



GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT & ENTREPRENEURSHIP
DIRECTORATE GENERAL OF TRAINING
NATIONAL SKILL TRAINING INSTITUTE

NSTI (W) SALLAKE KOLKATA

CERTIFICATE

This is to certify that following trainees have completed their project titled

“NSTI Smart Cloud Campus”

For IBM Program – IT, Networking and Cloud (Technical Diploma)

ROLL NO	NAME
ADIT19AU03683	PUSHPA KUMARI

Miss.Arputa Roy
IBM Faculty

Mr. K.L Kuli
ADIT Director

Mr. Ramamurthy
Head of office/Principal

Mr. Sarbojit Neogi
Section In-charge

ACKNOWLEDGEMENT

We would like to express my sincere gratitude to several individuals and organizations for supporting me throughout my diploma study. First, I wish to express my sincere gratitude to my trainer, Mr. Boddu Lingaiah and Miss Arpita Roy, for Their enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my study and writing of this project report. His immense knowledge, profound experience and professional expertise in computer science has enabled me to complete this project successfully.

I also wish to express my sincere thanks to the National Skill Training Institute, Salt lake Kolkata for accepting me into the diploma program. In addition, I am deeply indebted to the Ministry of Skill Development & Entrepreneurship and IBM for granting me the diploma course. This technical and financial support has enabled me to complete my diploma course studies successfully. Also, I am grateful the Mr. Boddu Lingaiah faculty of NSTI(W) Salt Lake Kolkata for supporting me for course completion in the specific subject.

I am also grateful to the following NSTI Regional Director Mr. K.L Kuli and Principal of NSTI(W)Salt Lake Kolkata -G.C Ramamurthy for their consistent support.

ABSTRACT

The system aims at the maintenance and management of the different Hotels that are available in the different parts of the world. It mainly takes care of the Hotel management at the core area of the database. The system provides the information regarding the different Hotels that are available and their status specific to availability. The guests can visit the site and give the feedback themselves with the required information that is expected by the system. Each registered guest can raise a request for the unit bookings. The Guests are scheduled with the information of the availability of the units for they have requested the time

The total front end was dominated using HTML, CSS and JAVASCRIPT standards applied with the dynamism of NodeJS server pages. The communicating client was designed using AI, Watson Assistant learns from customer conversations, improving its ability to resolve issues the first time while removing the frustration of long wait times, tedious searches and unhelpful chatbot at all proper levels high care was taken to check that the system manages the date consistency with proper business validations. The database connectivity was planned using the MongoDB Database Connectivity, the authorization and authorization was cross checked at all stages. The user level accessibility has been restricted into two zones the administrative and the normal user zone.

INDEX

Sr. No.	Table of Contents	Page No.
1	Chapter 1: Introduction	4-5
2	Chapter 2: Services and Tools Required	6-16
2.1	Used services	6
2.2	Build a chatbot using IBM Watson Assistant	6-13
3	Chapter 3: Project Architecture	17
4	Chapter 4: Architecture Blocks Detail Working	18
5	Chapter 5: create a webpage using node and add data with mongo DB Atlas	
5.1. a	Creating files	19
5.1. b	Install <i>dependencies</i> modules for <i>NodeJS</i> server	20
5.2	<i>Connect our data with mongo dB Atlas</i>	21-35
5.2. a	Create an account in mongo dB	22-24
5.2. b	<i>Account setup</i>	24-25
5.2.c	<i>Create cluster</i>	25-26
5.2. d	<i>Add data with mongo dB database</i>	27-35
6	Source Code	36-71
7	Screenshot	70-75
8	Conclusion	76
9	References	77

CHAPTER 1

INTRODUCTION

- 1. Overview**
- 2. Feature**
- 3. Scope**
- 4. Future Work**

INTRODUCTION

1. Overview

User can easily enter this website and know all about fabulous hotel.

And Watson assistant assist the user's what food available here or not available and anything according to users need and ask the questions Watson assistant will give those answers
user can be easily giving his feedback etc...

2. Feature

- User easily Althing know about hotel and hotel's special dishes and food menu
- In food menu user can be known about what is available in breakfast, dinner, main dishes, lunch and deserts dishes and what is the cost of food.
- our Watson assistant will assist the users
- user can be contact and also know about the hotel opening and closing timing.

3. Scope

This website specially design for saves the time of users and hotel's staff and worker

Half work of staff does this website and chatbot.

4. Future Work

The purpose of the hotel website is to save the users and hotel staff time and as a hotel boss no need to more workers he does his work via few workers. A clear understanding of the hotel website and its functionality will allow for the correct software to be developed for the end-user and will be used for the development of the future stages of the project. On our website, there is an option to book a table online but our future plan is to order food online also because we want to provide full customer satisfaction and we are working on that.

CHAPTER 2

SERVICES AND TOOLS REQUIRED

2.1. Services Used

a) IBM Watson Assistant

Watson Assistant is chatbot that provides customers with fast, consistent and accurate answers across any application, device or channel. Using AI, Watson Assistant learns from customer conversations, improving its ability to resolve issues the first time while removing the frustration of long wait times, tedious searches and unhelpful chatbots.

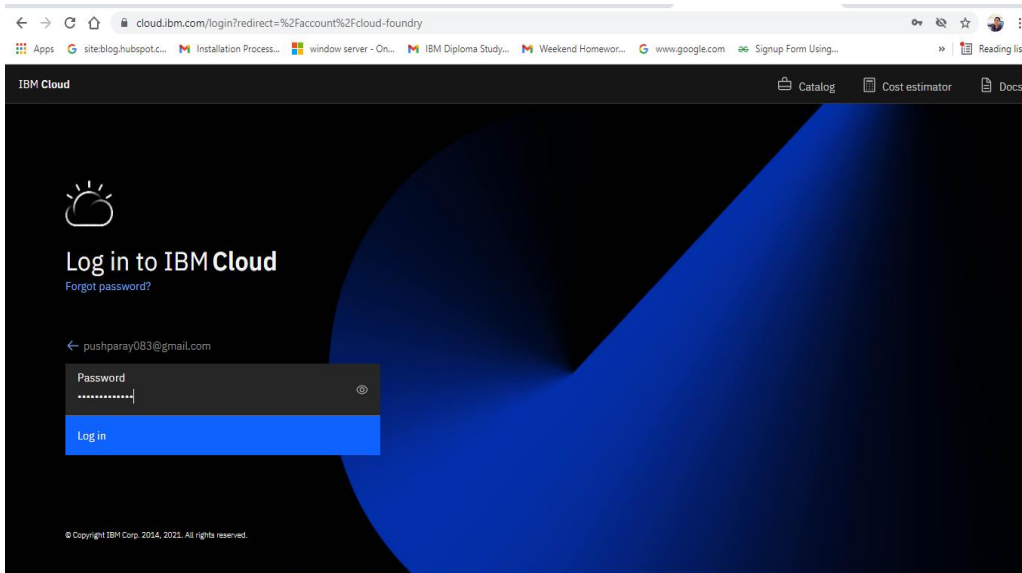
b) What is Actions Method in IBM Watson Assistant?

A new feature has been added to IBM Watson Assistant called Actions. This new feature allows users to develop dialogs in a rapid fashion.

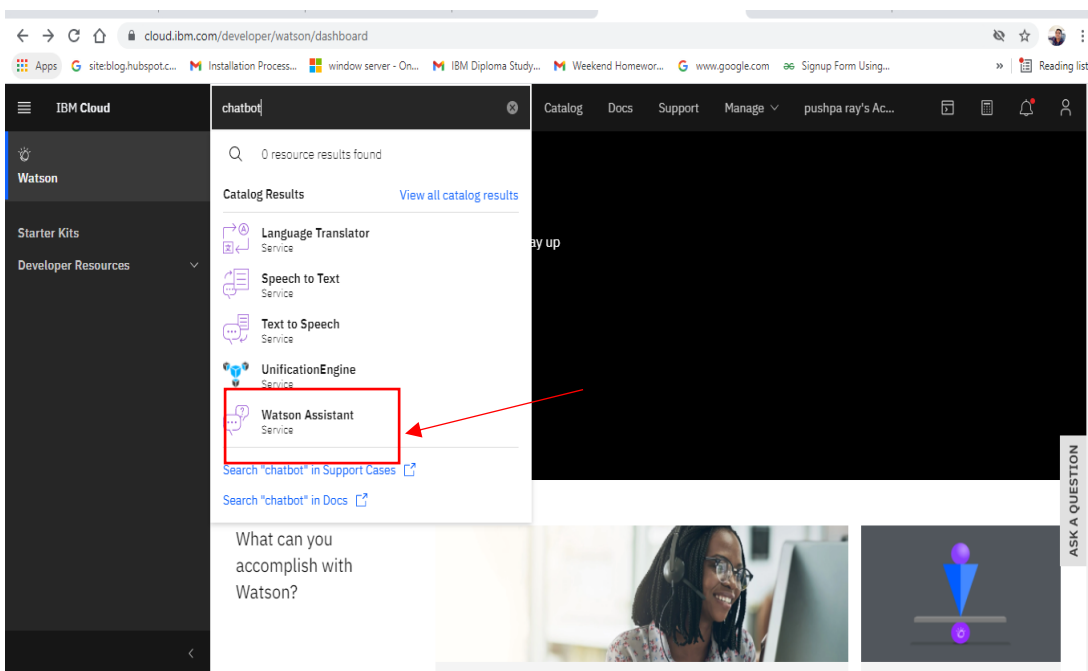
c) Build a chatbot using actions

Step 1

- Click on this link
<https://cloud.ibm.com/login?redirect=%2Faccount%2Fcloud-foundry>
- Login to IBM cloud
- Give your IBM cloud credential



- You can see open your dashboard
- Search on search box chatbot
- Select Watson Assistant



- We go this page.
- Prove the certain name of application.
- Click on create.

The screenshot shows the IBM Cloud Watson Assistant creation interface. The browser address bar displays `cloud.ibm.com/catalog/services/watson-assistant`. The page header includes the IBM Cloud logo, a search bar, and navigation links for Catalog, Docs, Support, and Manage. The user's account name, 'pushpa ray's Ac...', is visible in the top right. The main content area is titled 'Watson Assistant' and includes a 'Create' button. Below this, there are sections for 'Select a location' (with a dropdown menu showing 'London (eu-gb)') and 'Select a pricing plan'. A table lists the available plans:

Plan	Features	Pricing
Lite	Everything you need to get started, free for as long as you need it Up to 1,000 unique monthly active users (MAUs) chatting with your assistant Up to 10,000 messages per month	Free

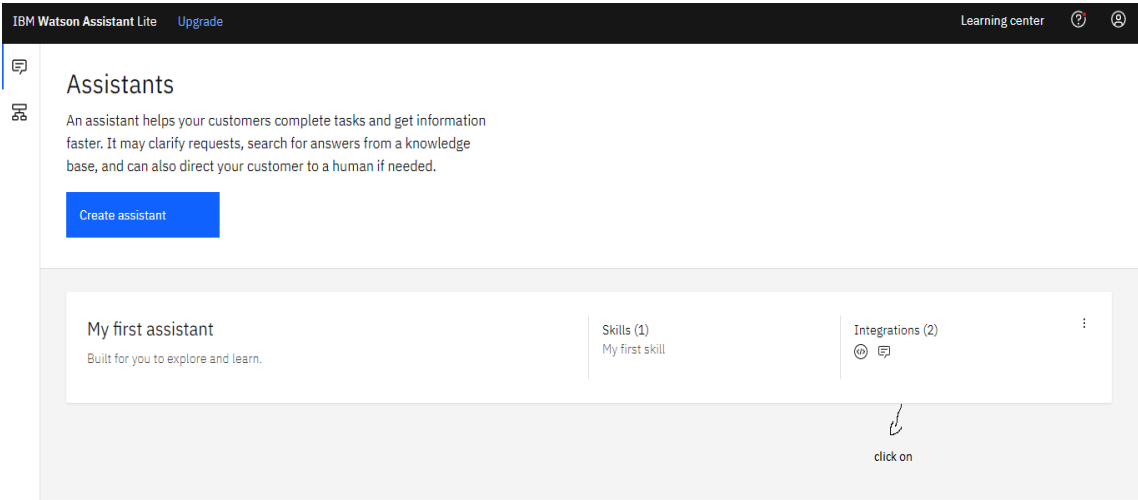
On the right side, a 'Summary' panel displays the configuration: Watson Assistant, Free, Location: London, Plan: Lite, Service name: Watson Assistant-5y, and Resource group: Default. At the bottom of this panel are buttons for 'Create', 'Add to estimate', and 'View terms'.

- You can see created waston assistant
- Click on launch waston assistant

The screenshot shows the IBM Cloud Watson Assistant-5y resource page. The browser address bar displays `cloud.ibm.com/services/conversation/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aeu-gb%3Aa%2F523486e2e8a34b4680ab99c758a62611%3Ae7b02d08-153f...`. The page header includes the IBM Cloud logo, a search bar, and navigation links for Catalog, Docs, Support, and Manage. The user's account name, 'pushpa ray's Ac...', is visible in the top right. The main content area is titled 'Watson Assistant-5y' and includes a 'Launch Watson Assistant' button. Below this, there are sections for 'Start by launching the tool' (with links for 'Getting started tutorial' and 'API reference') and 'Credentials' (with fields for 'API key' and 'URL'). A red arrow points from the 'Launch Watson Assistant' button to the 'API key' field.

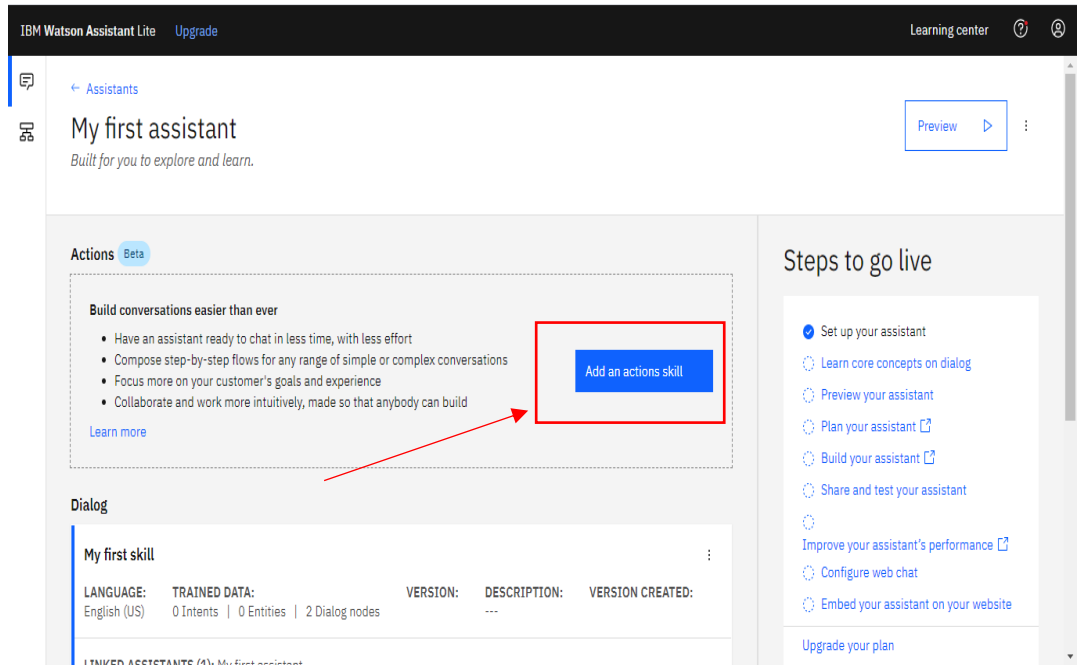
Step 2

Click on integration

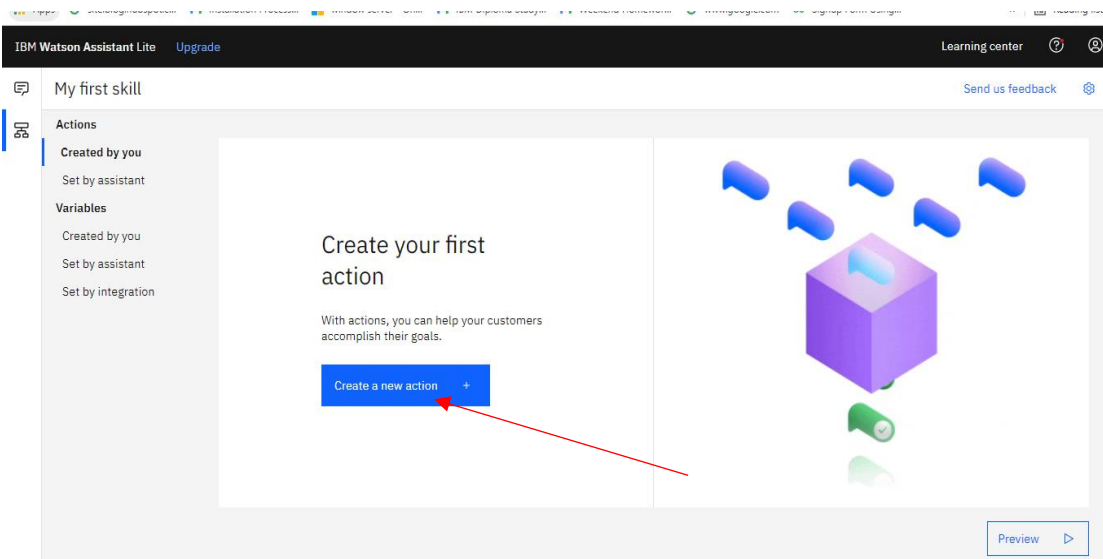


After that click on

That indicated to image

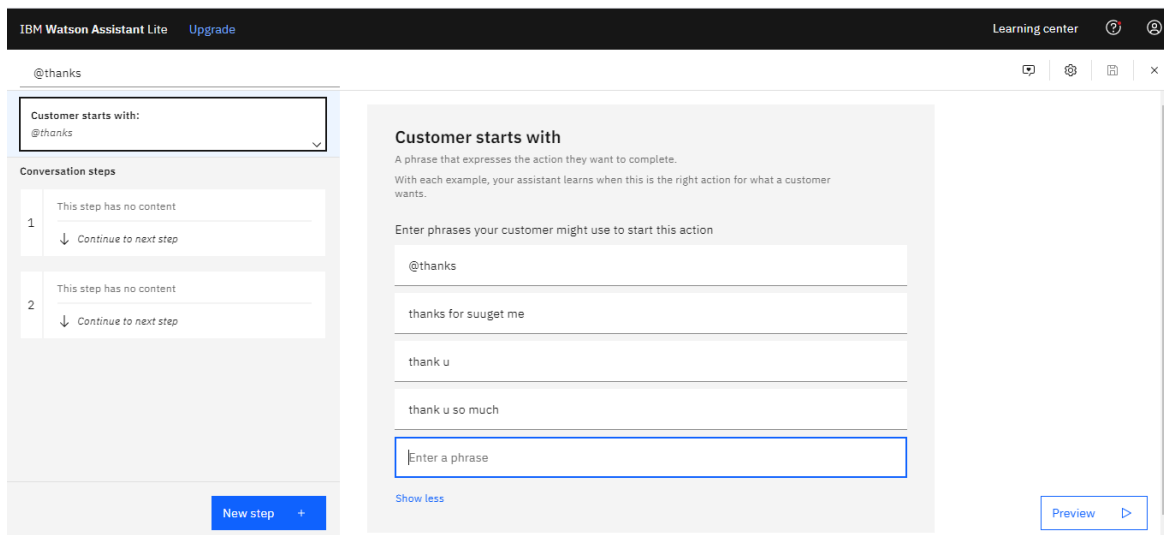


Click on create a new action



We get this page

- Give customer questions
- click on new step
- Provide the assistant response
- Following this step create a chatbot waston assistant using actions



