

Driver Application with AI framework for a Chat-bot

Kartik Nagesh Pauskar 191IT223

Information Technology

National Institute of Technology Karnataka

Surathkal, India 575025

Email: kartikpauskar.191it223@nitk.edu.in

Shashikant Kumar 191IT249

Information Technology

National Institute of Technology Karnataka

Surathkal, India 575025

Email: shashikantkumar.191it249@nitk.edu.in

Akshay Sreekumar Nair 191IT103

Information Technology

National Institute of Technology Karnataka

Surathkal, India 575025

Email: a.191it103@nitk.edu.in

Shiv Kumar Dubey 191IT250

Information Technology

National Institute of Technology Karnataka

Surathkal, India 575025

Email: shiv.191it250@nitk.edu.in

Abstract—The application helps a user to book a driver if one has a vehicle but has no-one to drive it, and also book different cars for many scenarios. The application is built using VS code, is written in HTML and CSS, along JAVASCRIPT. For the back-end, total 7 modules are being used for proper collection of data entered by the user. Also a Chat-Bot is integrated into the application for more interactive user experience.

I. INTRODUCTION

This Web application helps a user to book a driver if one has a vehicle but has no-one to drive it, it allows a user to get a driver for many scenarios' like driver for daily basis, for loading and transferring heavy furniture's to a distant place, also off trips and off-roading, etc.

This application also lets you book cars for your journey, weather for daily basis or for somewhere adventurous. It lets you choose from a variety of different cars like Luxurious Premium Cars, Classic Sedans, Micro cars, Electric cars, Cabs and even Off-Roading vehicles.

Now for the technical stuff, we have built this application from the front-end to the back-end and made it user friendly by making it easier for a user to use. The application is loaded with a front-end where the user can interact with the application, can sign-up or login and has a variety of options to make a booking. Also we have a Chat-Bot to make this application even more interactive and easy to use.

II. LITERATURE SURVEY

This application is designed and built keeping in mind today's modern web applications which allows users to work around the application with minimal efforts thus making it more user friendly. That is why the integration of the chat-bot is a crucial part in the application, where a user can interact with the application. We conducted researches regarding making an application more intuitive and simple to use, yet be as modern and clean looking as possible, also researches

for building a chat-bot were conducted and were able to build and integrate one in the application.

- AI-based Chabot's in customer service and their effects on user compliance
Martin Adam, Michael Wessel, Alexander Benlian
- Design and implement a chatbot for e-commerce
Amir-reza Asadi, Reza Hemadi.
- An E-commerce website based chatbot
Siddharth Gupta, Deep Borkar, Chevelyn De Mello, Saurabh Patil.

III. OUTCOME OF LITERATURE SURVEY

The previous researches conducted stressed on the same fact of efficiency and more effective customer satisfaction with the use of chatbot in a website, which we too adhere to. We were able to design and build a web application based on the conducted research and make a clean and simple to use web application.

IV. MOTIVATION

- Encountered people facing this problem
- This problem created an opportunity
- And we decided to tackle on and started to implemented right away.
- This issue was prevalent in many areas and ignored too. We conducted surveys and concluded to solve this issue using whatever we have in our hands

V. PROBLEM STATEMENT

In remote places, finding driver is difficult. So, we planned on implementing a solution for this scenario. Our application also focuses on providing employment for the driver as well as ease customers to find drivers. Freelance drivers are to benefit the most of it, as this would ensure an almost regular source of income for them. Provide services at affordable price is our top priority. AI chat-bot would ensure fast access to the app/site and also solve tier-3(easy) problems faced.

VI. OBJECTIVES

- 1) Solve daily-life problem (as mentioned above) faced by the people.
- 2) Build a user-friendly website, to focus all sorts of users.
- 3) Fast response and efficient work.
- 4) Integrate an Effective chat-bot for an interactive web application

VII. PROPOSED WORK

- 1) To design an interactive and user friendly interface
- 2) To build a modern and intuitive UI design
- 3) To design and integrate a Chat-bot to make the application interactive
- 4) To build a back-end where user data can be stored safely
- 5) To make the application safe, where user the passwords entered by the users are encrypted

VIII. METHODOLOGY

A. Designing the Front-End

The Front-end is the most important aspect of a Web application, it's like the face of an application where all the work is done. So the Front-end of this application is designed keeping in mind the user experience and how do we make it more interactive and easier to use.

