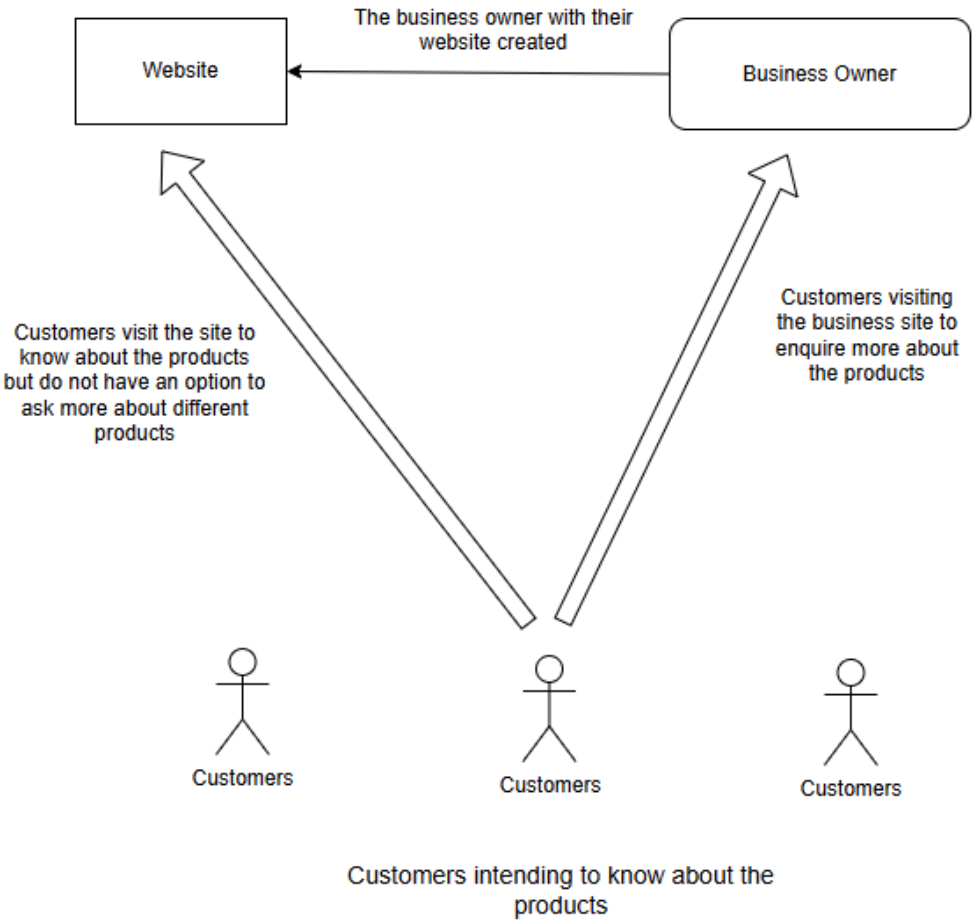
Business AI Receptionist

Table of contents

Overview

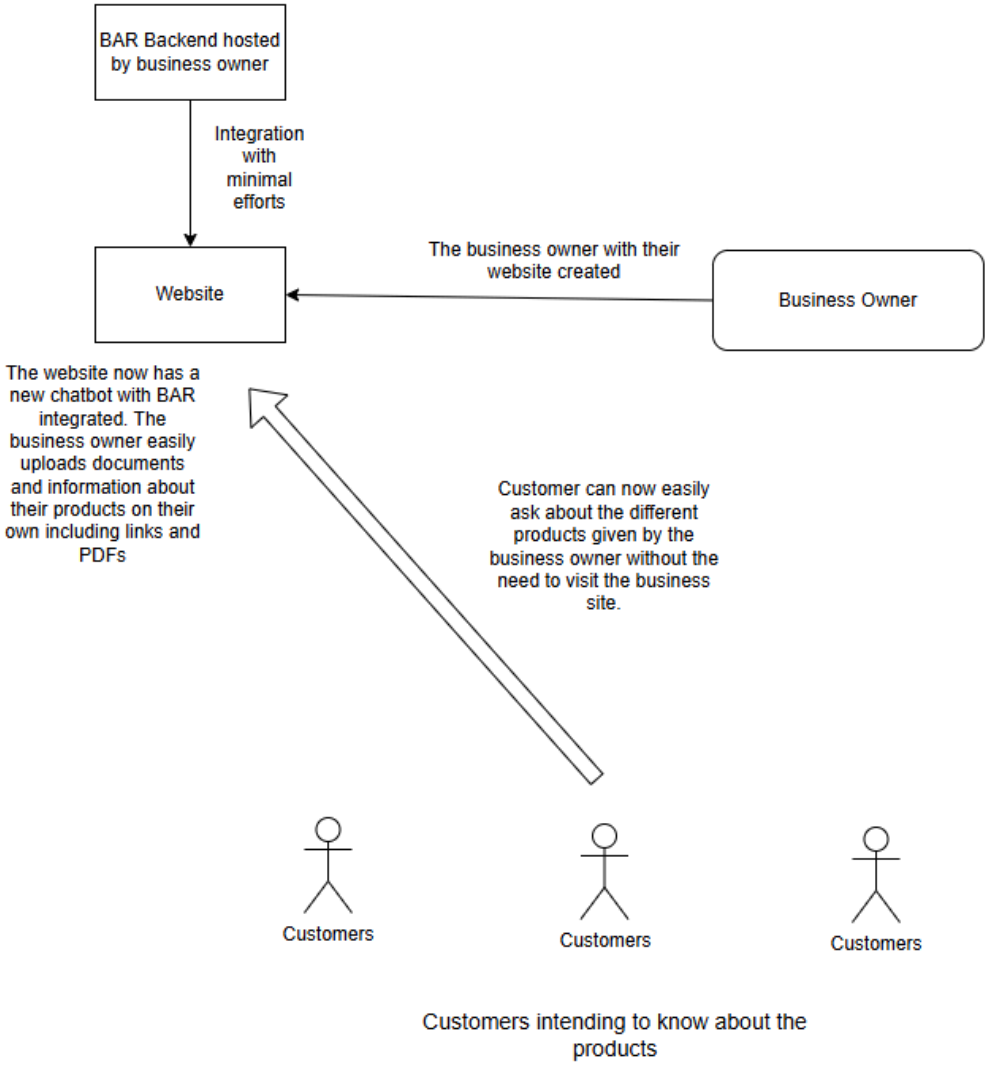
Version: 1.1

In a typical scenario for small business owners (as well as larger business owners in many cases), there is a website present for the general customers highlighting the different products. Customers visit the site and get to know about the different products sold or provided by the business. For more queries, customers either have to visit the business location or contact the establishments via call or emails. At time, this can be cumbersome since calls and emails take time, while visiting the sites may not be feasible all the time.



Thus, introducing Business AI Receptionist, from Baideac!

Business AI Receptionist (BAR) is an easily integrable chat assistant which can be used by business owners on their websites. With some easy steps to set up and connect, businesses can easily add their details about the products, links, PDFs or even videos (with voice). Once done, the BAR will be able to handle all queries from the customers visiting their sites. Thus, the need to constantly respond for mere “Whats” and “Hows” will significantly drop since BAR will do all the hard work of fetching the details and responding. What’s more!! The BAR can work even without the compulsory need of expensive GPUs.