There are actions created by me

IBM Watson Assistant Lite Upgrade

Learning center ?

waston assistant Beta

Send us feedback

Actions

Created by you

Provided by assistant

Variables

Created by you

Actions: created by you

The goals your customer wants to accomplish and your assistant is trained to help with. Examples: paying a bill, updating an account, or answering a question. [Learn more](#)

New action +

Name	Last edited	Disambiguation	Status	
@special	4 days ago	On	✓	⋮
hi	4 days ago	On	✓	⋮
breakfast for vegetarian	4 days ago	On	✓	⋮
@thanks	a few seconds ago	On	✓	⋮

Items per page: 50 Showing 1-7 of 7 actions

1 1 of 1 pages

Preview ▶

Step 3

A. How to check chatbot properly run or not:-

Click on preview

IBM Watson Assistant Lite Upgrade

Learning center ?

waston assistant Beta

Send us feedback

Actions

Created by you

Provided by assistant

Variables

Created by you

Actions: created by you

The goals your customer wants to accomplish and your assistant is trained to help with. Examples: paying a bill, updating an account, or answering a question. [Learn more](#)

New action +

Name	Last edited	Disambiguation	Status	
@special	4 days ago	On	✓	⋮
hi	4 days ago	On	✓	⋮
breakfast for vegetarian	4 days ago	On	✓	⋮
@thanks	a few seconds ago	On	✓	⋮

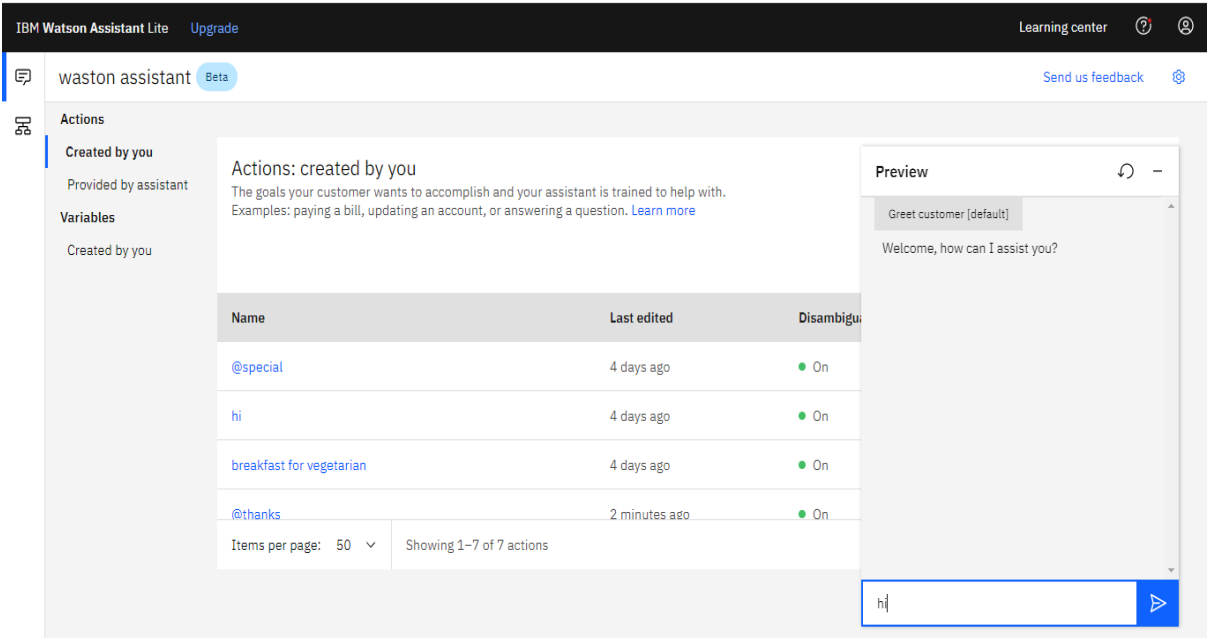
Items per page: 50 Showing 1-7 of 7 actions

1 1 of 1 pages

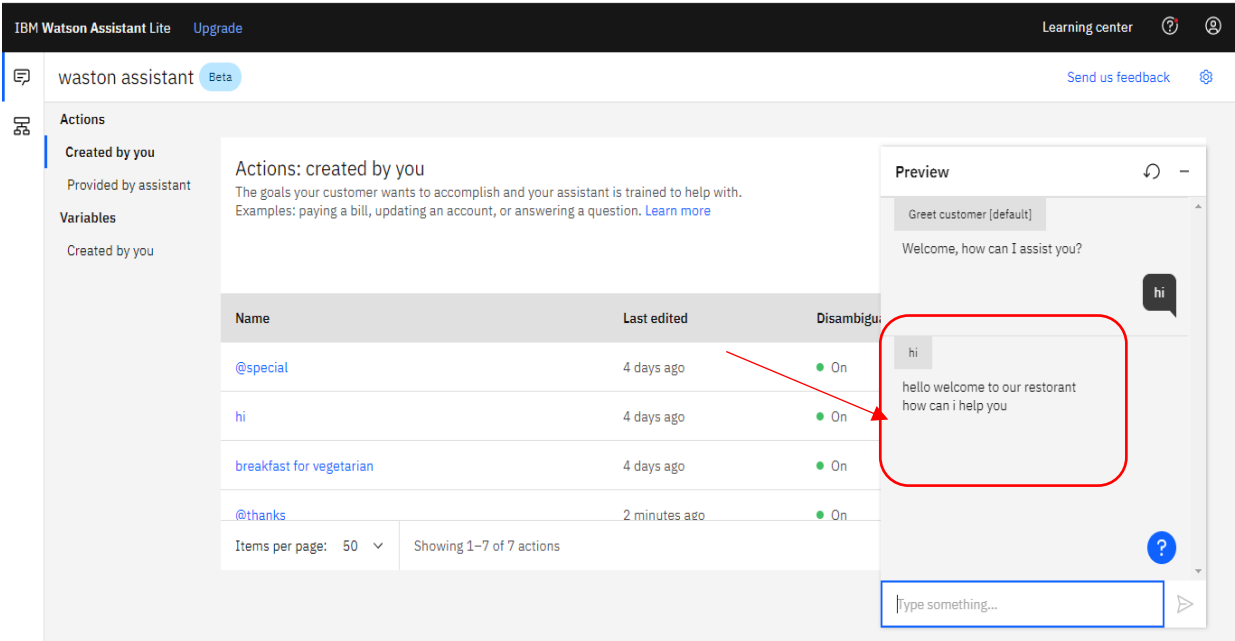
Preview ▶

Write the questions according to the user-

I have provided “hi” in the chatbot



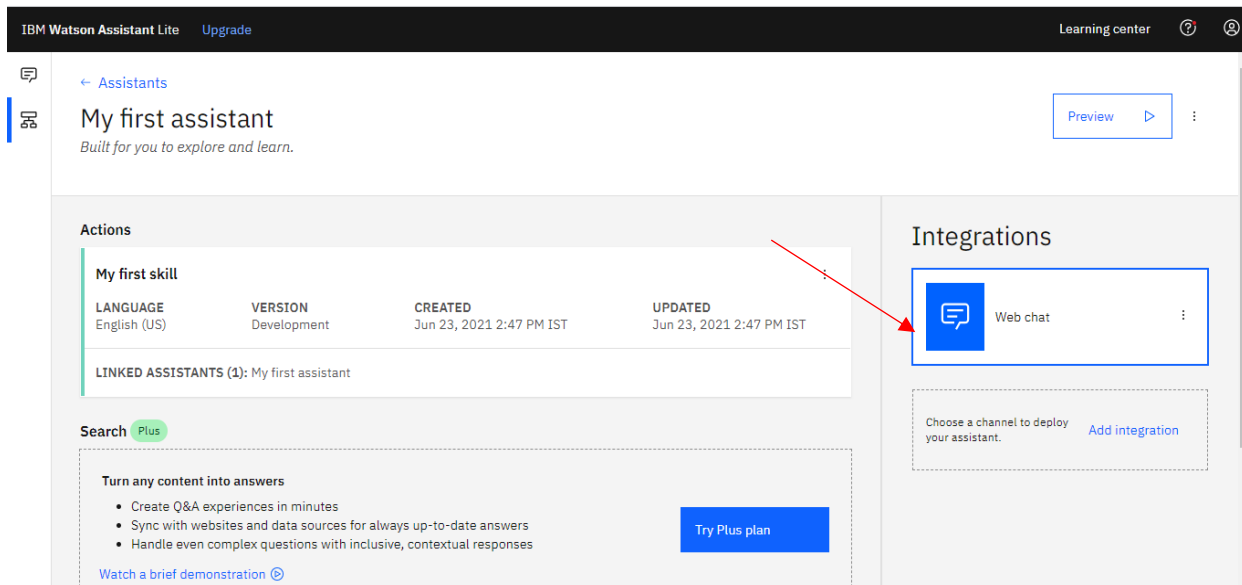
Now, you can see in image chatbot give the user response, that means our chatbot successfully created



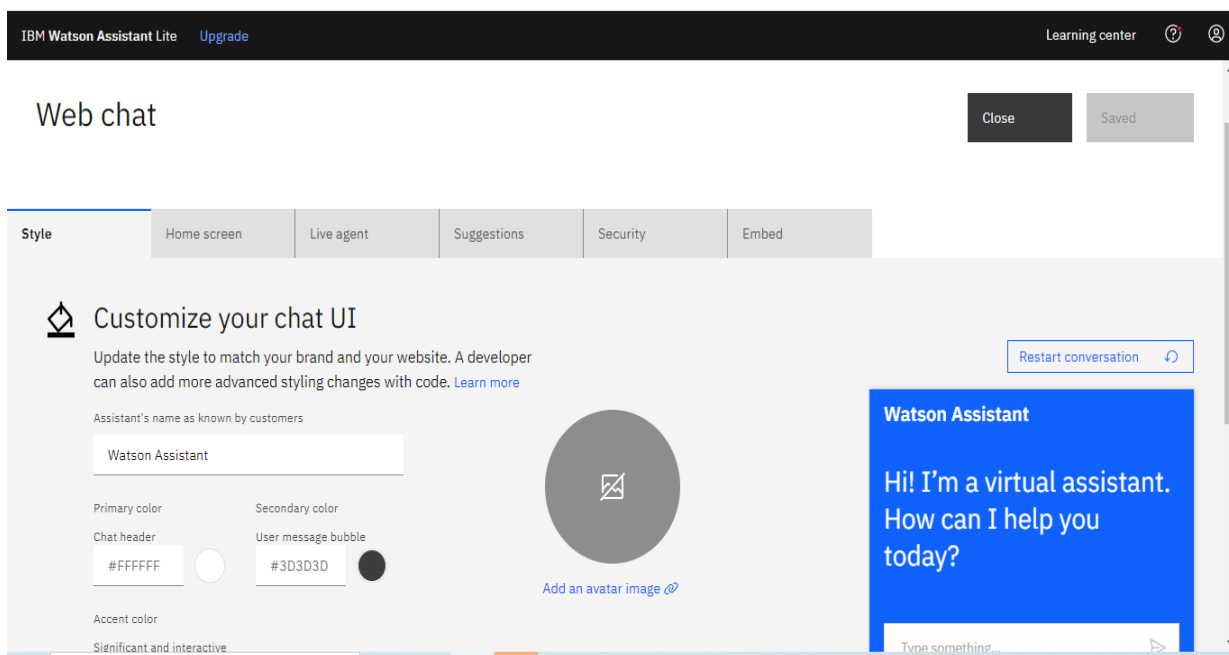
Step 4

How to add IBM watson assistant with a HTML webpage:-

Go to web chat



- Click on embedded
- Copy the link
- Paste it your html page



2.2 Tools and Software used

Used Software

- Node.js
- MongoDB Compass
- VS Code Editor

Used Hardware

Processor: - Intel Core i3 & Higher Version

Operating System: - Windows 10

Memory: - :- 4GB RAM

Hard Disk: - :- 1TB Hard Disk

Technology Used

- HTML
- CSS
- JAVASCRIPT
- EXPRESS.JS
- EJS

Run time Environment

Node.js

Database

Mongo dB Atlas

Technology Used

I. HTML

The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

II. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.

CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file which reduces complexity and repetition in the structural content as well as enabling the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

III. JavaScript

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g., functional programming) styles. Read more about JavaScript.

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about API specifics to Web pages, please see Web APIs and DOM.

IV. Express.js

Express.js, or simply Express, is a back-end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.[3] It has been called the de facto standard server framework for Node.js.

V. EJS

Ejs stands for Embedded JavaScript templates. It can be used as an alternative template engine.

In this article, you will learn how to apply EJS to an Express application, include repeatable parts of your site, and pass data to the views.

VI. NodeJS

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. In the following "hello world" example, many connections can be handled concurrently. Upon each connection, the call back is fired, but if there is no work to be done, Node.js will sleep.

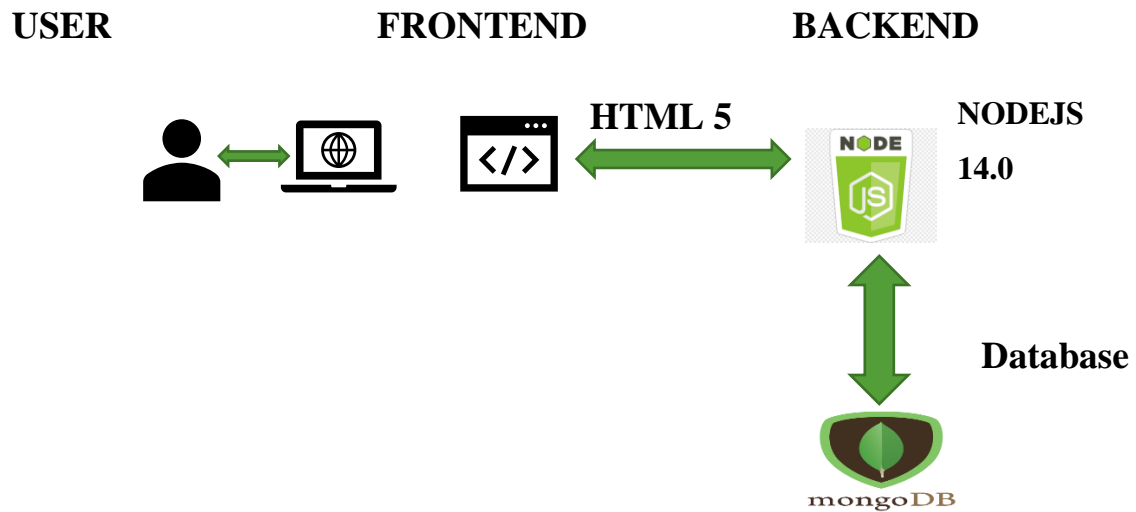
VII. Mongo dB Atlas

MongoDB Atlas is a fully-managed cloud database developed by the same people that build MongoDB. Atlas handles all the complexity of deploying, managing, and healing your deployments on the cloud service provider of your choice (AWS, Azure, and GCP).

