



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

Mr. Boddu Lingaiah  
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, for his 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, Saltlake 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 to Mr. Boddu Lingaiah faculty of NSTI Saltlake Kolkata for supporting me for course completion in the specific subject.

I am also grateful to the following NSTI Regional Director Sir Mr. Sarbojit Neogi and Assistant Director Sir Mr. K.L Kuli for their consistent support and assistance.

## **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	<a href="#">5</a>
2	Chapter 2: Services and Tools Required	6-17
3	Chapter 3: Project Architecture	18
4	Chapter 4: Architecture Blocks Detail Working	19
5	Chapter 5: create a webpage using node and add data with mongo DB Atlas	21-38
6	Source Code	39-74
7	Conclusion	75
	References	76
8	Screenshot	77-80

# CHAPTER 1

## INTRODUCTION

1. overview
2. *Feature*
3. *Advantages*
4. *Scope*
5. *Future Work*

# **INTRODUCTION**

## **1. Overview**

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

And waston assistant assist the user's what food available here or not available and anything according to users need and ask the questions waston 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 waston 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

S.I No.	Topics	Page No
1	Services Used	7-14
a	Waston Assistant	7
b	What is actions in waston	7
c	Create a waston assistant using actions method	7-13
d	how to add html page a waston	14
2	Used technology	16-17

# 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 in IBM Watson Assistant?

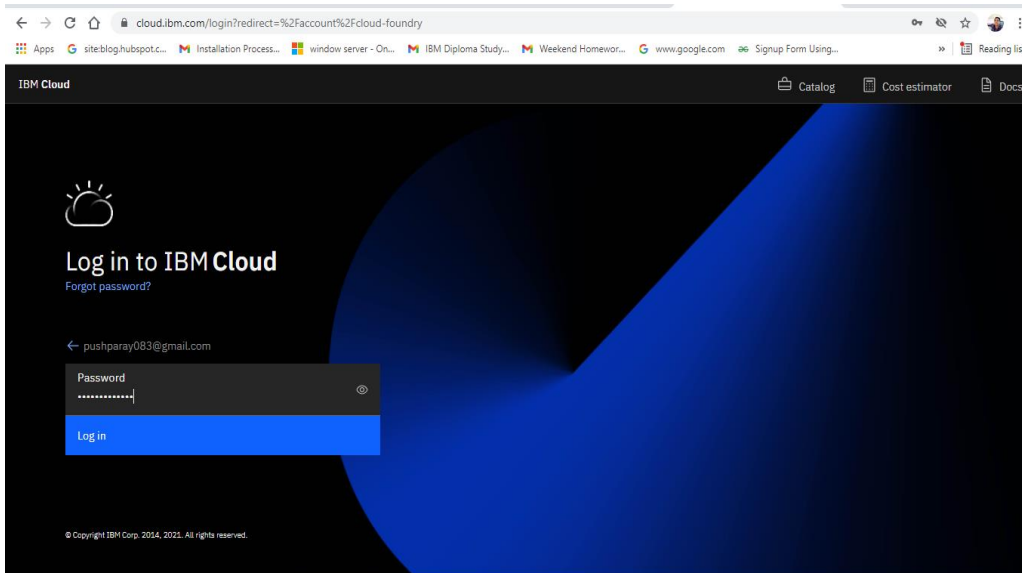
A new feature has been added to IBM Watson Assistant called Actions. This new feature allows users to develop dialogs in a rapid fashion.