Also these pages are routed in all the possible combinations so that a user can navigate inside the application with an ease.

The application is designed to make it responsive to different screen sizes. The front-end is written in HTML and CSS and the navigation is the done using JAVASCRIPT, also various modules like NodeJS, ExpressJS, etc. are used.

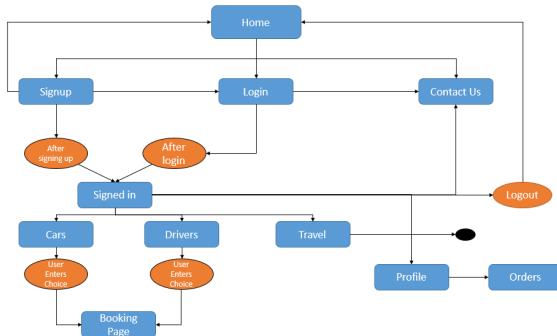


Fig. 1. Layout of the application

The application comprises of 10 pages which will guide the user to make the Booking, namely:

- 1) The Home page

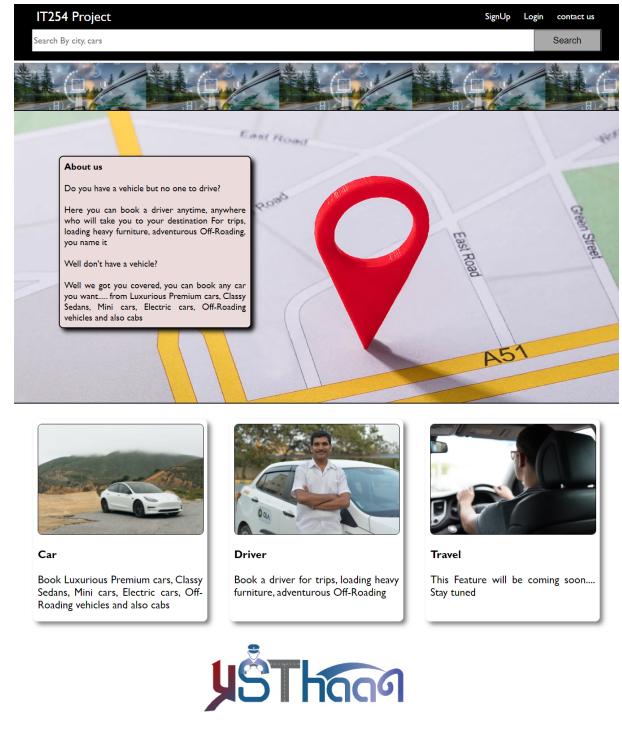


Fig. 2. Home page



2) A Sign-Up page

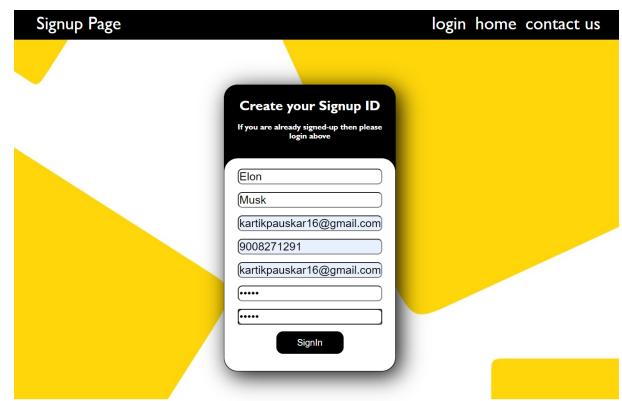


Fig. 3. SignUp page

3) A Login page

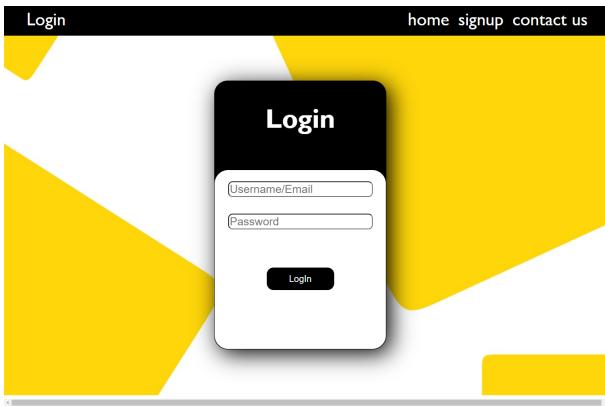


Fig. 4. Login page

- 4) A Cars page – Where a user can choose from a variety of cars to book depending on the need.

Fig. 5. Cars page

- 5) A Drivers page – Where a user can book a driver based on the need.

Fig. 6. Drivers page

- 6) A Booking page – Where a user can book from the options provided.

Fig. 7. Booking page

- 7) A Signed in page – It's like the user's own space within the application.

Fig. 8. Signed in page

- 8) An About Us page – Which contains the data of the team.

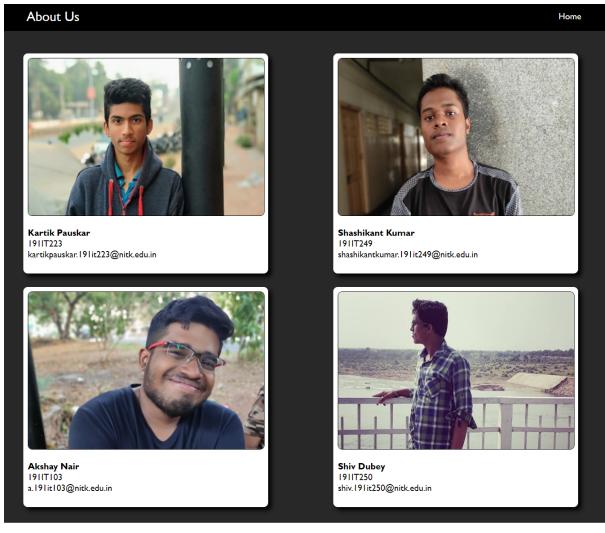


Fig. 9. About Us page

9) A Profile page – Where a user can view their data.