CHAPTER 3

PROJECT ARCHITECTURE

3.1 Architecture

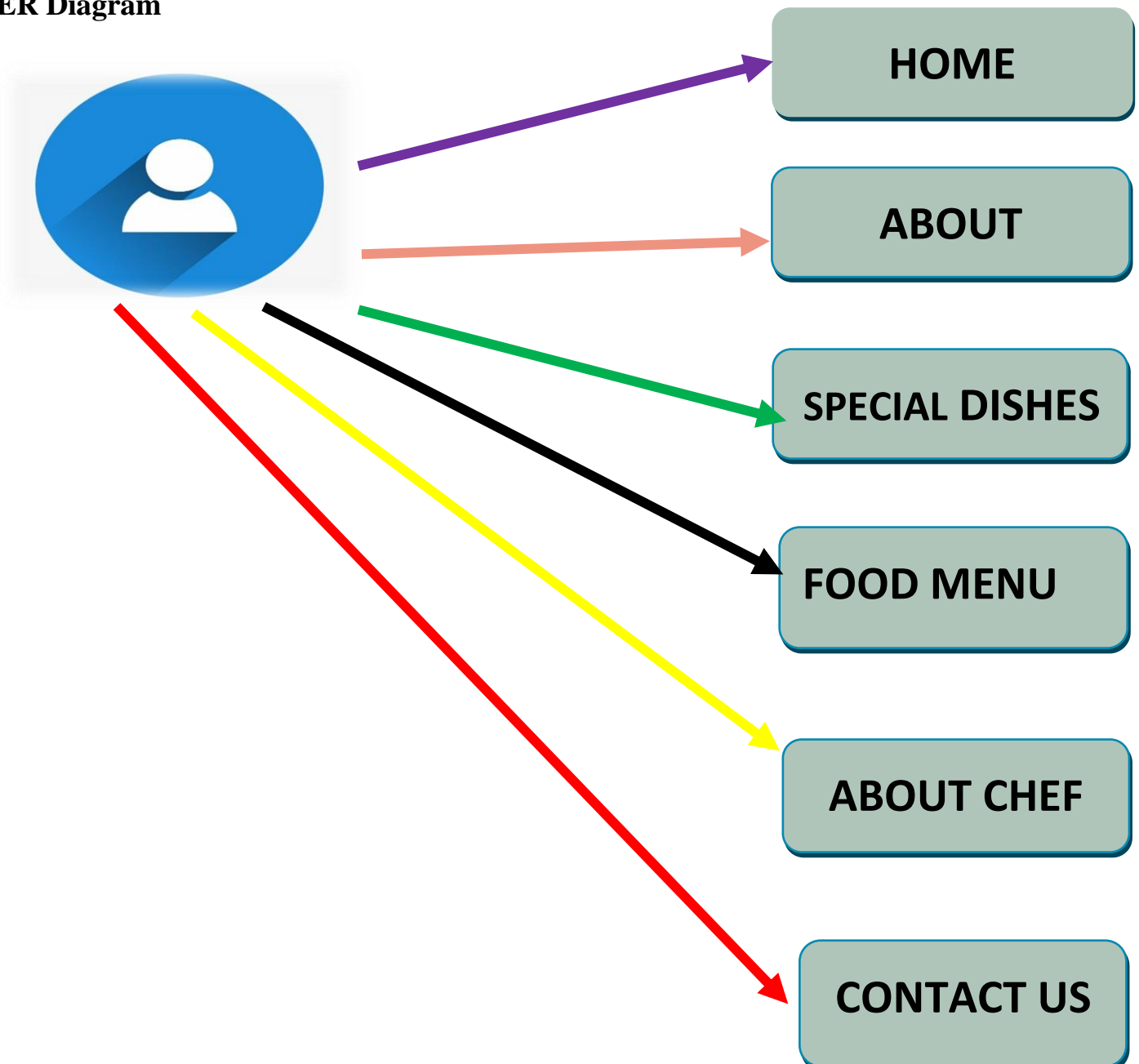


CHAPTER 4

ARCHITECTURE BLOCKS DETAIL WORKING

4.1 Blocks

ER Diagram



CHAPTER -5

Develop an application using Node.js and add data with mongo DB Atlas database

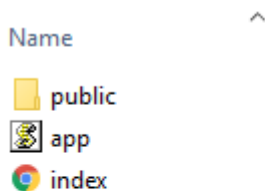
a) Creating files/project file Structure

Create folder

Inside project folder I have created those 3 files & folder

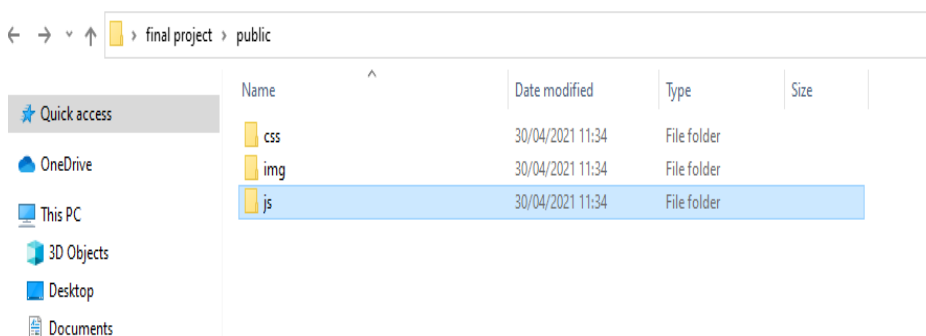
1. index.html
2. Public
3. App.js

You can see the images of those folders



Public folder

You can see the image



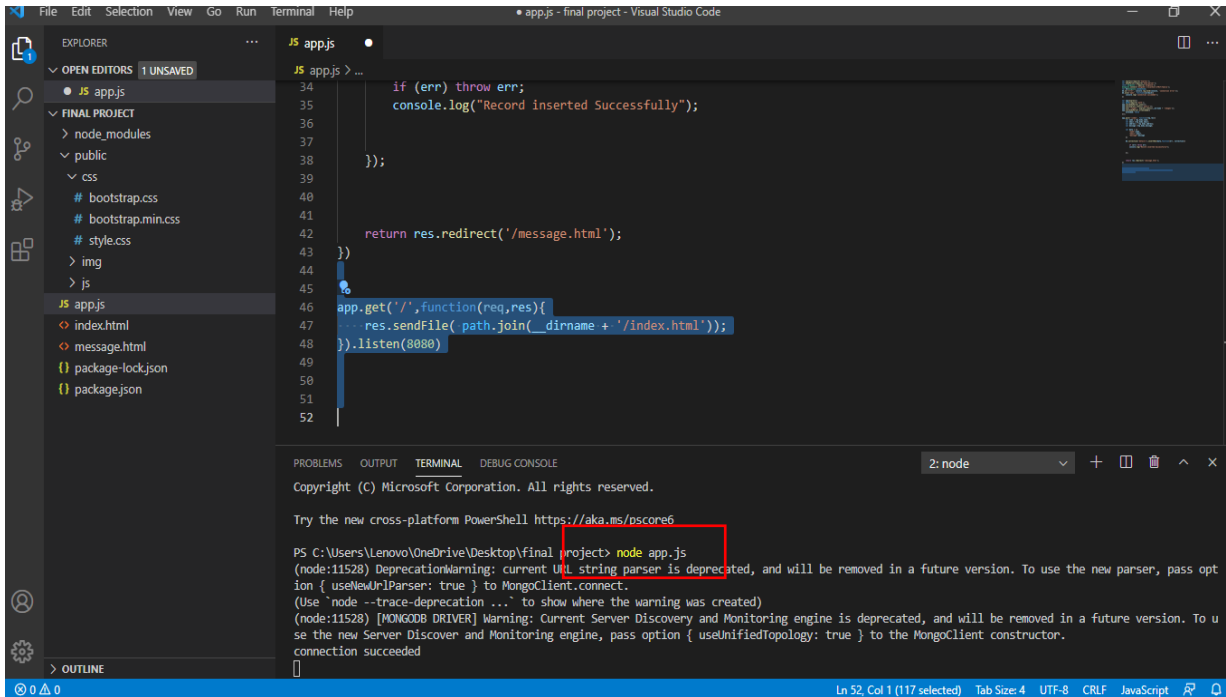
b) Install dependencies modules

1. Install express module using command `npm install express` – save
2. Install body-parser using command `npm install body-parser`

These are modules required to do this activity

After that start the application on node.js server

Write down on terminal `node app.js`



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the project structure with files like `app.js`, `index.html`, `message.html`, `package-lock.json`, and `package.json`. The main editor displays the `app.js` file with the following code:

```

34     if (err) throw err;
35     console.log("Record inserted Successfully");
36
37   });
38
39
40
41
42   return res.redirect('/message.html');
43 }
44
45
46 app.get('/', function(req, res){
47   res.sendFile(path.join(__dirname + '/index.html'));
48 }).listen(8080)
49
50
51
52

```

The terminal window at the bottom shows the command `node app.js` being executed. The output includes a deprecation warning about the URL string parser and a message from the MONGODB DRIVER. The terminal text is as follows:

```

PS C:\Users\Lenovo\OneDrive\Desktop\final project> node app.js
(node:11528) DeprecationWarning: current URL string parser is deprecated, and will be removed in a future version. To use the new parser, pass option { useNewUrlParser: true } to MongoClient.connect.
(node:11528) [MONGODB DRIVER] Warning: Current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
connection succeeded

```

A. Run application NodeJS server

go to browser and type `localhost:8080`



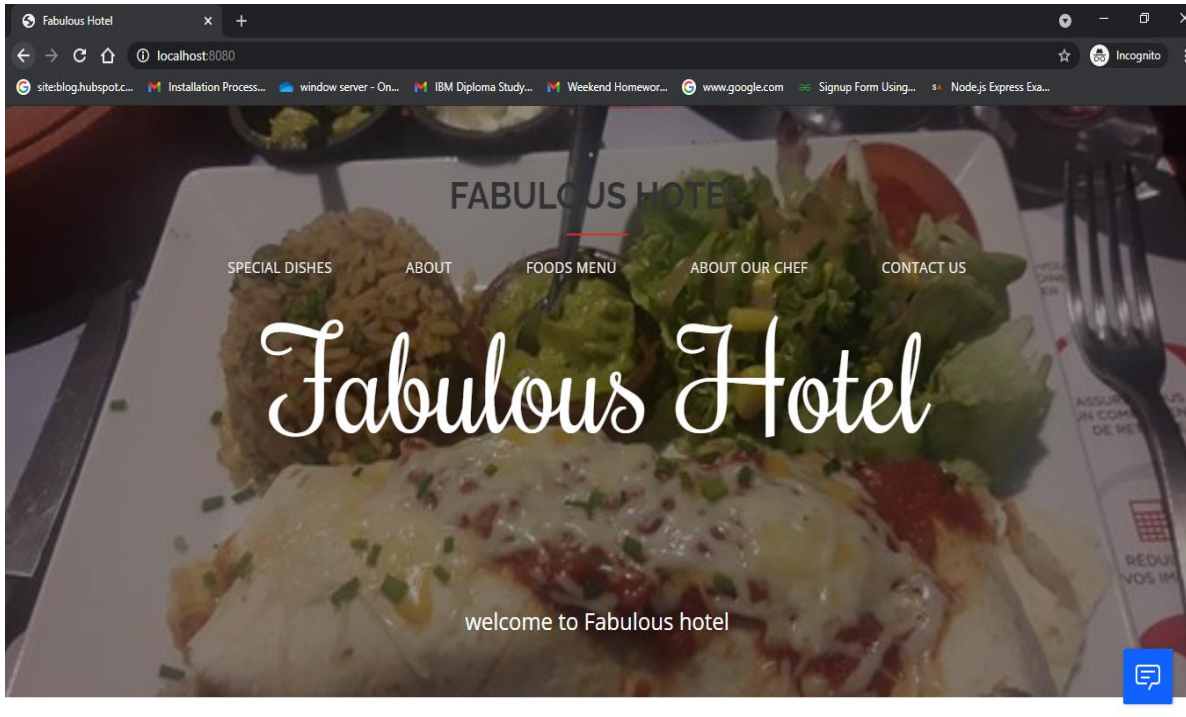
deploy the static webpage locally see the picture



This is the main page



You can my web page running locally on 8080 port No



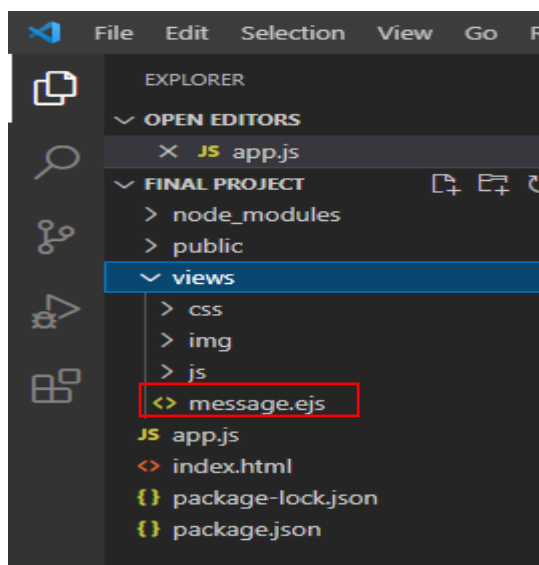
connect our data with mongo DB Atlas

1. install dependencies modules

- Install mongoose using command `npm install mongoose`
- Install ejs using command `npm install ejs`

For store user data we have to need created a views folder

- ✓ Create an ejs file
- ✓ I have created a message.ejs file
- ✓ here You can see the image



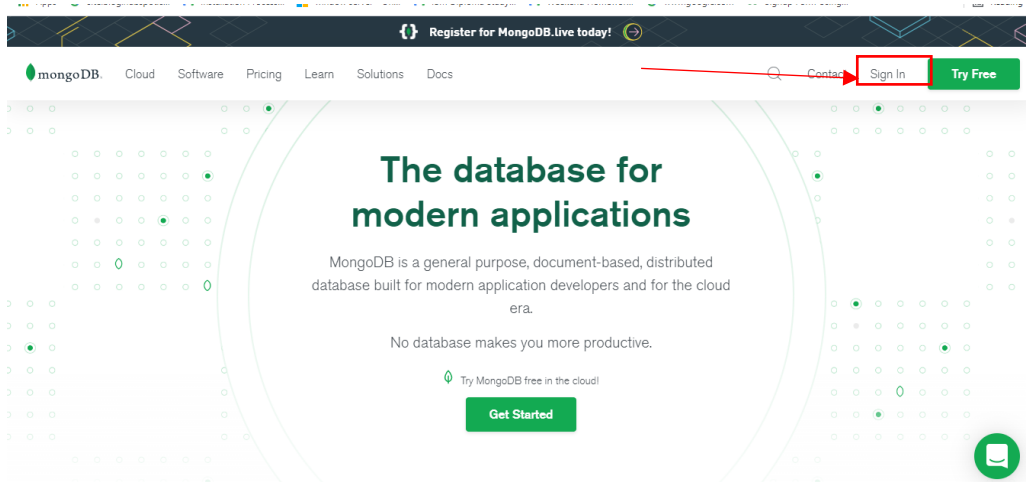
2. create an account in mongo dB Atlas

step 1

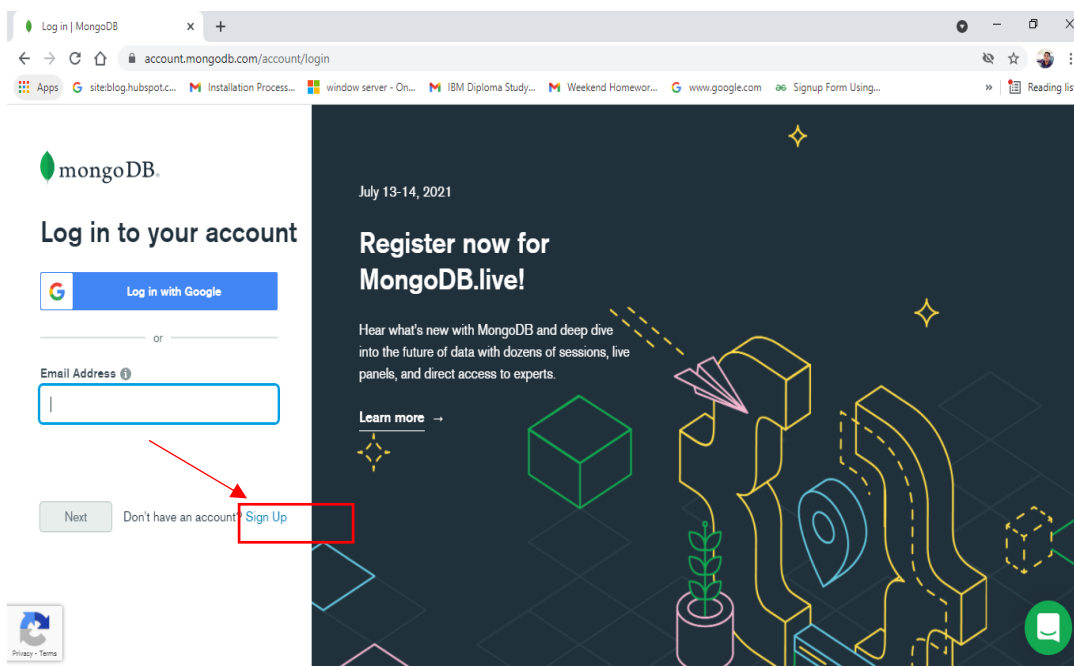
go this link

<https://www.mongodb.com/>

click on sign in button

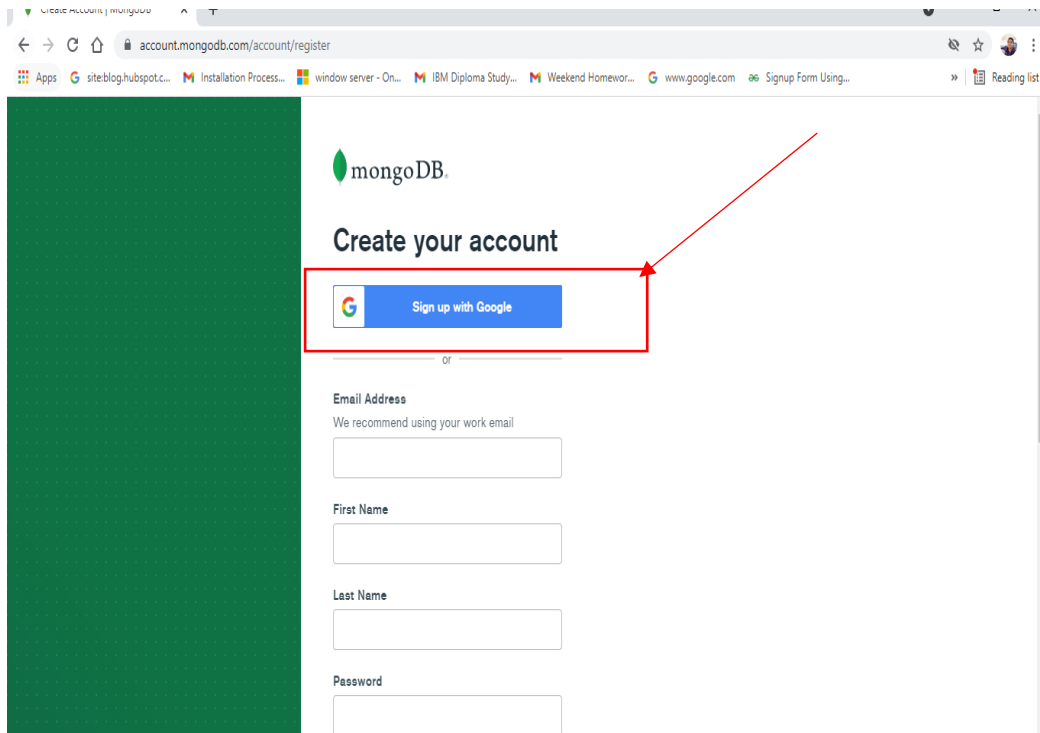


Click on sign up



Click on sign up with google

You select your email id



mongodb

Create your account

[Sign up with Google](#)

or

Email Address
We recommend using your work email

First Name

Last Name

Password

Accept term & condition

Check the button click on submit

mongodb

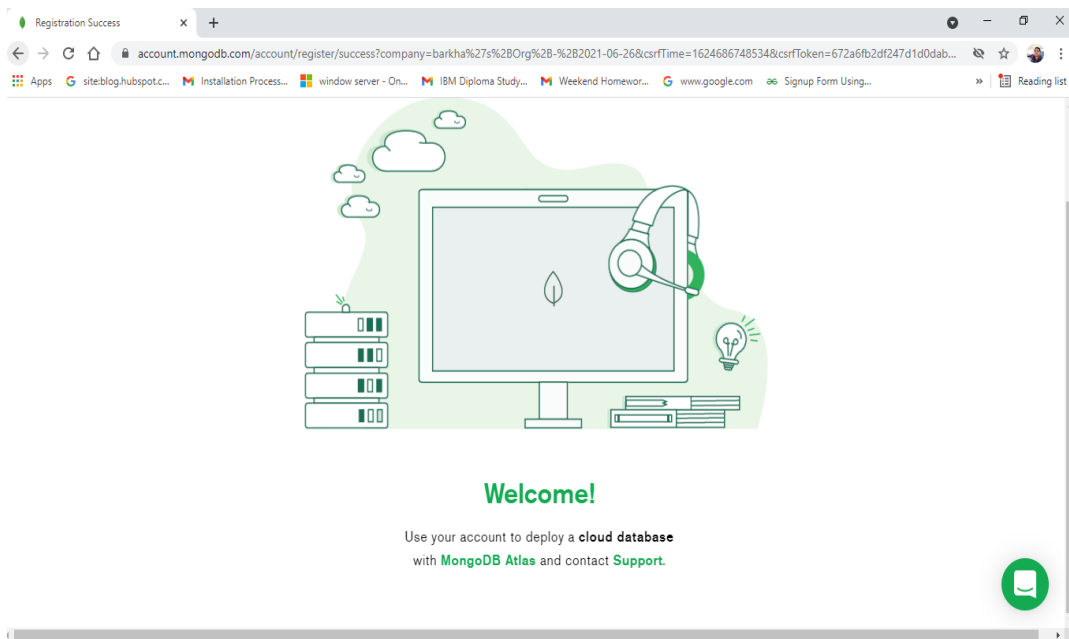
Accept Privacy Policy & Terms of Service

Please acknowledge the following terms and conditions to finish creating your account.