### c) Build a Watson chatbot using actions

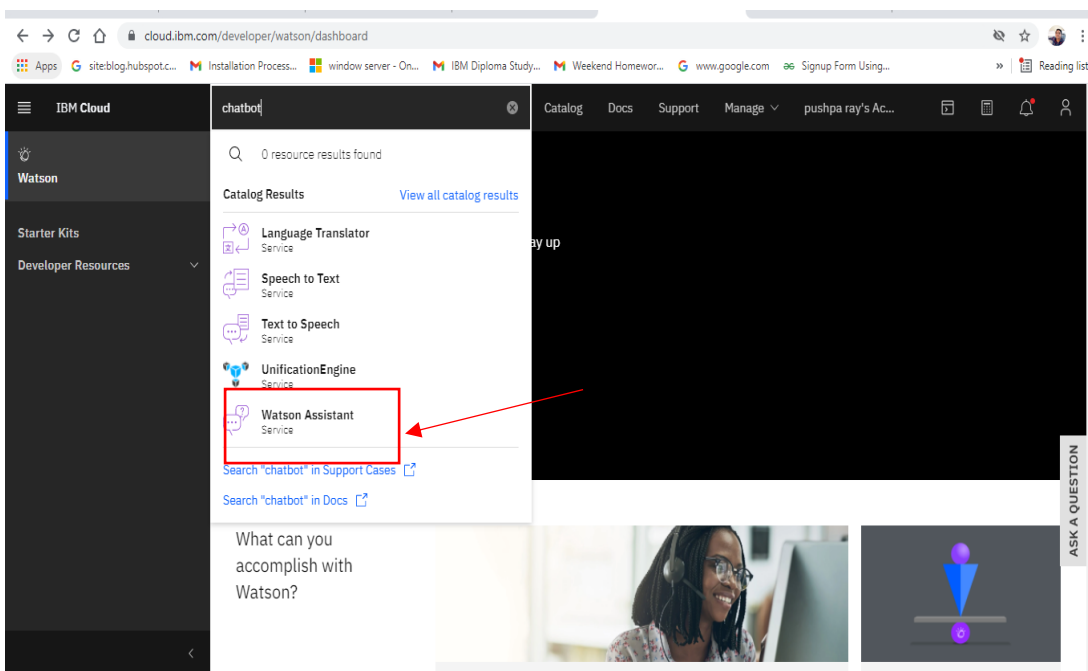
#### Step 1

- 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 catalog page for Watson Assistant. The page is titled "Watson Assistant" and includes a "Create" button. Below the title, there is a "Select a location" dropdown menu with "London (eu-gb)" selected. A "Select a pricing plan" section follows, with a table showing the "Lite" plan as "Free". A "Summary" sidebar on the right lists details: Location: London, Plan: Lite, Service name: Watson Assistant-5y, and Resource group: Default. A "Create" button is visible in the sidebar.

IBM Cloud

Search resources and offerings...

Catalog Docs Support Manage

pushpa ray's Ac...

Reading list

Catalog / Services /

Watson Assistant

IBM • Date of last update: 06/16/2021 • Docs • API docs

Create About

Select a location

Select a location

London (eu-gb)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

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

Create

Add to estimate

View terms

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

The screenshot shows the IBM Cloud resource page for "Watson Assistant-5y". The page is titled "Watson Assistant-5y" and includes a "Launch Watson Assistant" button. Below the title, there is a "Start by launching the tool" section with a "Launch Watson Assistant" button. A "Credentials" section shows the API key and URL. A red arrow points from the "Launch Watson Assistant" button in the "Start by launching the tool" section to the "Launch Watson Assistant" button in the "Credentials" section.

Resource list /

Watson Assistant-5y

Active Add tags

Details Actions...

Manage

Service credentials

Plan

Connections

Start by launching the tool

Launch Watson Assistant

Getting started tutorial

API reference

Plan

Lite

Upgrade

Credentials

Download Show credentials

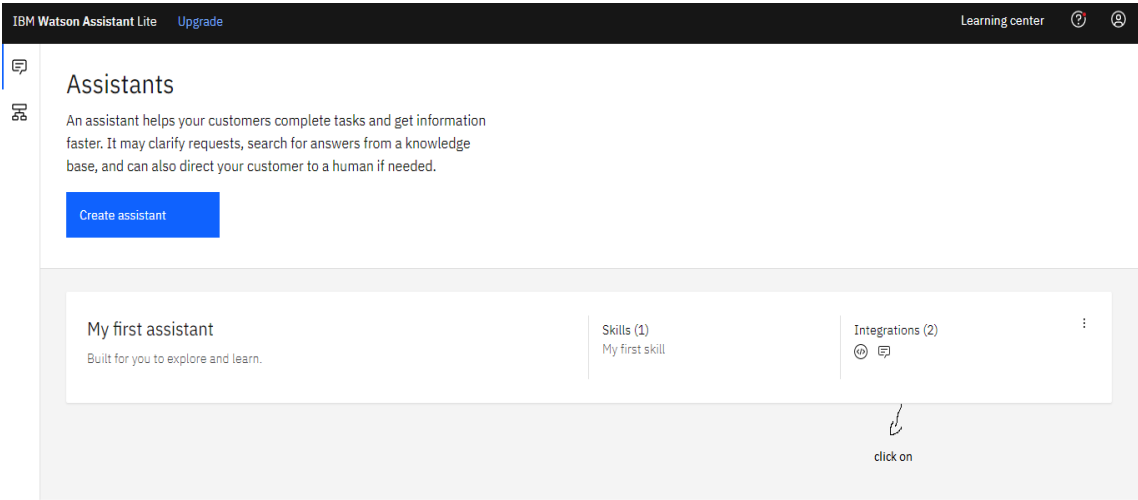
API key:

URL:

<https://api.eu-gb.assistant.watson.cloud.ibm.com/instances/>

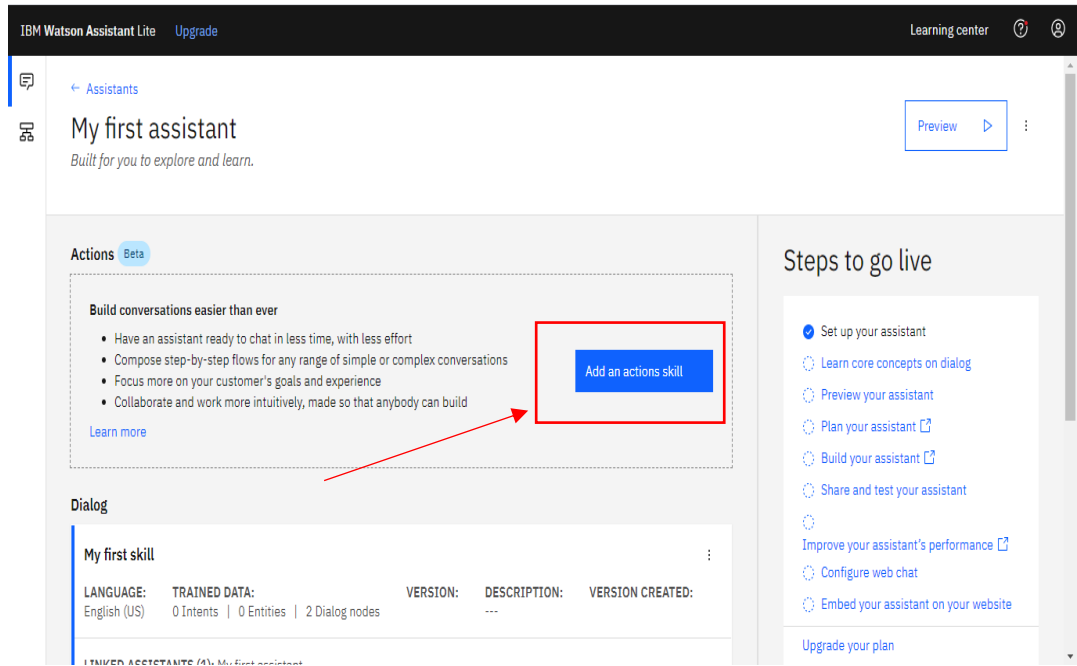
# Step 2

Click on integration

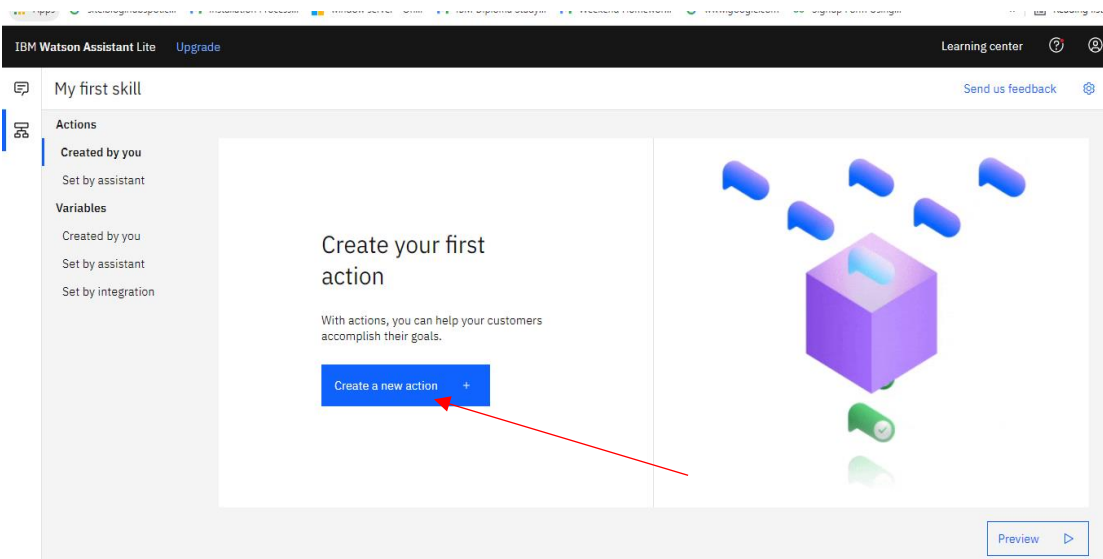


After that click on

That indicated to image

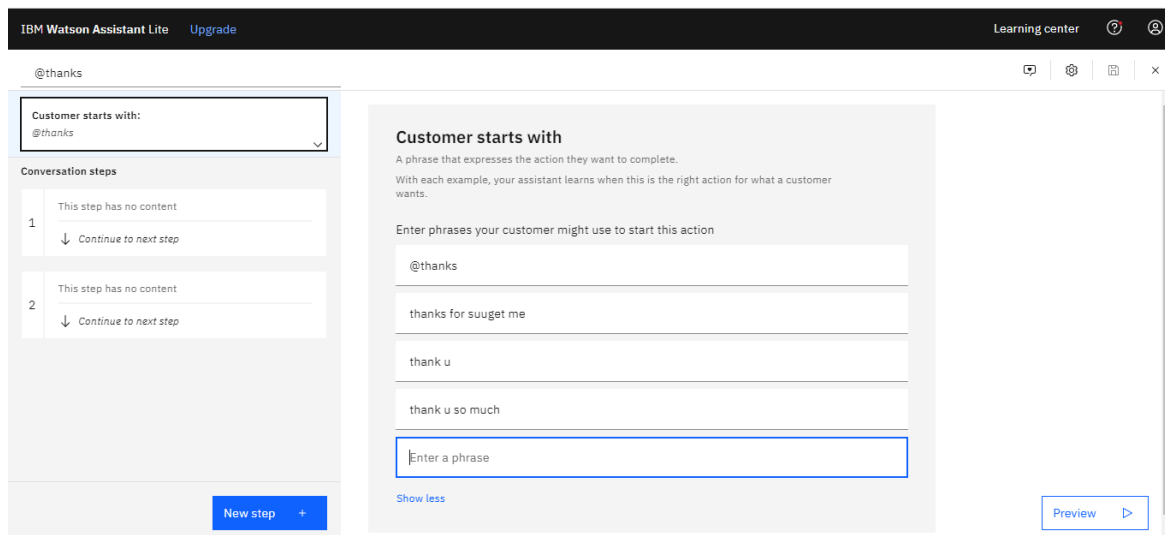


## Click on create a new action



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 LiteUpgrade

Learning center

waston assistantBeta