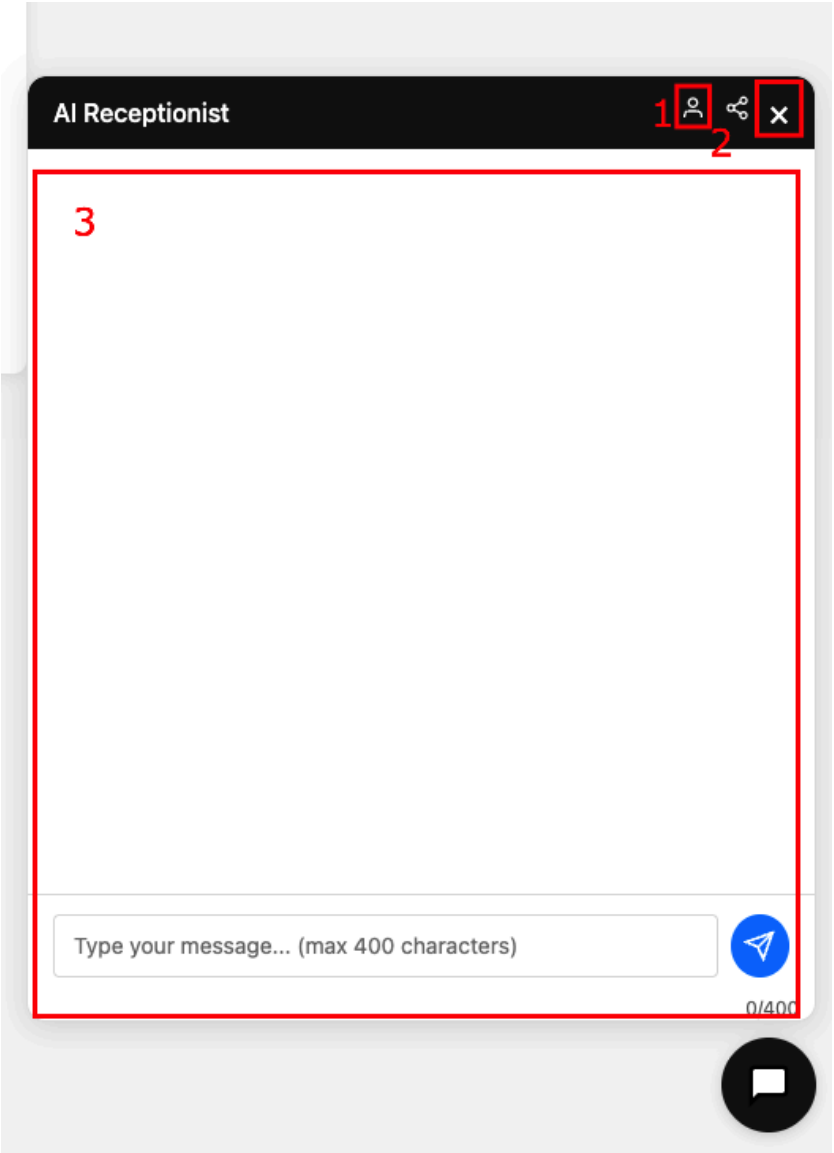


This guide aims to thoroughly introduce Business AI Receptionists. By the end of this guide, one will have an understanding of how to deploy and manage our BAR.

How it looks

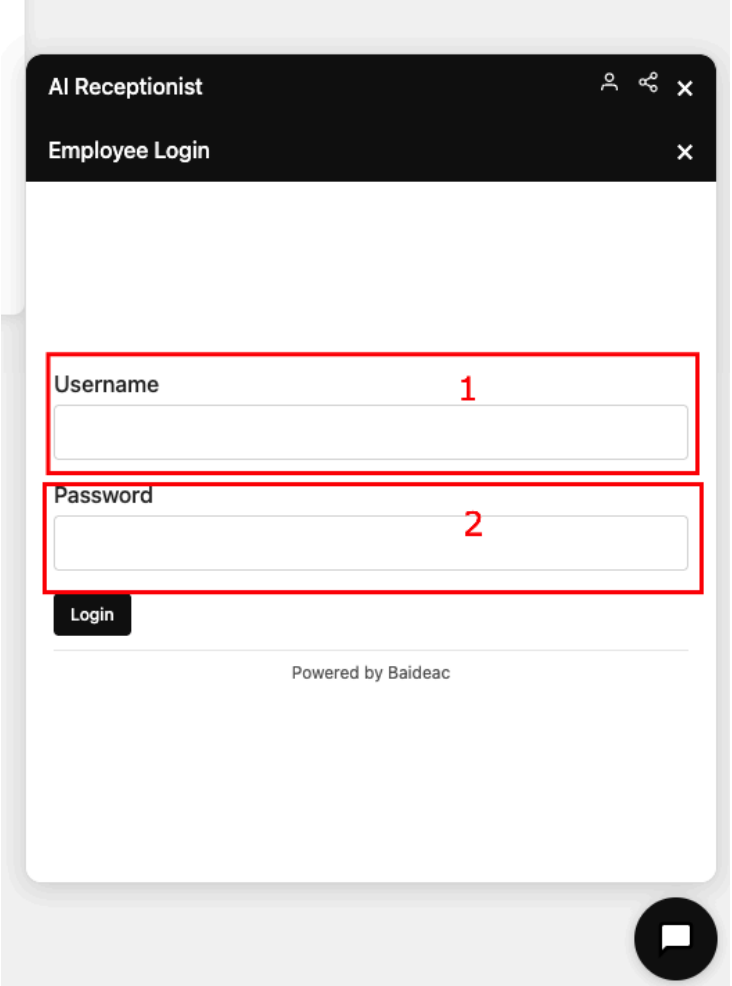
Here is how the BAR looks.