☒ I accept the [Privacy Policy](#) and the [Terms of Service](#)

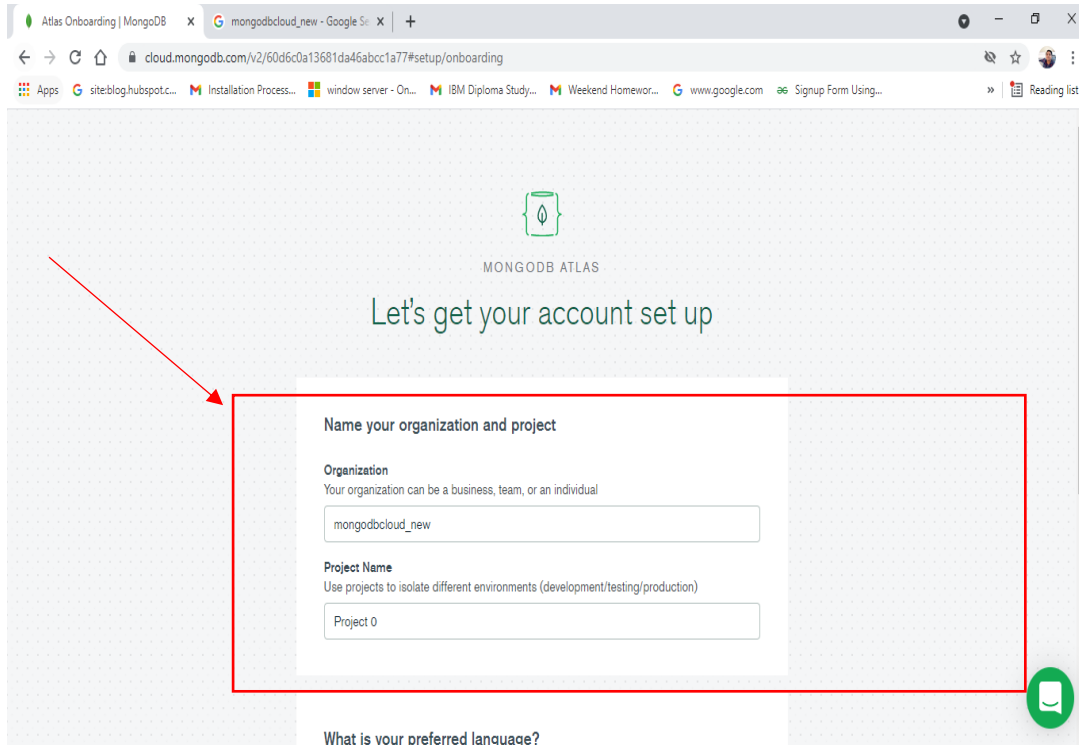
[Cancel Signup](#) [Submit](#)

You can create my account



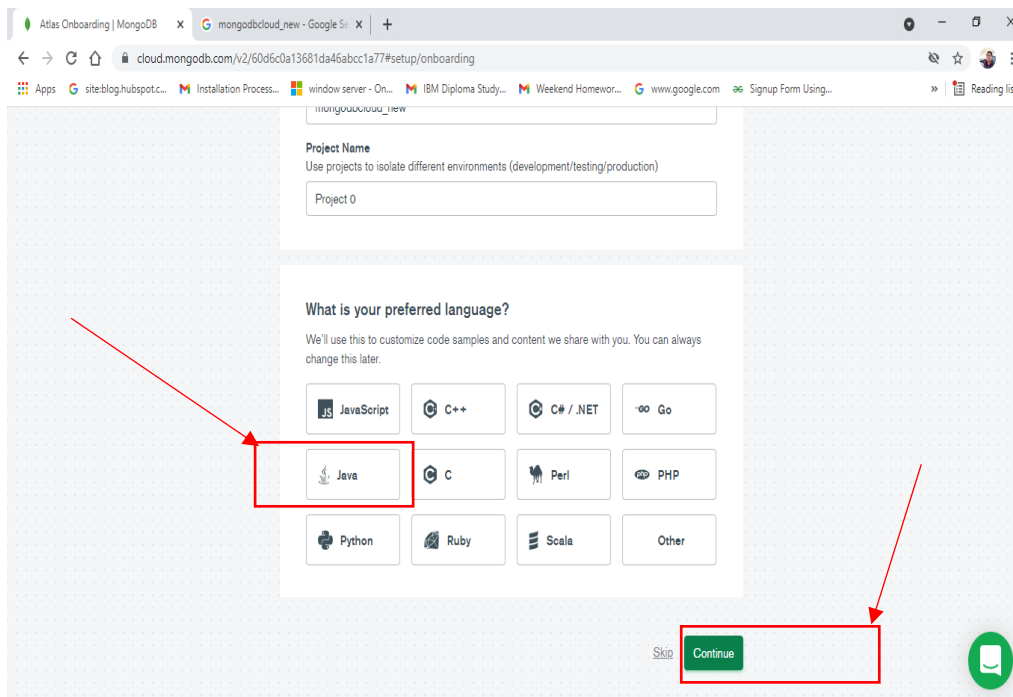
B.Account Setup

Setup your account or go ahead individual



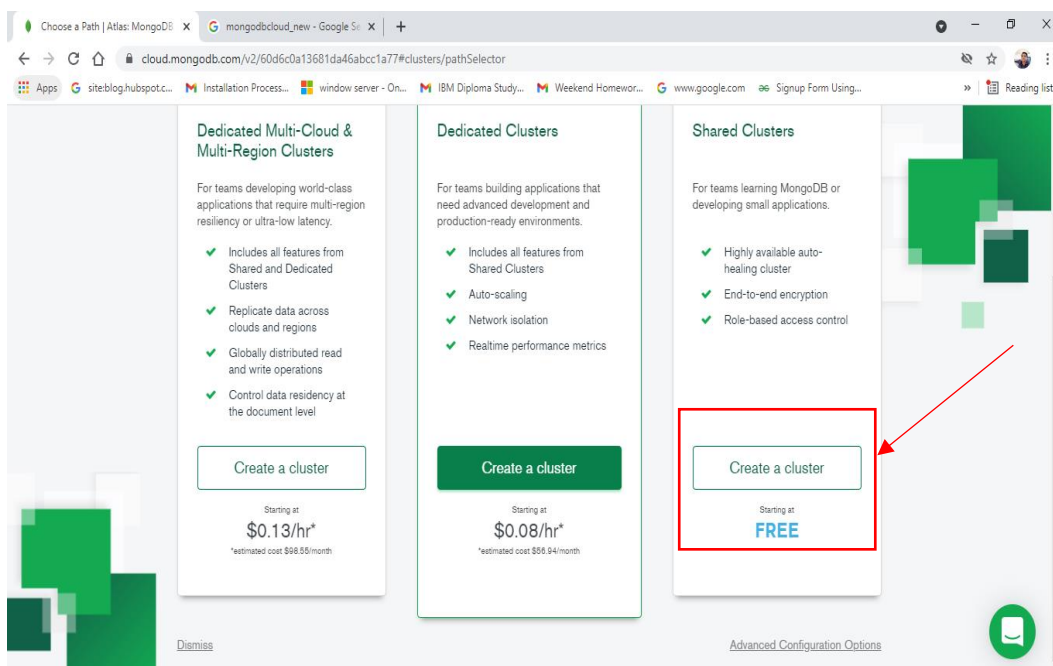
Select java language

Click on continue

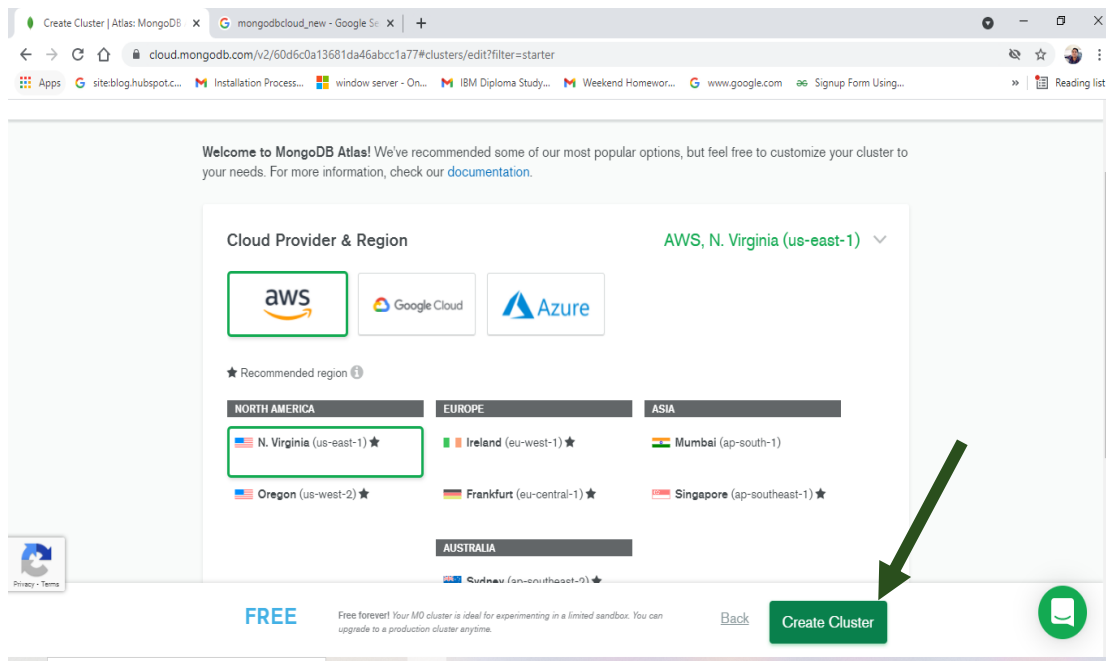


C. Create a cluster

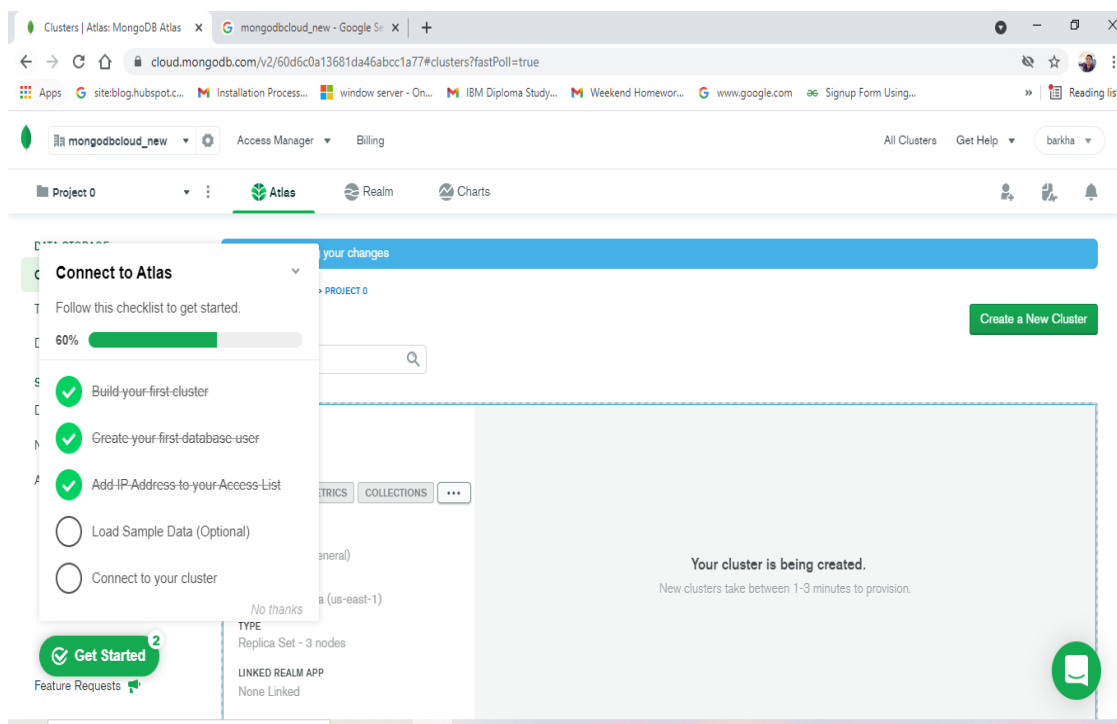
Select free plan



Click on create cluster

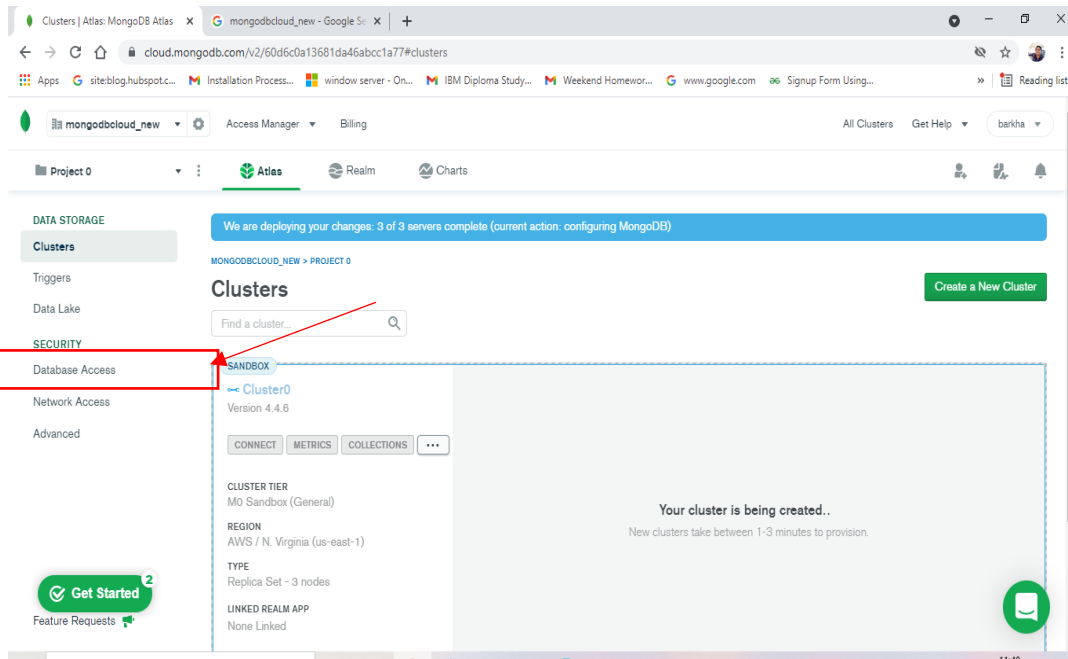


You can see open this page

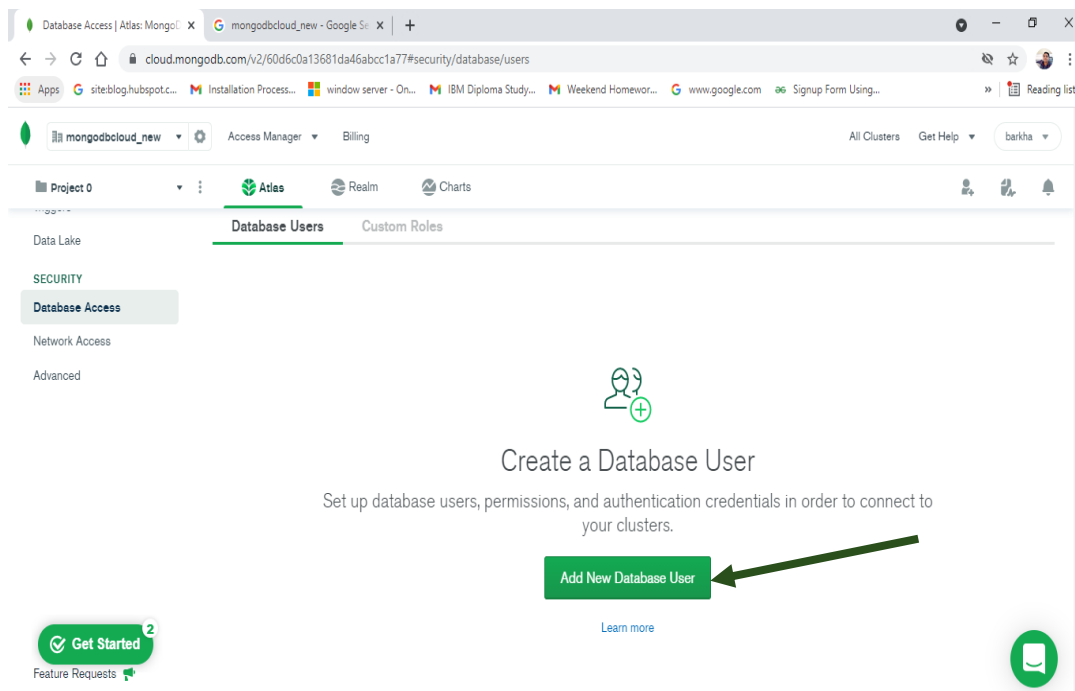


D.Add Data with Mongo dB

Click on data access



Click on add new user



Give the user's name & password

Click on add user

pushpa

..... SHOW

Autogenerate Secure Password Copy

Database User Privileges

Select a built-in role or privileges for this user.