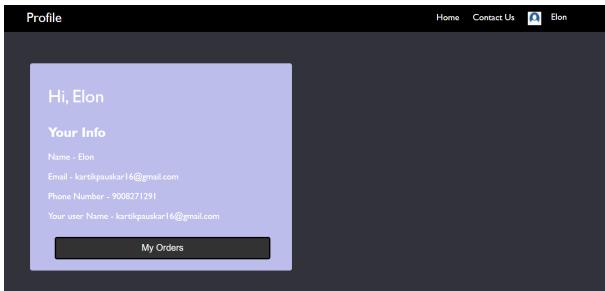


Fig. 10. Profile page

10) A My Orders page – Where a user can view the details of the orders they have booked.



Fig. 11. My Orders page

B. The Back-End

For Backend we took the aid of total 7 external modules to full fill the requirement of the project.
Namely:

- 1) Mongoose- Mongoose is an Object Data Modeling (ODM) library for MongoDB and NodeJS. It basically manages relationships between data, is responsible for schema validation. Now the database which we used for the project is MongoDB. Now MongoDB is a schemaless NoSQL document database.
- 2) BcryptJS – Now as safety is the number 1 priority, we use BcryptJS, which basically encrypts the passwords or sensitive data entered by the user for booking purposes and converts it into some encrypted code.
- 3) JSON web token - The JSON web token (JWT) is a method which allows authentication, without actually storing any information about the user on the system.
- 4) Cookie Parser – This is a middleware which parses cookies attached to the client request object. this can be used the same way as we use other middleware. Here is a sample syntax: `var cookieParser = require('cookie-parser');`
- 5) Express JS – Express is one of the most important modules which provides a special set of features to develop web applications.