1. This is the initial step of chat widget



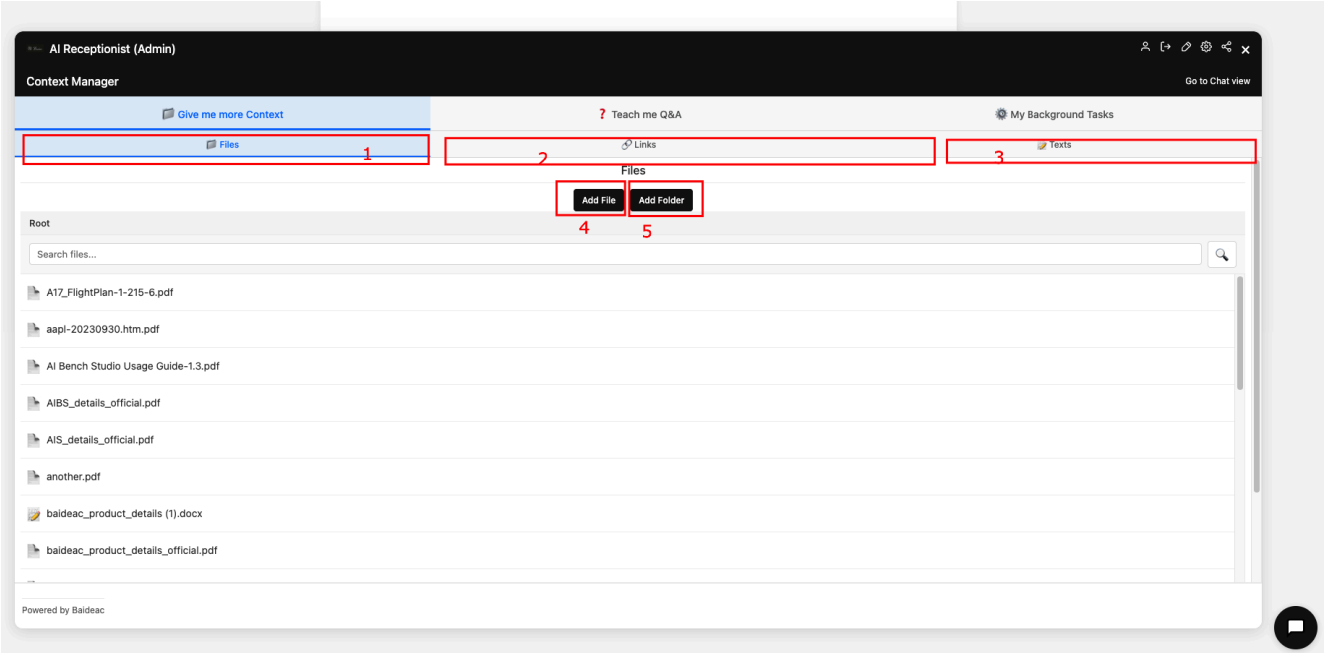
I. Employee/admin login
II. Close widget
III. Chat interface