Read and write to any database

Restrict Access to Specific Clusters/Data Lakes

Enable to specify the resources this user can access. By default, all resources in this project are accessible. ☐ OFF

Temporary User

This user is temporary and will be deleted after your specified duration of 6 hours, 1 day, or 1 week. ☐ OFF

Cancel Add User

You able to see added your user

We are deploying your changes (current action: configuring MongoDB)

MONGODBCLOUD_NEW > PROJECT 0

Database Access

Database Users Custom Roles

+ ADD NEW DATABASE USER

User Name	Authentication Method	MongoDB Roles	Resources	Actions
pushpa	SCRAM	readWriteAnyDatabase@admin	All Resources	EDIT DELETE

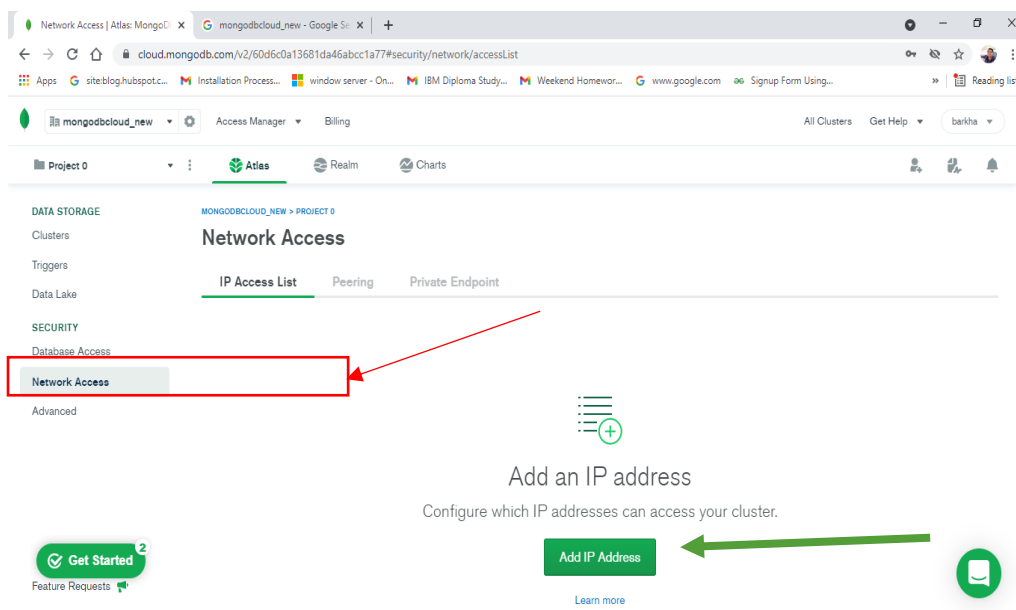
Get Started

System Status: All Good

©2021 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

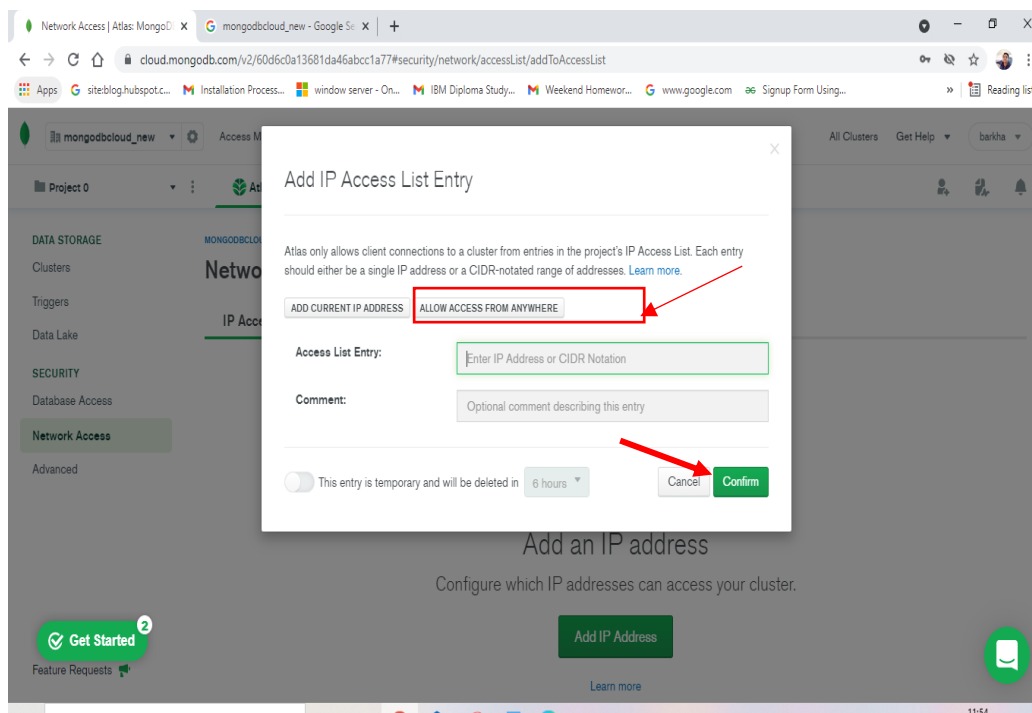
Click on network access

Add Ip address



Select allow anywhere

Click on confirm



You can see active status in Network Access page

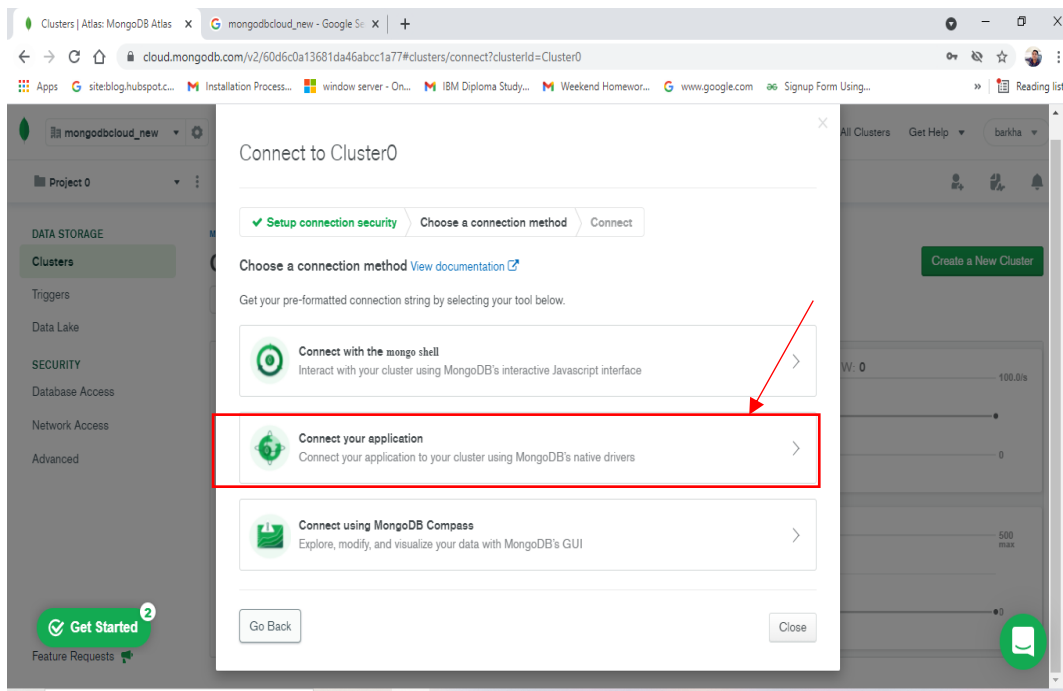
The screenshot shows the MongoDB Atlas interface for a project named 'mongodcloud_new'. The left sidebar has a 'Network Access' section highlighted. The main content area is titled 'Network Access' and shows the 'IP Access List' tab. A red box highlights the first entry in the list, which has an IP address of '0.0.0.0/0' and a status of 'Active'. The status is indicated by a green dot and the word 'Active'. There are 'EDIT' and 'DELETE' buttons next to the entry. The page also shows a 'Get Started' button and a 'System Status: All Good' message.

Go to cluster

Click on connect

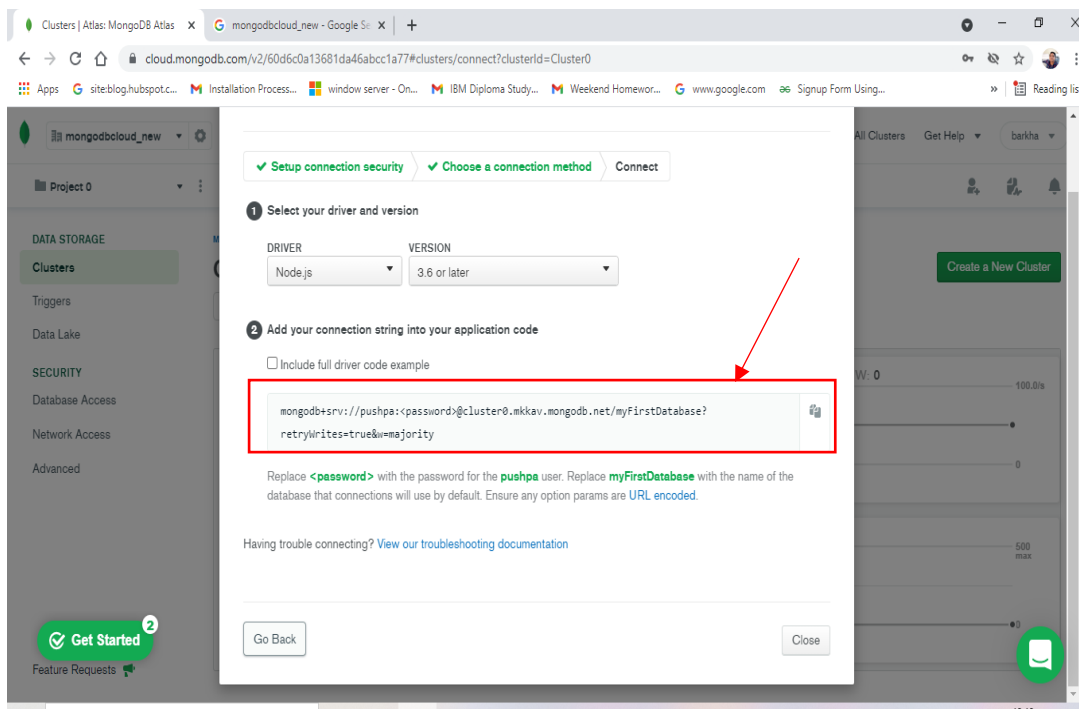
The screenshot shows the MongoDB Atlas interface for a project named 'mongodcloud_new'. The left sidebar has a 'Clusters' section highlighted. The main content area is titled 'Clusters' and shows a search bar and a 'Create a New Cluster' button. A red box highlights the 'CONNECT' button. The page also shows a 'Get Started' button and a 'System Status: All Good' message.

Select connect your application



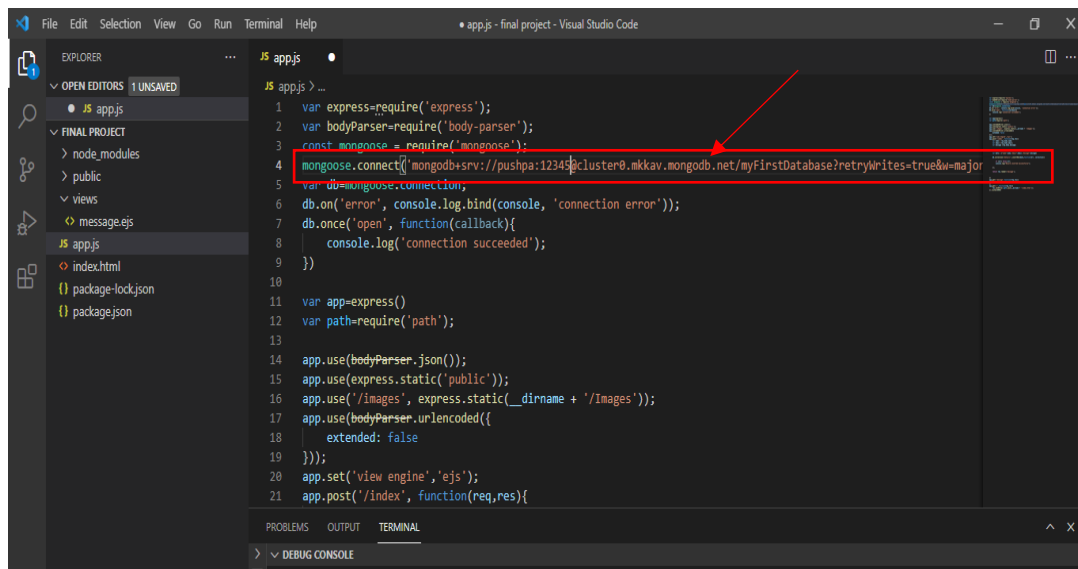
Copy the link

Paste it your app.js file



You can see it image

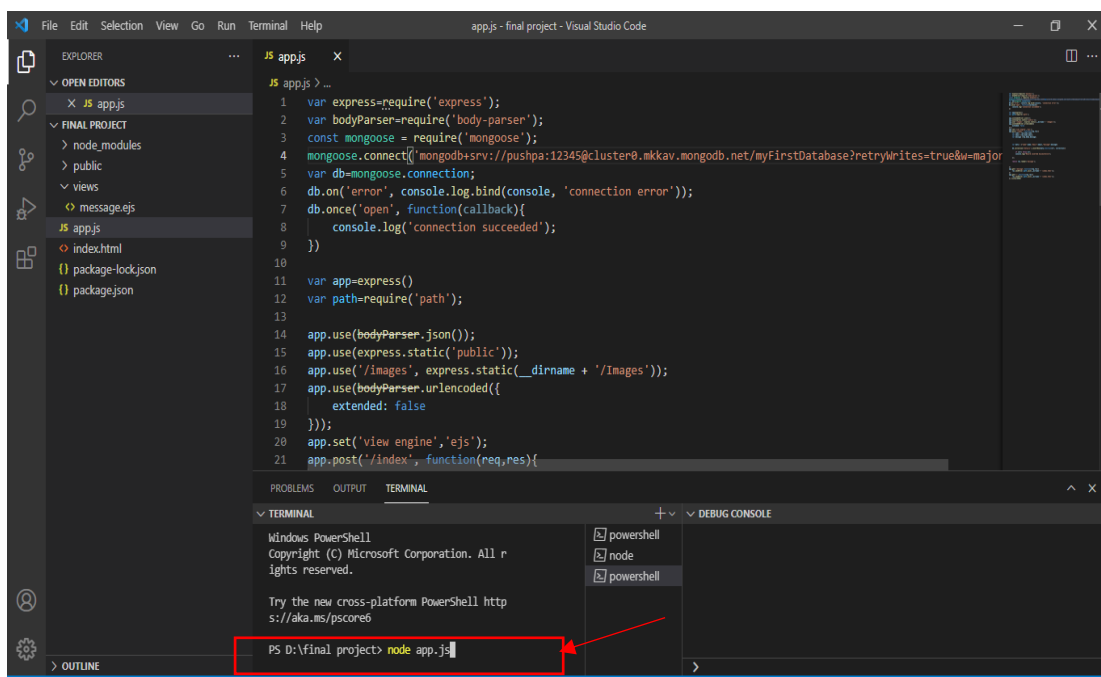
Save it



```
1 var express=require('express');
2 var bodyParser=require('body-parser');
3 const mongoose = require('mongoose');
4 mongoose.connect('mongodb+srv://pushpa:12345@cluster0.mkkav.mongodb.net/myFirstDatabase?retryWrites=true&w=majority');
5 var db=mongoose.connection;
6 db.on('error', console.log.bind(console, 'connection error'));
7 db.once('open', function(callback){
8   console.log('connection succeeded');
9 })
10
11 var app=express()
12 var path=require('path');
13
14 app.use(bodyParser.json());
15 app.use(express.static('public'));
16 app.use('/images', express.static(__dirname + '/Images'));
17 app.use(bodyParser.urlencoded({
18   extended: false
19 }));
20 app.set('view engine','ejs');
21 app.post('/index', function(req,res){
```