Now these are some core features of Express framework:

- Set up middleware's to respond to HTTP Requests.
- Defines a routing table which is used to perform different actions based on HTTP Method and URL.
- Dynamic rendering of HTML Pages based on passing arguments to templates.
- 6) Validator - Form validation normally takes place at the server, after the client enters all the necessary data and submits the data. If the data entered by a client is incorrect or is missing, the server would have to send back all the data to the client and request to resubmit the form. JavaScript provides a way to validate form's data on the client's computer before sending it to the web server. Here we can performs two functions.
 - Basic Validation The form must be validated to make sure all the mandatory fields are filled in. It

would require just a loop through each field in the form and check for data.

- Data Format Validation – The data that is entered must be validated for correct values with appropriate logic to test the correctness of data.

The screenshot shows a booking page titled "Booking Page". At the top, there are links for "Contact Us", "home", and "Logout". Below these are search fields for "Search By city, cars" and a "Search" button. The main area features a white Tesla Model 3 parked on a coastal road. To the left, a modal window titled "Please Enter Your Booking Details" contains the following input fields:

- Name: Elon
- Email: kartikpauskar16@gmail.com
- Phone: 900909090
- Age: 12
- Address: Nitk, Mangalore
- Dept: IT dept
- Pincode: 28912
- Car Type: 4
- Date: 24-Apr-2021
- Time: 12:47 PM
- Book Now



Fig. 12. Data entered by the user

The screenshot shows the MongoDB Compass interface connected to localhost:27017/IT254PROJECT. The "orderdetails" collection is selected. The table displays two documents with the following data:

```

[{"_id": "61f0507e700d164694455474", "booking_time": "2021-04-24T12:47:00Z", "customer_email": "kartikpauskar16@gmail.com", "customer_name": "Elon", "customer_pincode": "28912", "customer_type": "4", "dept": "IT dept", "pincode": "28912", "tokens": [{"token": "A"}, {"token": "B"}]}, {"_id": "61f0507e700d164694455475", "booking_time": "2021-04-24T12:47:00Z", "customer_email": "kartikpauskar16@gmail.com", "customer_name": "Nitk", "customer_pincode": "575009", "customer_type": "4", "dept": "IT dept", "pincode": "575009", "tokens": [{"token": "C"}, {"token": "D"}]}]

```

Fig. 13. Data received at the backend

C. Chat-Bot

The Chat-Bot is an important aspect of this project, as it makes it more interactive and easy to get information about the application. We have used created a framework that will allow users to choose and select the actions they want the Chat-Bot to perform. The bot is loaded with various keywords and quick toggles which makes the process more responsive.

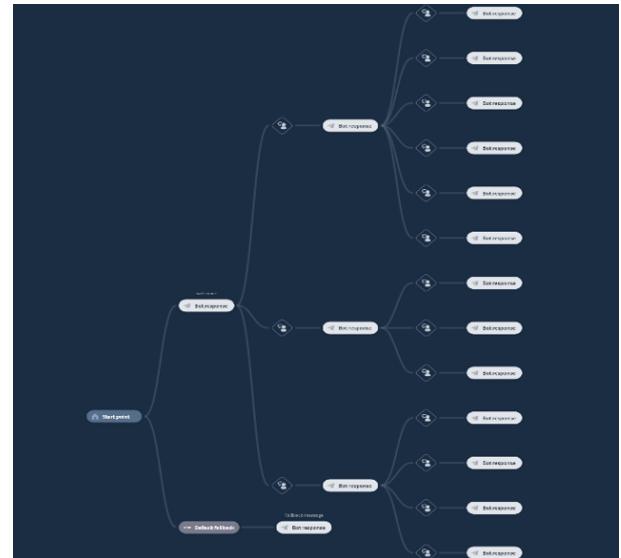


Fig. 14. Layout of the Chat-Bot commands

The working of the Chat-Bot in the application:

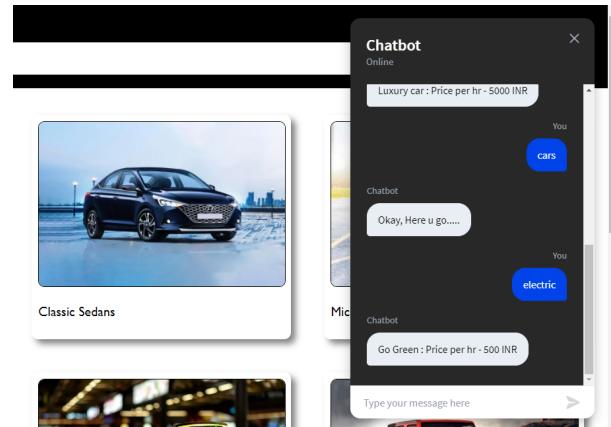


Fig. 15. Integrated Chatbot working

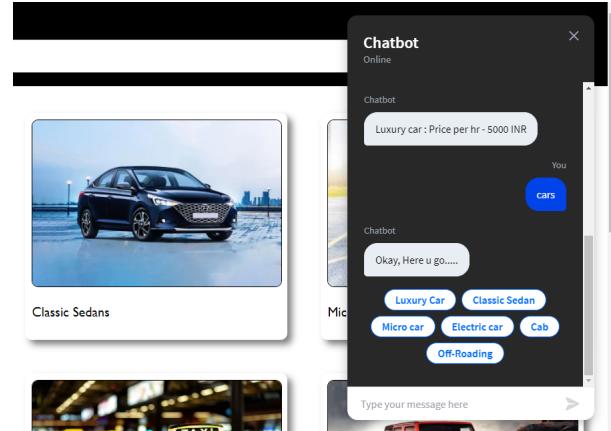


Fig. 16. Integrated Chatbot working

D. Chatbot in python using natural language processing and Neural networks

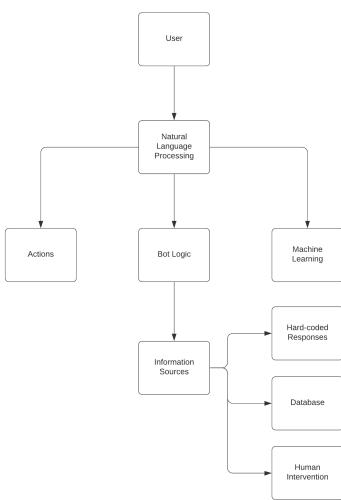


Fig. 17. Flowchart for Chatbot with AI framework

1) Created dataset:

The dataset file consists of three pieces of information - tag, patterns, responses. Patterns specify what the user might have requested, Responses have hard-coded value which the chatbot will respond to and Tag specifies which patterns and responses come under this category.

2) Formatted data for use:

Reduce the words to its stem so that we don't lose any performance because it's looking for the exact word. Pass that text to the loads function and as a result we get the json object which is a dictionary essentially in python.

3) Trained Data.

4) Frame Neural Network.

5) Format data for feeding into Neural Network.

6) Tokenized strings into words for parsing the data in an efficient manner.

A neural network needs numerical values; use bag of words. Individual word values to either zero or one depending on if it's occurring in that particular.

7) Work with the training list

work with this training list in order to train the neural network.

- Pre-processing before we get into building the neural network.
- Random.shuffle the training data.
- And then we're going to also turn it into a numpy array.
- And then we're going to split it into x and y values.

8) Build the Neural Network

Create a simple sequential model and add a couple of layers. Specify their neurons quantity, activation function, learning rate and other parameters.

9) Feed data into Neural Network.

10) Compile the model.

11) Create Chatbot application that uses the trained model

Load the model, Read file in reading binary mode Use a bunch of functions because the feeded data is numerical.

- A Function to clean up the sentences.
- A function for getting the bag of words.
- A function for predicting the class based on the sentence.
- A function for getting a response.

We specify an Error Threshold that allows for a certain uncertainty but if that uncertainty is too high we're not going to take it into a result so threshold.

12) User feeds his data.