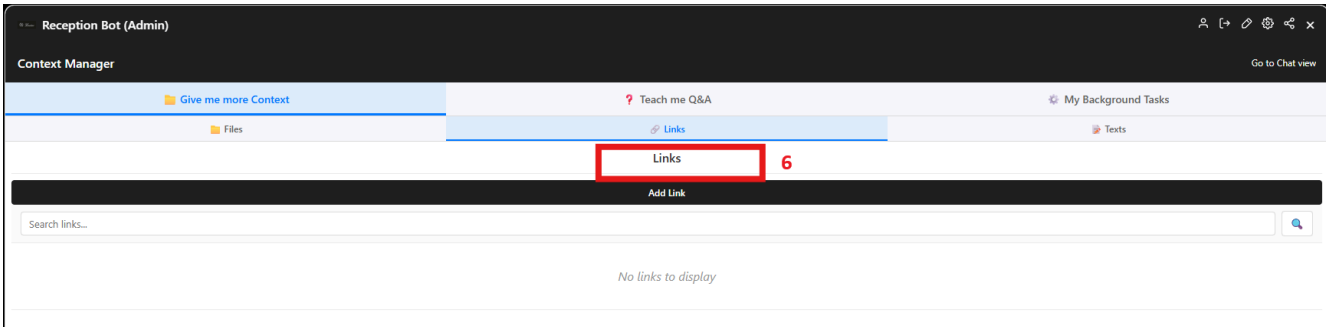
2. Employee/admin login

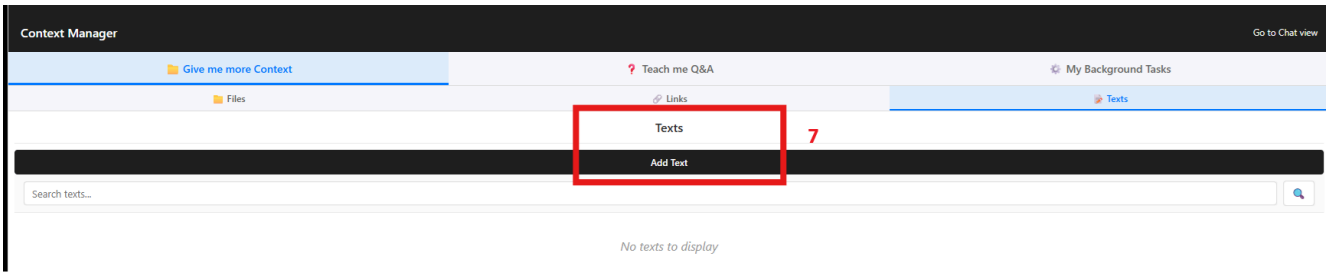


I. Enter username of employee or admin
II. Enter Password
III. Click on Login

3. Context model view



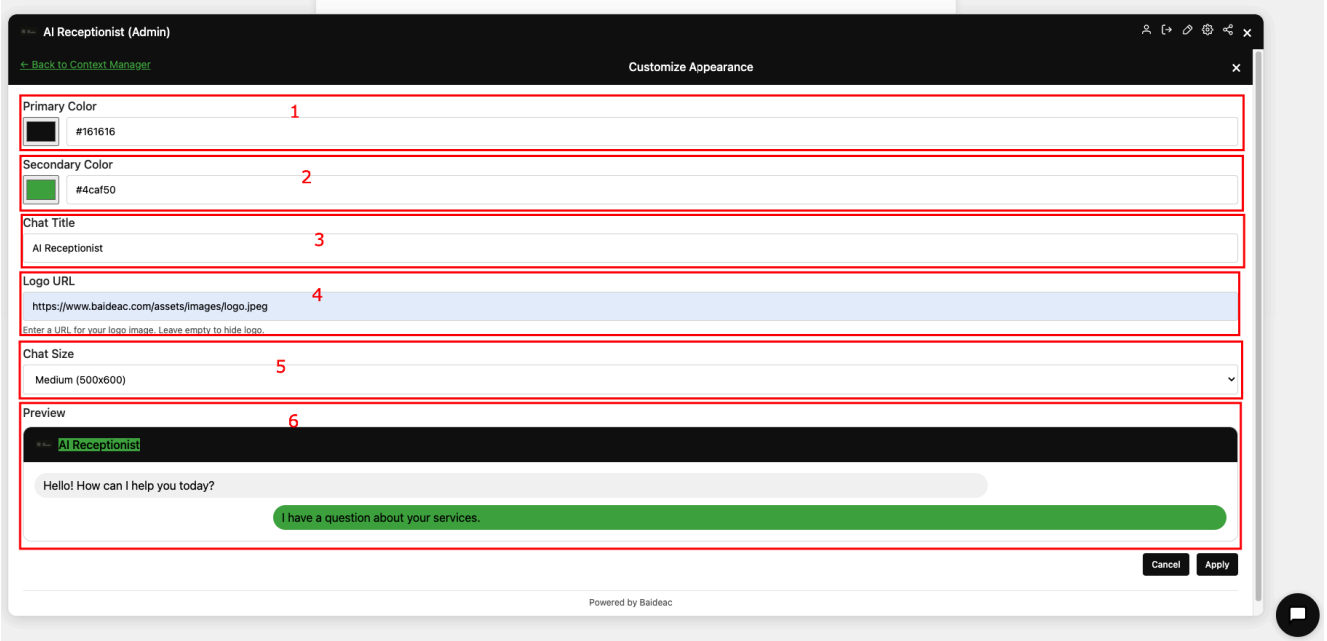




I. Files/Folder view
II. Links view
III. Text view
IV. “Add File” to add new files
V. “Add Folder” to add new folders
VI. “Add Links” to add new web access links

VII. “Add Text” to add text Paragraphs

4. Customize Appearance



I. Primary colour setup which modifies the overall background and theme
II. Secondary colour setup which used in chat interface
III. Chat Title indicate widget title
IV. Logo URL to show in chat title bar to user
V. Preview of theme.

Click on “Apply” button to apply theme

Installation setup with container

Supported Platforms

- Ubuntu

Pre-requisites

1. **License Key**
    - You can get your license to use the platform from https://www.baideac.com/licensing
    - Create account
    - Request Trial Key
    - Select “BAR”
    - Click on Request
    - In listed table, you can get license key
2. **Download assets**

- On same subscription portal
- At the record of BAR, we can see button called “Download”
- Click on it and get download script “baideac-par-installer.sh”
- Make it executable
    i. chmod +x baideac-par-installer.sh

3. **How to get ssl (**In the market, there are many ssl providers present. Here we are creating 1 from zerossl.com. Any user may choose any other provider as well. We must have the files as below)
    - Sign up at zerossl.com.
    - Create a new certificate, enter your domain.