Run the command node app.js in terminal

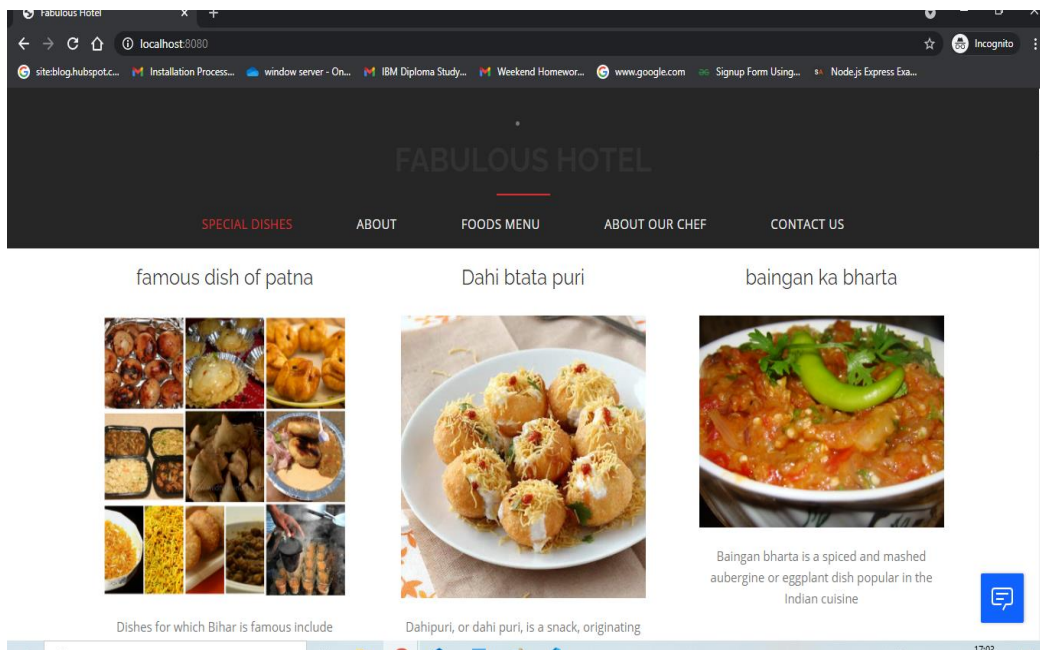
Press enter



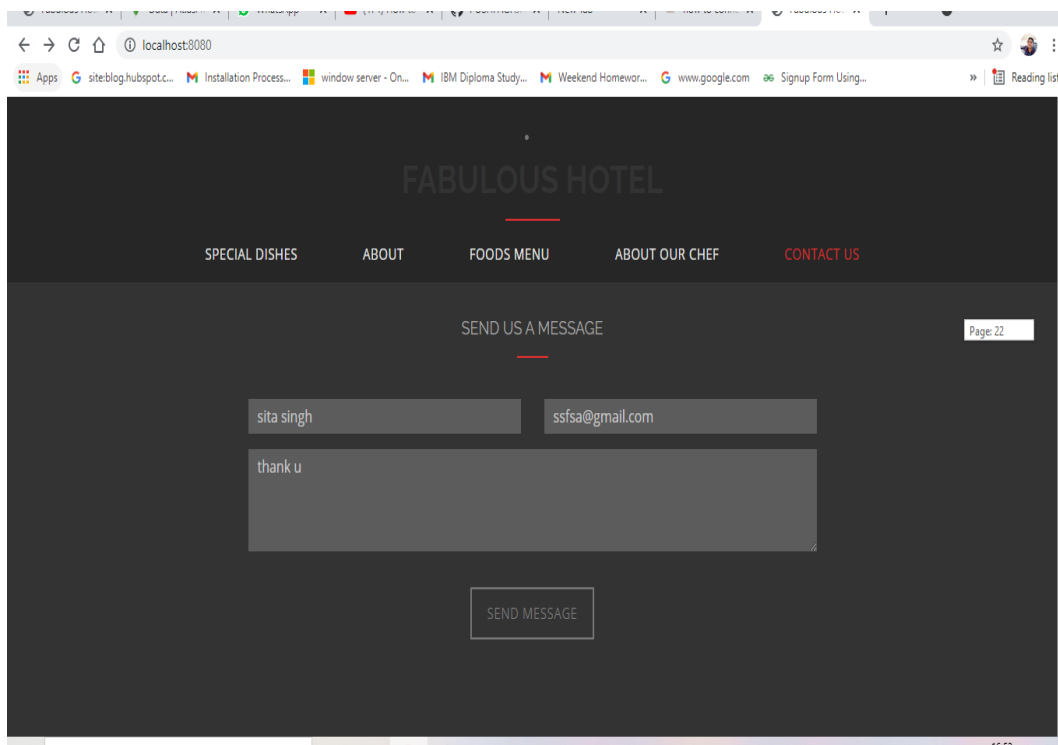
```
1 var express=require('express');
2 var bodyParser=require('body-parser');
3 const mongoose = require('mongoose');
4 mongoose.connect('mongodb+srv://pushpa:12345@cluster0.mkkav.mongodb.net/myFirstDatabase?retryWrites=true&w=majority');
5 var db=mongoose.connection;
6 db.on('error', console.log.bind(console, 'connection error'));
7 db.once('open', function(callback){
8   console.log('connection succeeded');
9 })
10
11 var app=express()
12 var path=require('path');
13
14 app.use(bodyParser.json());
15 app.use(express.static('public'));
16 app.use('/images', express.static(__dirname + '/Images'));
17 app.use(bodyParser.urlencoded({
18   extended: false
19 }));
20 app.set('view engine','ejs');
21 app.post('/index', function(req,res){
```

```
PS D:\final project> node app.js
```

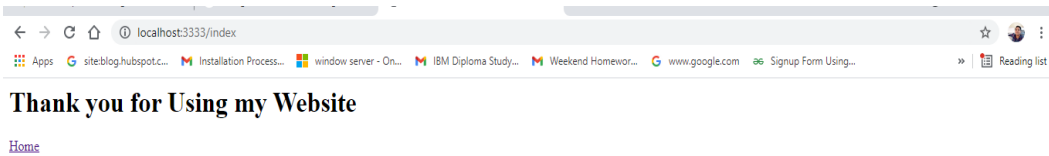
My webpage run on 8080 port no Sucessfully



User sending his/her feedback

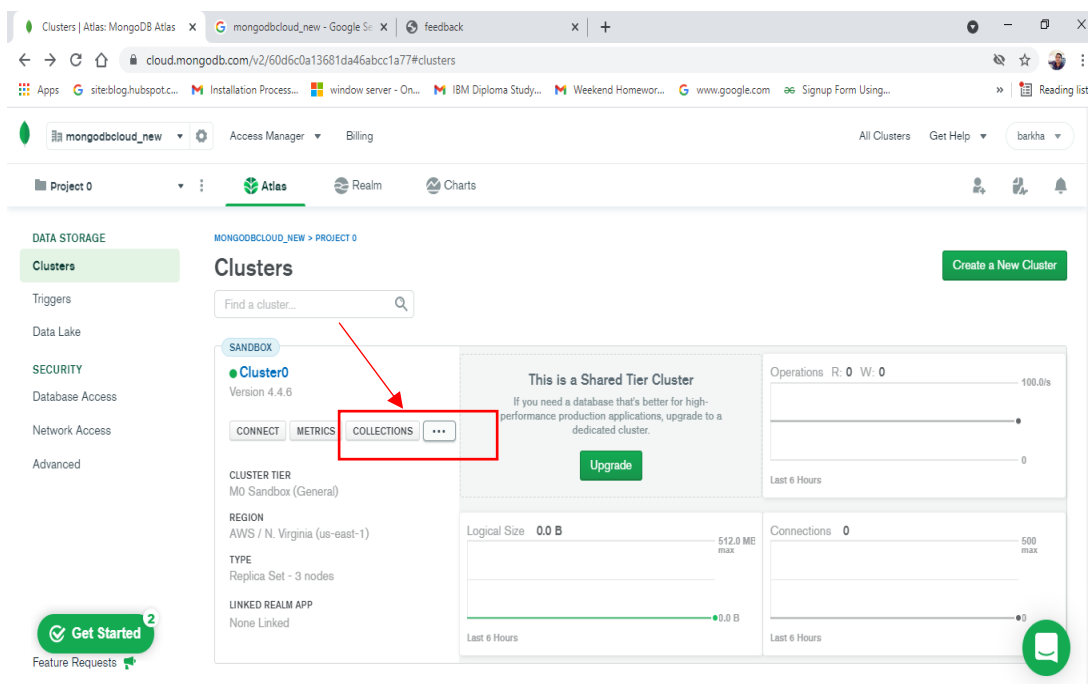


After sending the user query thank page will appear



See where store your data in atlas

Click on collections



You can see by default creating a database myFirstDatabase and collection details

Store user data in details

The screenshot displays the MongoDB Atlas web interface. The left sidebar shows the 'Database Access' section with a tree view containing 'myFirstDatabase' and its collection 'details'. A red box highlights the 'details' collection, with a red arrow pointing to it. The main panel shows the 'myFirstDatabase.details' page. At the top, it displays 'COLLECTION SIZE: 91B', 'TOTAL DOCUMENTS: 1', and 'INDEXES TOTAL SIZE: 4KB'. Below this, there are tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. The 'Find' tab is active, showing a filter input field with the text '(*filter": "example")'. Below the filter, it says 'QUERY RESULTS 1-1 OF 1'. A red box highlights the single document returned, which contains the following fields: '_id: ObjectId("60d5b994f45c823741f8de2")', 'name: "sita singh"', 'email: "ssfsa@gmail.com"', and 'message: "thank u"'. A red arrow points to this document. The bottom of the interface shows the 'System Status: All Good' and copyright information for MongoDB, Inc. 2021.

Source code

Index.html

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Fabulous Hotel</title>


<link rel="shortcut icon" href="img/favicon.ico" type="image/x-icon">

<link rel="apple-touch-icon" href="img/apple-touch-icon.png">

<link rel="apple-touch-icon" sizes="72x72" href="img/apple-touch-icon-72x72.png">

<link rel="apple-touch-icon" sizes="114x114" href="img/apple-touch-icon-114x114.png">


<!-- Bootstrap -->

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">

<link rel="stylesheet" type="text/css" href="fonts/font-awesome/css/font-awesome.css">


<!-- Stylesheet -->

<link rel="stylesheet" type="text/css" href="css/style.css">

<link href="https://fonts.googleapis.com/css?family=Raleway:300,400,500,600,700" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,400,600,700" rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Rochester" rel="stylesheet">


<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
```

```

</head>

<body id="page-top" data-spy="scroll" data-target=".navbar-fixed-top">

<nav id="menu" class="navbar navbar-default navbar-fixed-top">

  <div class="container">

    <!-- Brand and toggle get grouped for better mobile display -->

    <div class="navbar-header">

      <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1"> <span class="sr-only">Toggle navigation</span> <span class="icon-bar"></span>
      <span class="icon-bar"></span> <span class="icon-bar"></span> </button>

    </div>

    <!-- Collect the nav links, forms, and other content for toggling -->

    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">

      <ul class="nav navbar-nav"></ul>

      <li><h2><b>fabulous hotel</b></h2> </li>

      <ul class="nav navbar-nav">

        <li><a href="#features" class="page-scroll">Special dishes</a></li>

        <li><a href="#about" class="page-scroll">About</a></li>

        <li><a href="#restaurant-menu" class="page-scroll"> foods Menu </a></li>

        <li><a href="#team" class="page-scroll">about our Chef</a></li>

        <li><a href="#contact" class="page-scroll">Contact us</a></li>

      </ul>

    </div>

    <!-- /.navbar-collapse -->

  </div>

```

```
</nav>

<!-- Header -->

<header id="header">

  <div class="intro">

    <div class="overlay">

      <div class="container">

        <div class="row">

          <div class="intro-text">

            <h1>Fabulous Hotel</h1>

            <p>welcome to Fabulous hotel</p>

          </div>

        </div>

      </div>

    </div>

  </div>

</header>

<!-- Features Section -->

<div id="features" class="text-center">

  <div class="container">

    <div class="section-title">

      <h2>Our Specials dishes</h2>

    </div>

    <div class="row">

      <div class="col-xs-12 col-sm-4">

        <div class="features-item">

          <h3>famous dish of patna</h3>

          
```

<p>Dishes for which Bihar is famous include Bihari kebabs, litti chokha, Bihari boti, Bihari chicken masala, sattu paratha (parathas stuffed with roasted gram flour), chokha (spicy mashed eggplant and potatoes), fish curry and posta-dana ka halwa.

</p>

</div>

</div>

<div class="col-xs-12 col-sm-4">

<div class="features-item">

<h3>Dahi btata puri</h3>

<p>Dahipuri, or dahi puri, is a snack, originating from India, The dish is a form of chaat and originates from the city of Mumbai. It is served with mini-puri shells, which are more popularly recognized from the dish pani puri</p>

</div>

</div>

<div class="col-xs-12 col-sm-4">

<div class="features-item">

<h3>baingan ka bharta</h3>

<p>Baingan bharta is a spiced and mashed aubergine or eggplant dish popular in the Indian cuisine</p>

</div>

</div>

</div>

</div>

</div>

<!-- About Section -->

<div id="about">


```

<div class="container-fluid">

  <div class="row">

    <div class="col-xs-12 col-md-6 about-img"> </div>

    <div class="col-xs-12 col-md-3 col-md-offset-1">

      <div class="about-text">

        <div class="section-title">

          <h2>About Our hotel</h2>

        </div>

        <p>Fabulous one of the fine dining restaurants . It's the place to go for North Indian food, as well as anything with a touch of North West Frontier Province flavors.</p>

        <p> The ambiance is somewhat classical with solid furniture and exposed red brick walls. Specialities include meat heavy Mughali, Afghan, and tandoori dishes.</p>

      </div>

    </div>

  </div>

</div>

<!-- Restaurant Menu Section -->

<div id="restaurant-menu">

  <div class="container">

    <div class="section-title text-center">

      <h2>Food Menu</h2>

    </div>

    <div class="row">

      <div class="col-xs-12 col-sm-6">

        <div class="menu-section">

          <h2 class="menu-section-title">Breakfast & Starters</h2>

          <div class="menu-item">

```

<h2 class="menu-item">For vegetarian</h2>

<div class="menu-item-name">Multigrain Roti</div>

<div class="menu-item-price"> 90Rs. </div>

<div class="menu-item-name">Oats Upma</div>

<div class="menu-item-price"> 70Rs. </div>

<div class="menu-item-name">Ven Pongal</div>

<div class="menu-item-price"> 80Rs. </div>

<div class="menu-item-name">Methi Thalipeeth</div>

<div class="menu-item-price"> 110Rs. </div>

<div class="menu-item-name">Green Tea</div>

<div class="menu-item-price"> 70Rs. </div>

<div class="menu-item-name">Coffee</div>

<div class="menu-item-price"> 60Rs. </div>

<h2 class="menu-item">For Non-vegetarian</h2>

<div class="menu-item-name">Grilled Chicken Escalope with Fresh Salsa</div>

<div class="menu-item-price">290Rs. </div>

<div class="menu-item-name">Mutton Korma.</div>

<div class="menu-item-price"> 170Rs. </div>

<div class="menu-item-name">Pina Colada Pork Ribs</div>

<div class="menu-item-price"> 280Rs. </div>

<div class="menu-item-name">Tandoori Lamb Chops</div>

<div class="menu-item-price"> 110Rs. </div>

<div class="menu-item-name">Eggs</div>

<div class="menu-item-price"> 40Rs. </div>

<div class="menu-item-name">Malabar Fish Biryani</div>

<div class="menu-item-price"> 170Rs. </div>

```
<div class="menu-item-description"> Kerala style Fish Biryani is prepared using Khaima/ Jeerakasala rice. Seer Fish/ Neymmeen or any other fleshy fish can be used for making this </div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Fried Ravioli</div>
```

```
<div class="menu-item-price"> 700RS. </div>
```

```
<div class="menu-item-description">Dredge ravioli in the bread crumbs. Place the ravioli on a baking sheet, and continue with the remaining ravioli. When the oil is hot, fry the ravioli in batches, turning occasionally, until golden brown, about 3 minutes. Using a slotted spoon, transfer the fried ravioli to paper towels to drain.
```

```
</div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Wild Mushroom Arancini</div>
```

```
<div class="menu-item-price"> 290Rs. </div>
```

```
<div class="menu-item-description"> Ingredients. 2 cups wild mushroom risotto, cooled. 2 large eggs, beaten to blend. Directions. In a large, heavy saucepan, add about 2" of oil. Heat the oil over medium heat until it's bout 350F. Notes. Feel free to use whatever leftover risotto you may have on hand for this – or your favorite risotto recipe! </div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="col-xs-12 col-sm-6">
```

```
<div class="menu-section">
```

```
<h2 class="menu-section-title">Main Dishes</h2>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Litti chokha</div>
```

```
<div class="menu-item-price"> 180Rs.</div>
```

```
<div class="menu-item-description">Litti, along with chokha, is a complete meal originated from the Indian state of Bihar. It is also popular in Jharkhand and parts of Eastern Uttar Pradesh. </div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Chicken & Mushroom Pasta</div>
```

```
<div class="menu-item-price"> 420Rs. </div>
```

```
<div class="menu-item-description">How to make Chicken Mushroom Pasta (quick summary) Melt butter in pan. Add mushrooms. Fry until golden. Add wine, reduce down, then pour in chicken stock. Pour in cream and simmer to thicken. Add parmesan, parsley and seasoning. Stir in pasta. Add chicken strips</div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Biryani</div>
```

```
<div class="menu-item-price"> 190Rs. </div>
```

```
<div class="menu-item-description"> How to make Chicken Mushroom Pasta (quick summary) Melt butter in pan. Add mushrooms. Fry until golden. Add wine, reduce down, then pour in chicken stock. Pour in cream and simmer to thicken. Add parmesan, parsley and seasoning. Stir in pasta. Add chicken strips</div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Papri chat</div>
```

```
<div class="menu-item-price"> 190Rs. </div>
```

```
<div class="menu-item-description">Papri chat or papri chaat is a popular traditional fast food and street food from the Indian subcontinent, probably in North India, Bangladesh and Pakistan.</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="row">
```

```
<div class="col-xs-12 col-sm-6">
```

```
<div class="menu-section">
```

```
<h2 class="menu-section-title">Dinner</h2>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Caprese Pasta Salad.</div>
```

```
<div class="menu-item-price"> 250Rs. </div>
```

```
<div class="menu-item-description">cups cherry tomatoes, halved. 8 oz bocconcini fresh mozzarella balls (or fresh mozzarella cut into 1/2" cubes) 12 oz dried orecchiette pasta </div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Malai Kofta </div>
```

```
<div class="menu-item-price"> 170Rs. </div>
```

```
<div class="menu-item-description">Malai Kofta is a very popular Indian vegetarian dish where balls (kofta) made of potato and paneer are deep fried and served with a creamy and spiced tomato based curry. The literal translation of this dish is malai=creamy/buttery and kofta=spiced balls (in this case made of potatoes and paneer).</div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Mushroom Kofta in Tomato Gravy. .</div>
```

```
<div class="menu-item-price"> 280Rs. </div>
```

```
<div class="menu-item-description"> How to Make Mushroom Kofta in Tomato Gravy. Wash and chop spinach leaves. Remove and add paneer, corn flour, wheat flour and salt. For the Filling: Take oil in a pan, add ginger paste, green chillies, finely chopped white button mushrooms and salt. Add lemon juice and honey and mix well. </div>
```

