. **ESRs:** A CSV file.

	Q3) Which was the format our spreadsheet had to be saved in?	
	You get Hats Off for your excellent work!	*Make sure you have given at least 2 Hats Off during the class for:* 
	Congratulations! You have set a new benchmark. Brace yourself! Your new challenge is ready. In the next class, we will be learning about creating a Chatbot for ordering food in restaurants.	
Project Overview	**BUS RESERVATION** **Goal of the Project:** In Class 47, we added additional functionalities to our chatbot such as having a multilingual bot and adding a suitable knowledge base.	

We also learned about the entities and integrated the chatbot in Thunkable.

In this project, you will apply what you have learned in the class to achieve the following goals.

Main Goal

- Create a chatbot to book a seat on the bus.

Additional Goal 1

- Integrate the chatbot in Thunkable.

Story:

A reputed travel company in your city is going digital. They have hired you to create a chatbot for them to grow digitally.

Here is what they want you to do:

- Make a Bus Reservation App that helps book a ticket by fetching details of the customer time, destination, and the number of passengers.
- Integrate your Bus Reservation App in Thunkable.

I am very excited to see your project solution and I know you both will do really well.

	Bye Bye!	
Teacher Clicks 		
Additional Activities - I	*Encourage the students to write reflection notes in their reflection journals using Markdown.* Use these as guiding questions: ● What happened today? ○ Describe what happened. ○ The code I wrote. ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? What did I find difficult?	*The students use the markdown editor to write their reflections in a reflection journal.*

Activity	Activity Name	Links
Teacher Activity 1	Teacher Resource	https://s3-whjr-curriculum-uploads.whjr.online/69c11876-b1b3-44f5-ae4c-c5d2d129e02b.pptx
Teacher Activity 2	Dialogflow	https://dialogflow.cloud.google.com/
Teacher Activity 3	Previous class	https://github.com/rashmi-sathya/Healthbot/blob/main/HealthBot%20(1).

		zip
Teacher Activity 4	Thunkable	https://Thunkable.com/#/
Teacher Activity 5	Web demo	https://console.dialogflow.com/api-client/demo/embedded/66befcb6-4a9d-4824-a90e-4e899a696272
Student Activity 1	Dialogflow	https://dialogflow.cloud.google.com/
Student Activity 2	Thunkable	https://Thunkable.com/#/
Student Activity 3	Web demo	https://console.dialogflow.com/api-client/demo/embedded/66befcb6-4a9d-4824-a90e-4e899a696272