    - Choose verification (DNS TXT record or HTTP file).
    - Complete verification, then download crt and key.
    - Install on your server and restart.
    - To get fullchain.pem you have to run this command
        - cat certificate.crt ca_bundle.crt > fullchain.pem
    - Create one directory called SSL and put this 4 files under that folder
        - ca_bundle.crt certificate.crt fullchain.pem private.key
4. **Installation process**
    - The installation includes the script as well as the docker image. Both are required. The script sets up the environment for the backend to run well, which is part of the docker container.
    - Once script is downloaded, simply run with sudo
        - sudo ./baideac-par-installer.sh
        - It will ask for two paths for the ssl certificate which needs to be set for nginx
            - certificate.crt
            - private.key
    - Once script is completed it will show this log as below giving the statuses about the nginx, ollama, docker, and postgres installations in your ubuntu system

5. **Setup the neo4j. (if enabled)**
    - We need to set up the neo4j DB**.** We can simply do this by executing the following command
        i. docker run -d \
        --name neo4j \
        -p 7474:7474 -p 7687:7687 \
        -e NEO4J_AUTH="neo4j/password" #This is the password you would like to set. For now, do not change this\
        -e NEO4J_server_default__listen__address=0.0.0.0 \
        -v neo4j_data:/data -v neo4j_logs:/logs \
        neo4j:5
        ii. The ports 7474 and 7687 needs to be enabled from the network perspective.