```
</div>
```

```
<div class="menu-item">
```

```
<div class="menu-item-name">Spicy Coconut Salmon</div>
```

```
<div class="menu-item-price"> 420Rs. </div>
```

```
<div class="menu-item-description"> In a saucepan over medium heat, warm 1 teaspoon olive oil and cook garlic, onions, ginger, and red pepper flakes until softened, 3 minutes. Meanwhile, preheat a
```

large cast-iron skillet over medium-high heat. Serve salmon with warm spicy coconut milk sauce and top with red pepper flakes and cilantro.

25 mins</div>

</div>

</div>

</div>

<div class="col-xs-12 col-sm-6">

<div class="menu-section">

<h2 class="menu-section-title">Desserts</h2>

<div class="menu-item">

<div class="menu-item-name">Chocolate Mud Cake</div>

<div class="menu-item-price"> 410Rs. </div>

<div class="menu-item-description">Deliciously moist and rich, this is definitely THE best ever cake for chocolate lovers. One piece is never enough! </div>

</div>

<div class="menu-item">

<div class="menu-item-name">Bourbon-Pecan Tart</div>

<div class="menu-item-price"> 240Rs. </div>

<div class="menu-item-description">Bourbon Pecan Tart ... Gooey and sticky sweet, pecan pie is the quintessential Southern dessert. Here it is transformed into a buttery tart that requires less sugar</div>

</div>

<div class="menu-item">

<div class="menu-item-name">Texas Sheet Cake</div>

<div class="menu-item-price"> 500Rs. </div>

<div class="menu-item-description">A moist and delicious chocolate cake made in a jellyroll pan, covered with chocolate frosting and walnuts.</div>

</div>

<div class="menu-item">

```
<div class="menu-item-name">Vanilla Cheesecake</div>
```

```
<div class="menu-item-price">280Rs. </div>
```

```
<div class="menu-item-description"> A cheesecake can be savory (and served with crackers as an
appetizer), however, most of us think of the term as describing a luscious, rich and sweet dessert. The tex-
ture can vary greatly-from light and airy to dense and rich to smooth and creamy depending on the ingredi-
ents and mixing methods used.</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Gallery Section -->
```

```
<div id="gallery">
```

```
<div class="container-fluid">
```

```
<div class="row">
```

```
<div class="col-xs-6 col-md-3">
```

```
<div class="gallery-item"> </div>
```

```
</div>
```

```
<div class="col-xs-6 col-md-3">
```

```
<div class="gallery-item"> </div>
```

```
</div>
```

```
<div class="col-xs-6 col-md-3">
```

```
<div class="gallery-item"> </div>
```

```
</div>
```

```
<div class="col-xs-6 col-md-3">
```

```
<div class="gallery-item"> </div>
```

```
</div>
```



```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Team Section -->
```

```
<div id="team">
```

```
<div class="container">
```

```
<div id="row">
```

```
<div class="col-md-6">
```

```
<div class="col-md-10 col-md-offset-1">
```

```
<div class="section-title">
```

```
<h2>Meet Our Chef</h2>
```

```
</div>
```

```
<p>Our passionate culinary team is made up of innovative and experienced professionals with years of experience in foodservice. Their work combines food science, technology and the culinary arts to imagine the future of food.</p>
```

```
<p> They draw on current trends and culinary insights from across our global network to create customized solutions for all of our clients. Using proprietary flavor research, our team of forward-thinking chefs aims to guide the next phase of foodservice. Individually, they've earned accolades across the industry, but together they're crafting the future of flavor!
```

```
</p>
```

```
</div>
```

```
</div>
```

```
<div class="col-md-6">
```

```
<div class="team-img"></div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Contact Section -->

<div id="contact" class="text-center">

  <div class="container text-center">

    <div class="col-md-4">

      <h3>Reservations</h3>

      <div class="contact-item">

        <p>Please call</p>

        <p>+918998763320</p>

      </div>

    </div>

    <div class="col-md-4">

      <h3>Address</h3>

      <div class="contact-item">

        <p>Patna(Bihar)india</p>

        <p> boring road 80001</p>

      </div>

    </div>

    <div class="col-md-4">

      <h3>Opening Hours</h3>

      <div class="contact-item">

        <p>Mon-Thus: 10:00 AM - 11:00 PM</p>

        <p>Fri-Sun: 11:00 AM - 02:00 AM</p>

      </div>

    </div>

  </div>

</div>

<div class="container">

  <div class="section-title text-center">
```

<h3>Send us a message</h3>

</div>

<div class="col-md-8 col-md-offset-2">

<form action="/index" method="POST" name="sendMessage" id="contactForm" novalidate>

<div class="row">

<div class="col-md-6">

<div class="form-group">

<input type="text" id="name" name="name" class="form-control" placeholder="Type your Name" required="required">

<p class="help-block text-danger"></p>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

<input type="email" id="email" name="email" class="form-control" placeholder="Type your Email" required="required">

<p class="help-block text-danger"></p>

</div>

</div>

</div>

<div class="form-group">

<textarea name="message" id="message" class="form-control" name="message" rows="4" placeholder="Type your Message" required></textarea>

<p class="help-block text-danger"></p>

```

</div>

<div id="success"></div>

<button type="submit" class="btn btn-custom btn-lg">Send Message</button>

</form>

</div>

</div>

</div>

<div id="footer">

<div class="container text-center">

<div class="col-md-6">

</div>

<div class="col-md-6">

<div class="social">

<ul>

<li><a href="#"><i class="fa fa-facebook"></i></a></li>

<li><a href="#"><i class="fa fa-twitter"></i></a></li>

<li><a href="#"><i class="fa fa-youtube"></i></a></li>

</ul>

</div>

</div>

</div>

</div>

<!--waston assistant -->

<script>

window.watsonAssistantChatOptions = {

  integrationID: "5103c32c-f97b-475c-a8a8-01e5c1dc3fed", // The ID of this integration.

  region: "eu-gb", // The region your integration is hosted in.

```

```
serviceInstanceId: "5ce2bc46-b04c-4517-8eb0-fcef75251731", // The ID of your service instance.
```

```
onLoad: function(instance) { instance.render(); }
```

```
};
```

```
setTimeout(function(){
```

```
const t=document.createElement('script');
```

```
t.src="https://web-chat.global.assistant.watson.appdomain.cloud/loadWatsonAssistantChat.js";
```

```
document.head.appendChild(t);
```

```
});
```

```
</script>
```

```
<script type="text/javascript" src="js/jquery.1.11.1.js"></script>
```

```
<script type="text/javascript" src="js/bootstrap.js"></script>
```

```
<script type="text/javascript" src="js/SmoothScroll.js"></script>
```

```
<script type="text/javascript" src="js/jqBootstrapValidation.js"></script>
```

```
<script type="text/javascript" src="js/main.js"></script>
```

```
</body>
```

```
</html>
```

Style.css

```
body, html {
```

```
font-family: 'Open Sans', serif;
```

```
text-rendering: optimizeLegibility !important;
```

```
-webkit-font-smoothing: antialiased !important;
```

```
color: #777;
```

```
font-weight: 400;
```

```
width: 100% !important;
```

```
height: 100% !important;
}
h1, h2, h3, h4 {
    font-family: 'Raleway', sans-serif;
}
h2 {
    font-weight: 400;
    font-size: 38px;
    color: #333;
    text-transform: uppercase;
    position: relative;
    margin-top: 10px;
    margin-bottom: 15px;
    padding-bottom: 20px;
}
h2::after {
    position: absolute;
    content: "";
    background: #d43031;
    height: 2px;
    width: 70px;
    bottom: 0;
    margin-left: -35px;
    left: 50%;
}
h3 {
    font-size: 24px;
```

```
font-weight: 400;
color: #333;
}
h4 {
font-size: 24px;
text-transform: uppercase;
font-weight: 400;
color: #333;
}
h5 {
text-transform: uppercase;
font-weight: 700;
line-height: 20px;
}
p {
font-size: 15px;
}
p.intro {
margin: 12px 0 0;
line-height: 24px;
}
a {
color: #8eb640;
}
a:hover, a:focus {
text-decoration: none;
color: #222;
```

```
}

ul, ol {

    list-style: none;

}

ul, ol {

    padding: 0;

    webkit-padding: 0;

    moz-padding: 0;

}

.btn:active, .btn.active {

    background-image: none;

    outline: 0;

    -webkit-box-shadow: none;

    box-shadow: none;

}

a:focus, .btn:focus, .btn:active:focus, .btn.active:focus, .btn.focus, .btn:active.focus, .btn.active.focus {

    outline: none;

    outline-offset: none;

}

/* Navigation */

.navbar .navbar-nav {

    display: inline-block;

    float: none;

    vertical-align: top;

}

.navbar .navbar-collapse {

    text-align: center;
```



```
}

#menu {

    padding: 20px;

    transition: all 0.8s;

}

#menu.navbar-default {

    background-color: rgba(248, 248, 248, 0);

    border-color: rgba(231, 231, 231, 0);

}

#menu.navbar-default .navbar-nav > li > a {

    text-transform: uppercase;

    color: #eee;

    font-weight: 400;

    font-size: 15px;

    padding: 5px 0;

    border: 2px solid transparent;

    letter-spacing: 0.5px;

    margin: 0 40px 0 40px;

}

#menu.navbar-default .navbar-nav > li > a:hover {

    color: #d43031;

}

.on {

    background-color: #262626 !important;

    padding: 0 !important;

    padding: 10px 0 !important;

}
```

```
.navbar-default .navbar-nav > .active > a, .navbar-default .navbar-nav > .active > a:hover, .navbar-default
.navbar-nav > .active > a:focus {

    color: #d43031 !important;

    background-color: transparent;

}

.navbar-toggle {

    border-radius: 0;

}

.navbar-default .navbar-toggle:hover, .navbar-default .navbar-toggle:focus {

    background-color: #d43031;

    border-color: #d43031;

}

.navbar-default .navbar-toggle .icon-bar {

    background-color: #FFF;

}

.navbar-default .navbar-toggle:hover>.icon-bar {

    background-color: #FFF;

}

.section-title {

    margin-bottom: 70px;

}

.section-title .overlay {

    padding: 80px 0;

    background: rgba(0, 0, 0, 0.7);

}

.section-title p {

    font-size: 22px;

}
```

```
.section-title hr {  
    margin: 0 auto;  
    margin-bottom: 40px;  
}  
  
.btn-custom {  
    text-transform: uppercase;  
    color: #777;  
    background-color: transparent;  
    border: 2px solid #777;  
    padding: 14px 20px;  
    margin: 0;  
    font-size: 15px;  
    font-weight: 500;  
    letter-spacing: 0.5px;  
    border-radius: 0;  
    margin-top: 20px;  
    transition: all 0.3s;  
}  
  
.btn-custom:hover, .btn-custom:focus, .btn-custom.focus, .btn-custom:active, .btn-custom.active {  
    color: #ddd;  
    border: 2px solid #d43031;  
    background-color: #d43031;  
}  
  
/* Header Section */  
  
.intro {  
    display: table;  
    width: 100%;
```

```
padding: 0;

background: url(../img/intro-bg.jpg) center center no-repeat;

background-color: #e5e5e5;

-webkit-background-size: cover;

-moz-background-size: cover;

background-size: cover;

-o-background-size: cover;

}

.intro .overlay {

    background: rgba(0,0,0,0.4);

}

.intro h1 {

    font-family: 'Rochester', cursive;

    color: #fff;

    font-size: 10em;

    font-weight: 400;

    margin-top: 0;

}

.intro span {

    color: #a7c44c;

    font-weight: 600;

}

.intro p {

    color: #fff;

    font-size: 24px;

    font-weight: 400;

    margin-top: 150px;
```

```
}

header .intro-text {

    padding-top: 200px;

    padding-bottom: 50px;

    text-align: center;

}

/* Features Section */

#features {

    padding: 120px 0;

}

#features .features-item {

    margin: 0 20px;

}

#features img {

    margin: 30px auto 20px;

}

#features .about-img:before {

    display: block;

    content: "";

    position: absolute;

    top: 8px;

    right: 8px;

    bottom: 8px;

    left: 8px;

    border: 1px solid rgba(255, 255, 255, 0.5);

}

#features p {
```

```
        line-height: 24px;

        margin: 15px 0 30px;

    }

/* About Section */

#about {

    padding: 0;

    background: #f6f6f6;

}

#about h2::after {

    bottom: 0;

    margin-left: 0;

    left: 0;

}

#about .about-text {

    padding: 80px 0;

}

#about .about-img {

    background: #444 url(..img/gallery-bg.jpg) center center no-repeat;

    background-size: cover;

    height: 570px;

}

#about p {

    line-height: 24px;

    margin: 15px 0 30px;

}

/* Menu Section */

#restaurant-menu {
```

```
padding: 100px 0 60px 0;
}
#restaurant-menu img {
    width: 300px;
    box-shadow: 15px 0 #a7c44c;
}
#restaurant-menu h3 {
    padding: 10px 0;
    text-transform: uppercase;
}
#restaurant-menu .menu-section hr {
    margin: 0 auto;
}
#restaurant-menu .menu-section {
    margin: 0 20px 80px;
}
#restaurant-menu .menu-section-title {
    font-size: 32px;
    display: block;
    font-weight: 400;
    color: #444;
    margin: 20px 0;
    text-align: center;
}
#restaurant-menu .menu-item {
    margin: 45px 0;
    font-size: 18px;
```

```
}

#restaurant-menu .menu-item-name {

    font-weight: 400;

    font-size: 20px;

    color: #444;

    margin-bottom: 10px;

}

#restaurant-menu .menu-item-description {

    font-size: 15px;

    width: 85%;

}

#restaurant-menu .menu-item-price {

    float: right;

    font-weight: 400;

    color: #555;

    margin-top: -36px;

}

/* Gallery Section */

#gallery {

    padding: 0;

}

#gallery .gallery-item {

    margin: 0 -15px;

}

/* Team Section */

#team {

    padding: 120px 0;
```



```
}

#team h2::after {

    bottom: 0;

    margin-left: 0;

    left: 0;

}

#team img {

    max-width: 550px;

}

#team .team-img {

    display: inline-block;

    position: relative;

}

/* Contact Section */

#contact {

    padding: 100px 0;

    background: #333;

}

#contact .contact-item p {

    font-size: 15px;

    color: #777;

}

#contact .section-title {

    margin-top: 60px;

    margin-bottom: 40px;

}

#contact form {
```

```
padding: 0;
}
#contact h3 {
    position: relative;
    text-transform: uppercase;
    font-size: 18px;
    font-weight: 400;
    color: #aaa;
    padding: 20px 0;
}
#contact h3::after {
    position: absolute;
    content: "";
    background: #d43031;
    height: 2px;
    width: 40px;
    bottom: 0;
    margin-left: -20px;
    left: 50%;
}
#contact .text-danger {
    color: #cc0033;
    text-align: left;
}
label {
    font-size: 12px;
    font-weight: 400;
```

```
font-family: 'Open Sans', sans-serif;

float: left;

}

#contact .form-control {

    display: block;

    width: 100%;

    padding: 6px 12px;

    font-size: 16px;

    line-height: 1.42857143;

    color: #ccc;

    background-color: rgba(255,255,255,.2);

    background-image: none;

    border: 0;

    border-radius: 0;

    -webkit-box-shadow: none;

    box-shadow: none;

    -webkit-transition: none;

    -o-transition: none;

    transition: none;

}

#contact .form-control:focus {

    border-color: #999;

    outline: 0;

    -webkit-box-shadow: transparent;

    box-shadow: transparent;

}

.form-control::-webkit-input-placeholder {
```

```
color: #999;
}

.form-control:-moz-placeholder {
color: #999;
}

.form-control::-moz-placeholder {
color: #999;
}

.form-control:-ms-input-placeholder {
color: #999;
}

#contact .contact-item {
margin: 20px 0 40px 0;
}

#contact .contact-item span {
font-weight: 400;
color: #aaa;
text-transform: uppercase;
margin-bottom: 6px;
display: inline-block;
}

/* Footer Section */

#footer {
background: #262626;
padding: 40px 0 20px 0;
}

#footer .social {
margin-top: -5px;
```

```
        text-align: right;
    }
#footer .social ul li {
    display: inline-block;
    margin: 0 15px;
}
#footer .social i.fa {
    font-size: 24px;
    padding: 4px;
    color: #888;
    transition: all 0.3s;
}
#footer .social i.fa:hover {
    color: #eee;
}
#footer p {
    font-size: 14px;
    color: #666;
}
#footer a {
    color: #999;
}
#footer a:hover {
    color: #eee;
}
```

App.js

```
var express=require('express');

var bodyParser=require('body-parser');

const mongoose = require('mongoose');

mongoose.connect('mongodb+srv://pushpa:12345@cluster0.2oez1.mongodb.net/myFirstDatabase?retry-
Writes=true&w=majority');

var db=mongoose.connection;

db.on('error', console.log.bind(console, 'connection error'));

db.once('open', function(callback){

    console.log('connection succeeded');

})


var app=express ()

var path=require('path');

app.use(bodyParser.json());

app.use(express.static('public'));

app.use('/images', express.static(__dirname + '/Images'));

app.use(bodyParser.urlencoded({

    extended: false

}));

app.set('view engine','ejs');

app.post('/index', function(req,res){

    var name = req.body.name;

    var email =req.body.email;

    var message =req.body.message;
```

```
var data = {"name":name,"email":email,"message":message}

db.collection('details').insertOne(data,function(err, collection){

    if (err) throw err;

    console.log("Record inserted Successfully");

});

return res.render('message');

})

app.get('/message',function(req,res){

    res.sendFile( path.join(__dirname + '/index.html'));

})

app.get('/',function(req,res){

    res.sendFile( path.join(__dirname + '/index.html'));