13) Output based on calculated probabilities.

The screenshot shows a Windows desktop with Visual Studio Code open. The terminal window displays the output of the 'training.py' script. The output lists numerous training steps, each showing a pattern, its associated tag, the input text, and the accuracy of the prediction. The accuracy values fluctuate between 0.8000 and 1.0000, with most entries being 1.0000. The script uses TensorFlow's Keras API to load and train a model named 'chatter.h5'.

Fig. 18. Training.py - training the data to be fed into the chatbot

The screenshot shows a Windows desktop with Visual Studio Code open. The terminal window displays the output of the 'Chatbot.py' script. The output shows interactions with the chatbot, such as asking for car details and receiving responses like "I want a driver". The script uses TensorFlow's Keras API to load a model named 'chatter.h5' and processes user input through a loop.

Fig. 19. Chatbot.py - working of the chatbot

IX. FUTURE WORK

We plan to add a new service to the application, where users can book a vacation or a small trip.

Further improvements to the chatbot to be made by,

- Adding more training data, more patterns, more responses.
- Make text more intense.

X. CONCLUSION

Designed and built a web application for booking a driver and vehicles based on the need of the client. Designed it to

be responsive making it more interactive. Used MongoDB as a database for the project along with many important modules for building a back-end for the application. Integrated a Chat-Bot that can provide necessary information for the ease of the users.

XI. TIMELINE OF PROJECT

Object	Duration
Discussion and Research	Week-1
Front-end development	Week-2
Back-end development	Week-3
Back-end development	Week-4
AI Chat-bot Development	Week-5
Integration	Week-6

Fig. 20. Timeline

XII. INDIVIDUAL CONTRIBUTION

1) Kartik Pauskar:

- Basic Research
- Front-end Designing – Cars and Drivers page
- Back-end Design – ExpressJS and Routing
- Designing the layout commands for Chat-bot Integration
- Chat-Bot integration
- Report Making
- Demo video

2) Shashikant Kumar:

- Basic Research
- Front-end Designing – Profile and Orders page
- Back-end Design – Mongoose and Routing
- Building a python chatbot using AI framework
- Debug Chatbot
- Report Making
- Building the presentation

3) Akshay Sreekumar Nair:

- Basic Research
- Front-end Designing – Home and Signed page
- Back-end Design – Cookie parser and BcryptJS
- Building a python chatbot using AI framework
- Chatbot Dataset, building model
- Report Making
- Demo video

4) Shiv Kumar Dubey:

- Basic Research
- Accumulate Data Based on Research
- Front-end Designing – Signup and Login page
- Back-end Design – Validator and JSON web tokens
- Chat-Bot integration
- Report Making
- Building the presentation

ACKNOWLEDGMENT

A Special Thanks to Priyadarshini Ma'am for giving your valuable opinions and guiding us with this project.

RESOURCES

- VS Code for designing front-end.
- MoongoDB for Back-end
- Modules used:
 - Mongoose
 - BcryptJS
 - JSON web token
 - Cookie Parser
 - Express JS
 - Validator

REFERENCES

- [1] Scrip
<https://www.scirp.org/journal/PaperInformation.aspx?PaperID=77941>
- [2] Academia
<https://www.academia.edu/Documents/in/Chatbot>
- [3] Research Gate
https://www.researchgate.net/publication/A_Chatbot_system_for_Mental_Healthcare_Based_on_SAT_Counseling_Method
- [4] Research Gate
https://www.researchgate.net/publication/Programming_challenges_of_chatbot_Current_and_future_prospective
- [5] Ijert
<https://www.ijert.org/an-analytical-study-and-review-of-open-source-chatbot-framework-rasa>
- [6] AI-based Chabot's in customer service and their effects on user compliance
Martin Adam, Michael Wessel, Alexander Benlian.
- [7] Design and implement a chatbot for e-commerce
Amir-reza Asadi, Reza Hemadi.
- [8] An E-commerce website based chatbot
Siddharth Gupta, Deep Borkar, Chevelyn De Mello, Saurabh Patil.