Direct Subscription (Pull image)

****Ignore this if you’re using Aws Marketplace subscription**

Step 1: Load Docker Image from tar

Run the following command where you run the script. First verify if tar file is presented there

sudo docker load -i par-image.tar

AWS Marketplace Subscription (Pull image)

****Ignore this if you're using Direct subscription**

Before using AWS ECR, you need to configure AWS CLI with your credentials. Run the following command to configure AWS CLI:
aws configure

1. Provide the following information when prompted:
    - **AWS Access Key ID**: Your AWS access key.
    - **AWS Secret Access Key**: Your AWS secret key.
    - **Default region**: us-east-1 (or your desired region).
    - **Default output format**: json (or your preferred format).

Step 1: Authenticate Docker with AWS ECR

Run the following command to authenticate Docker to an AWS ECR registry:

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin \
709825985650.dkr.ecr.us-east-1.amazonaws.com
```

Step 2: Pull the Docker Images

Run the following loop to pull each image:

```
docker pull \
709825985650.dkr.ecr.us-east-1.amazonaws.com/bhojr/bhojr-prod/par:1.1
```

Run the dockerized setup

Step 1: Prepare content .env file

Here are just the fields which needs to be changed. Keep the rest as is. A few can be controlled as shown above.

```
########## START of .env #################
NEO4J_PASS=password #same password given to setup neo4j only if enabled
LICENSE_KEY=<> # your license key from Baideac.

# RBAC Configuration
RBAC_BASE_URL=http://<YOUR_RBAC_IP>:55502
RBAC_TIMEOUT=30

########## END of .env ###################
```

Enable RBAC (only if the user needs the rbac for admin user credentials)

- For enabling the RBAC you can refer this documentation
- Ensure that the RBAC backend port is enabled in the firewall.
- To enable RBAC, set AUTH_MODE=rbac in the .env file mentioned above. When setting AUTH_MODE=rbac, be sure to specify the RBAC_BASE_URL as the URL for the RBAC backend API.
- Currently, BAR does not use any role-based access system, but it may be implemented in future releases. Therefore, do not be concerned with the RBAC role—it will not impose any restrictions on the user created at this time.

Step 2: Run the Docker Images

Run the following loop to pull each image:

```
docker run -d  -v $(pwd)/context:/app/context \
 -e LICENSE_KEY="<License Key Value>" \
-v $(pwd)/SSL:/app/SSL -v $(pwd)/config.json:/app/config.json \
-v $(pwd)/contexts.json:/app/contexts.json -v $(pwd)/.env:/app/.env \
--name ai-receptionist   --add-host=host.docker.internal:host-gateway \
-p 5011:5011 \
709825985650.dkr.ecr.us-east-1.amazonaws.com/bhojr/bhojr-prod/par:1.1
```

Step 3: Check Logs