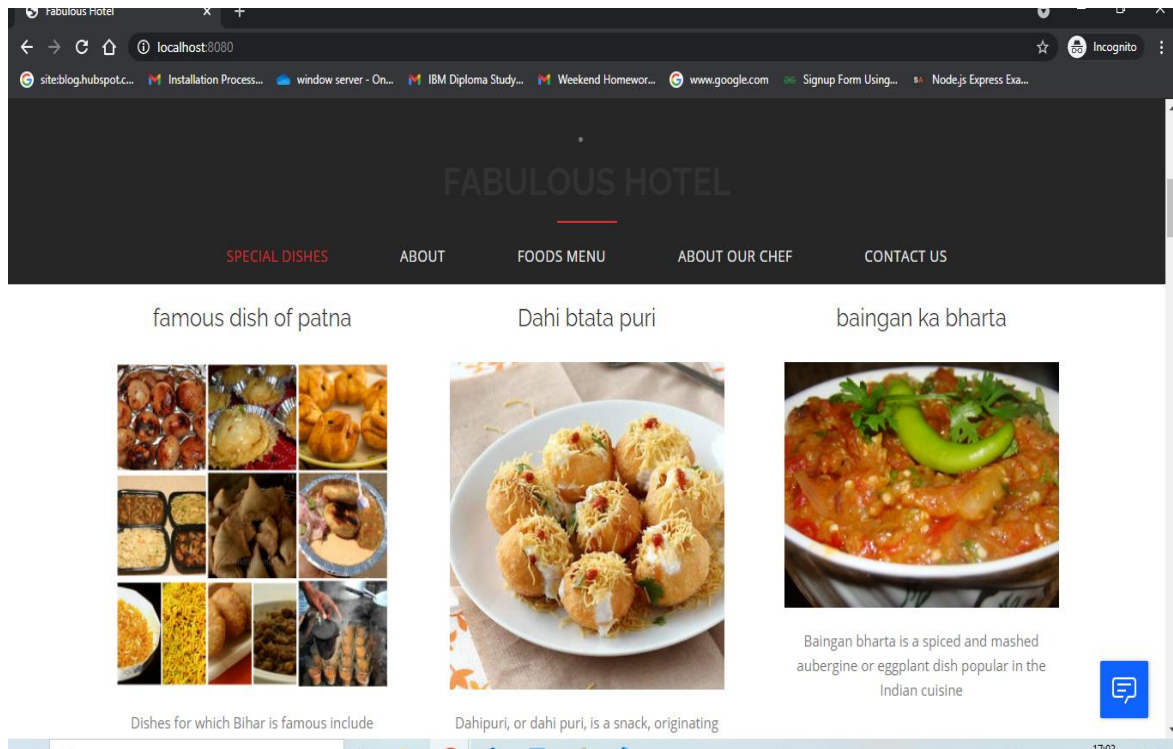
}).listen(8080)
```

Package.json

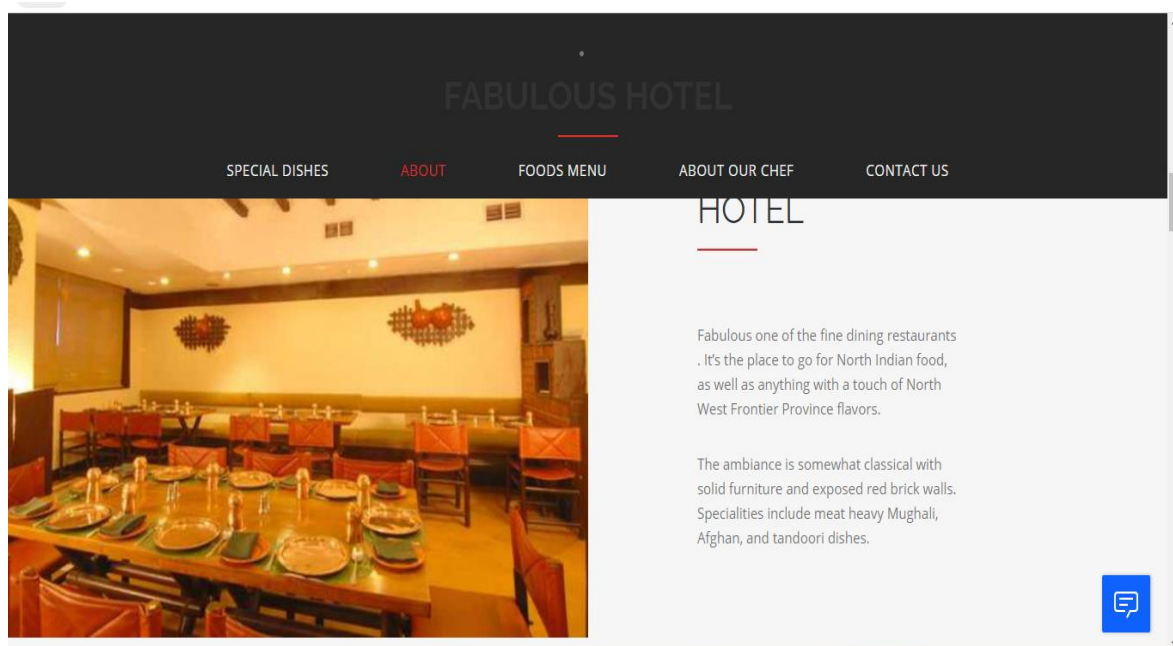
```
{  
  "name": "my-project",  
  "version": "1.0.0",  
  "main": "app.js",  
  "dependencies": {  
    "body-parser": "^1.19.0",  
    "ejs": "^3.1.6",  
    "express": "^4.17.1",  
    "mongodb": "^3.6.6",  
    "mongoose": "^5.12.14"  
  },  
  "devDependencies": {},  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "description": ""  
}
```


Screenshots

Home page



About page



food menu page

FABULOUS HOTEL

SPECIAL DISHES

ABOUT

FOODS MENU

ABOUT OUR CHEF

CONTACT US

DINNER

Caprese Pasta Salad.

250Rs.

cups cherry tomatoes, halved. 8 oz bocconcini fresh mozzarella balls (or fresh mozzarella cut into 1/2" cubes) 12 oz dried orecchiette pasta

Malai Kofta

170Rs.

Malai Kofta is a very popular Indian vegetarian dish where balls (kofta) made of potato and paneer are deep fried and served with a creamy and spiced tomato based curry. The literal translation of this dish is malai=creamy/buttery and kofta=spiced balls (in this case made of potatoes and paneer).

DESSERTS

Chocolate Mud Cake

410Rs.

Deliciously moist and rich, this is definitely THE best ever cake for chocolate lovers. One piece is never enough!

Bourbon-Pecan Tart

240Rs.

Bourbon Pecan Tart ... Gooley and sticky sweet, pecan pie is the quintessential Southern dessert. Here it is transformed into a buttery tart that requires less sugar

Texas Sheet Cake

500Rs.

A moist and delicious chocolate cake made in a jellyroll pan,

FABULOUS HOTEL



SPECIAL DISHES



ABOUT

FOODS MENU

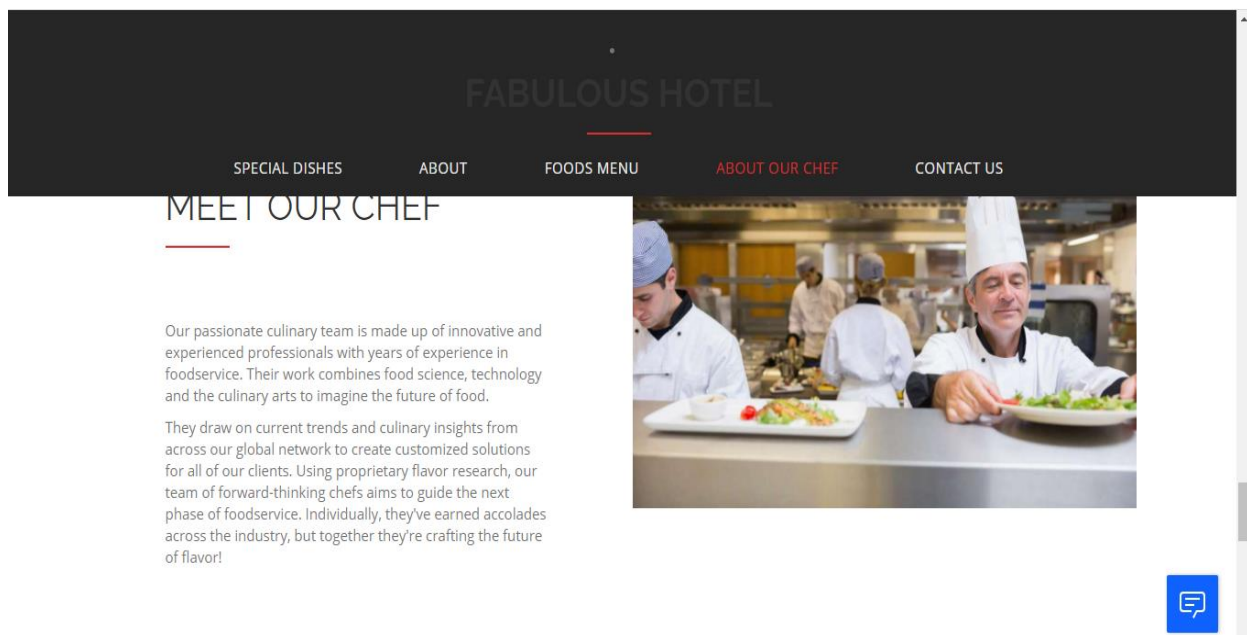
ABOUT OUR CHEF

CONTACT US

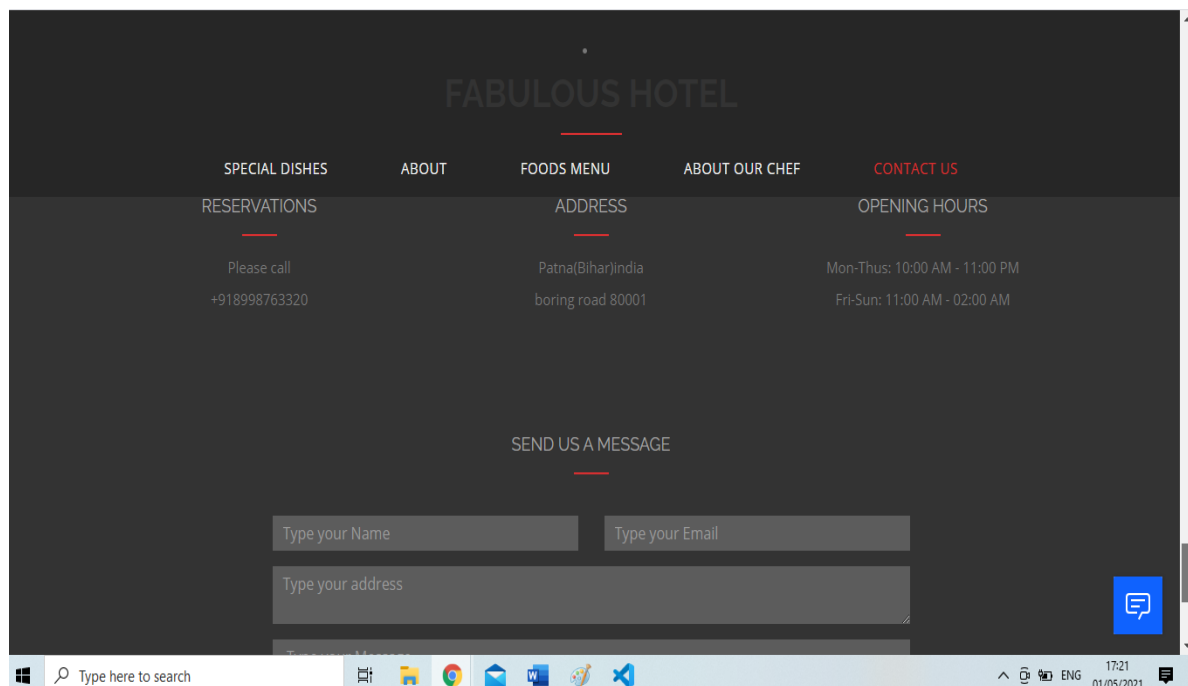



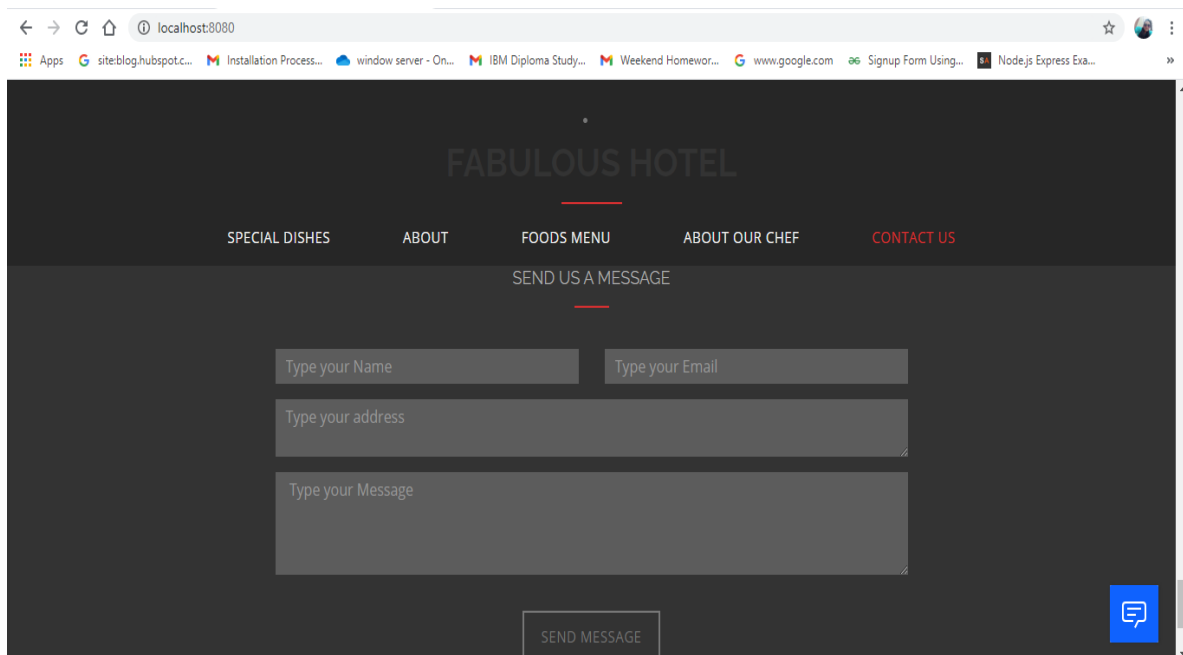
our chef page page



contact us page

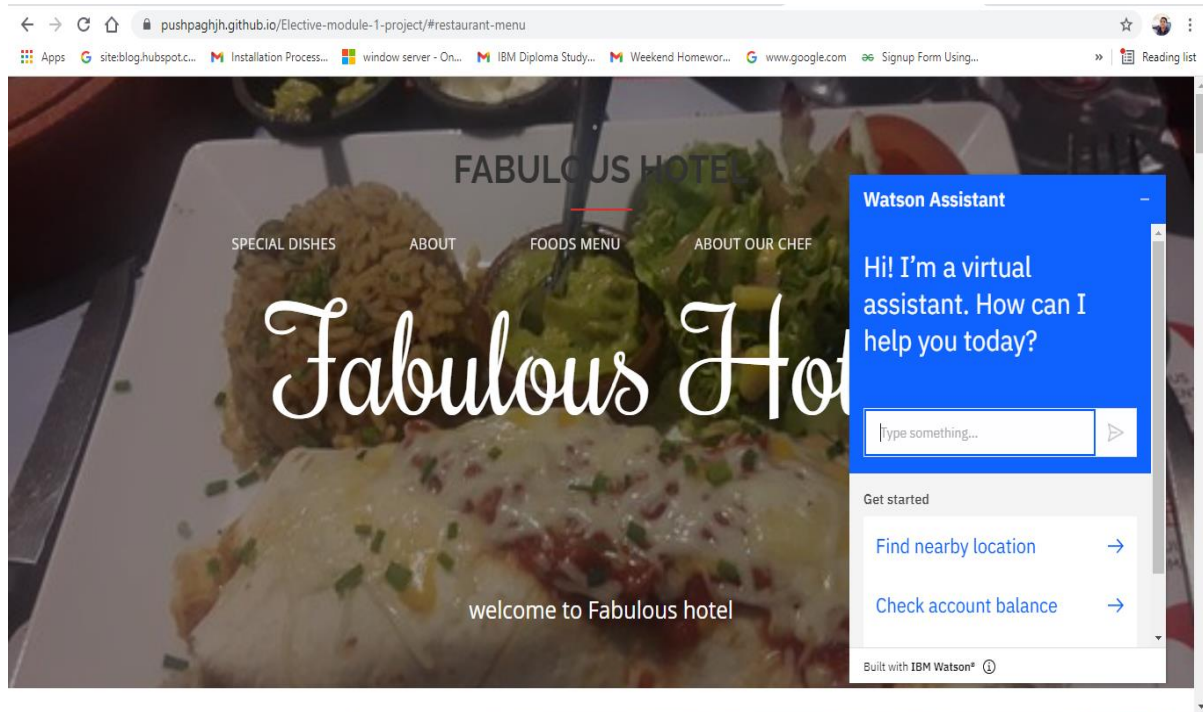


In this field you can send your message



The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The browser's tab bar includes several open tabs. The main content area of the browser shows a dark-themed website for 'FABULOUS HOTEL'. The website has a navigation menu with links: 'SPECIAL DISHES', 'ABOUT', 'FOODS MENU', 'ABOUT OUR CHEF', and 'CONTACT US'. Below the navigation menu, there is a section titled 'SEND US A MESSAGE'. This section contains four input fields: 'Type your Name', 'Type your Email', 'Type your address', and 'Type your Message'. At the bottom of this section is a 'SEND MESSAGE' button. A blue chat icon is visible in the bottom right corner of the website.

Waston assistant



The screenshot shows a web browser window with the address bar displaying 'pushpaghj.github.io/Elective-module-1-project/#restaurant-menu'. The browser's tab bar includes several open tabs. The main content area of the browser shows a restaurant menu for 'FABULOUS HOTEL'. The menu features a background image of a plate of food. The text 'FABULOUS HOTEL' is displayed at the top, and 'welcome to Fabulous hotel' is at the bottom. The menu items are 'SPECIAL DISHES', 'ABOUT', 'FOODS MENU', and 'ABOUT OUR CHEF'. A large, stylized 'Fabulous Hotel' logo is overlaid on the food image. A blue chatbot overlay titled 'Watson Assistant' is positioned on the right side of the screen. The chatbot displays the message 'Hi! I'm a virtual assistant. How can I help you today?' and includes a text input field with the placeholder 'Type something...'. Below the input field, there are two buttons: 'Find nearby location' and 'Check account balance', both with right-pointing arrows. At the bottom of the chatbot, it says 'Built with IBM Watson®'.

Conclusion

Hotel management system now-a-day have the advantage of modernization. Computer have done the work more easy. Computer is playing a important role in management. Reports are made on daily basis for every customer check in or check out which can easily be seen by the management. Hotel management system has also primary purpose is to provide facilities to customers. A software for computers makes the things many times easy, these are made as user friendly and to keep an check and balance in hotel management and accounts as well. So ,these things are important.

References

<https://www.mongodb.com/>

<https://cobusgreyling.medium.com/how-to-use-an-actions-skill-in-ibm-watson-assistant-4f05e6fd86b8>

<https://www.mongodb.com/blog/post/quick-start-nodejs-mongodb--how-to-get-connected-to-your-data-base>

THANK YOU...