```
docker logs -f ai-receptionist
```

Once done, the BAR setup is now ready to be embedded to any site.

Troubleshooting

- **Authentication Errors**: Ensure your AWS credentials are correctly configured and have the necessary permissions to access the ECR registry.
- **Docker Not Found**: Verify Docker is installed and running.
- **Use of sudo:** Use sudo command in all above steps if your docker is not accessible without sudo permission
- **Port Enable:** You have to enable ports in firewall 80, 443, 5011
- **EC2 type:** To get better utilisation and get correct response we are recommending to use t2.2xlarge or any ubuntu machine with ram of equal or more than 32 GB.
- **IP update:** In case you got new public IP address in that case you must have to update nginx block that is present at /etc/nginx/conf.d/custom-proxy.conf
- **Region Mismatch**: Make sure the specified region (us-east-1) matches the region of your ECR repository.

How to embed in your website

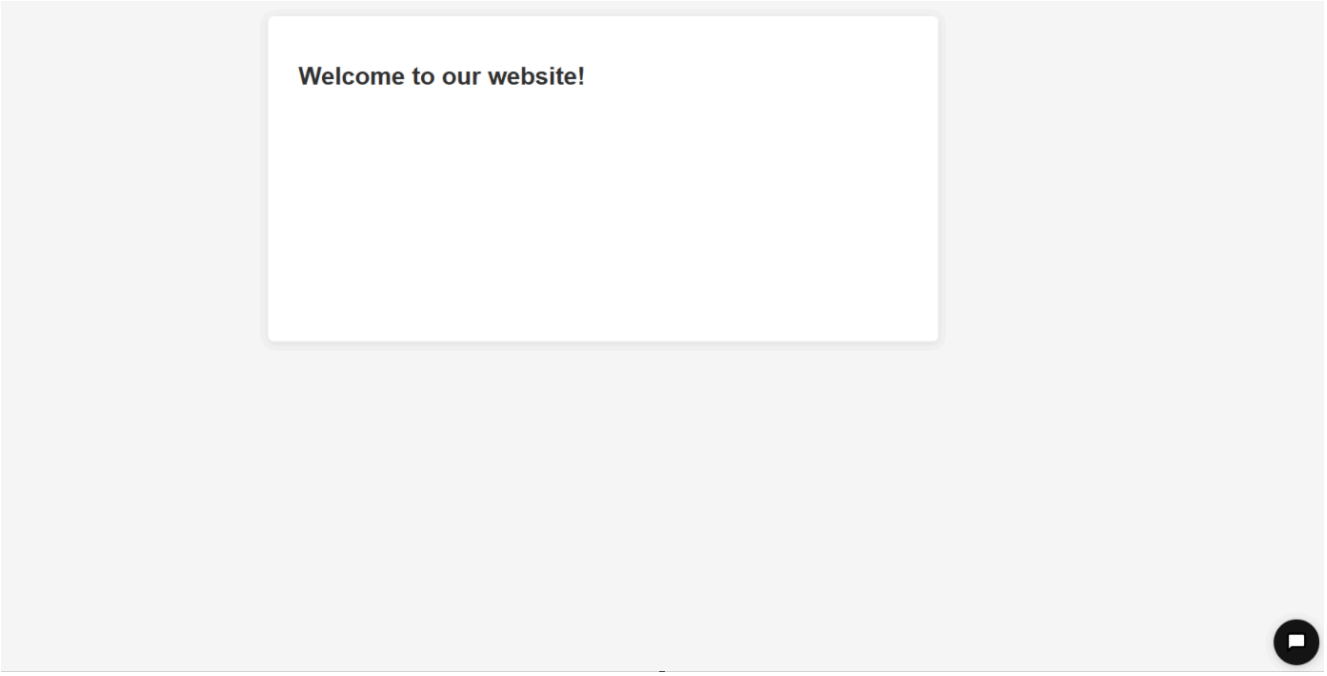
- Site Implementation Instructions:
    - To implement the site, you only need a single <script> tag and a <button> element with a custom ID, named "customer-chat-trigger".

- In the <script> tag, make sure to specify your BAR hosted IP address:
    - Use the **parurl** with port 5011 – This is used for the admin sections. It would help in setting the theme and updating documents for the BAR to refer

An example is given below

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Test AI Chat Widget</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 40px;
            background-color: #f5f5f5;
        }
        .content {
            max-width: 800px;
            margin: 0 auto;
            background: white;
            padding: 40px;
            border-radius: 8px;
            box-shadow: 0 2px 10px rgba(0,0,0,0.1);
        }
        h1 {
            color: #333;
            margin-bottom: 20px;
        }
        p {
            color: #666;
            line-height: 1.6;
            margin-bottom: 20px;
        }
    </style>
</head>
<body>
    <div class="content">…
    </div>

    <!-- Include the chat widget script -->
    <script id="par-script"
            data-parurl="34.200.40.108:5011"
            src="https://bhojr-mp.s3.us-west-2.amazonaws.com/script-bot.js">
    </script>
</body>
</html>
```



And voila!, the BAR can be easily set up and integrated!

For more queries, please drop us an email at support@baideac.com