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: 50Showing 1-7 of 7 actions

11 of 1 pages

Preview

Step 3

A. How to check chatbot properly run or not

Click on preview

IBM Watson Assistant LiteUpgrade

Learning center

waston assistantBeta

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		

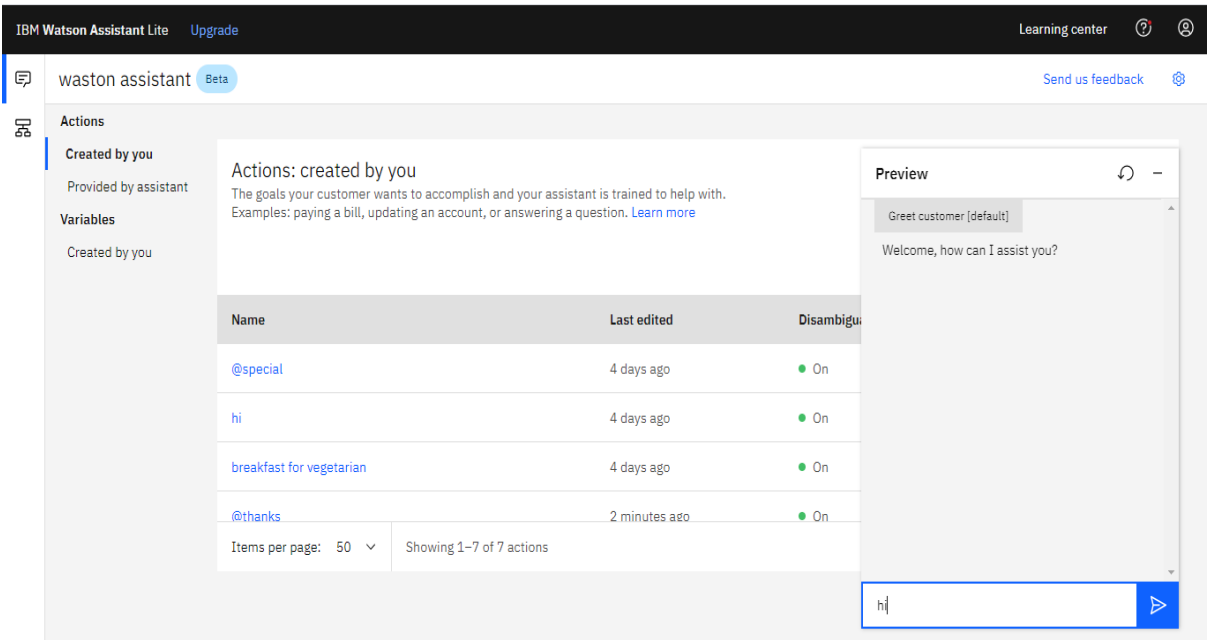
Items per page: 50Showing 1-7 of 7 actions

11 of 1 pages

Preview

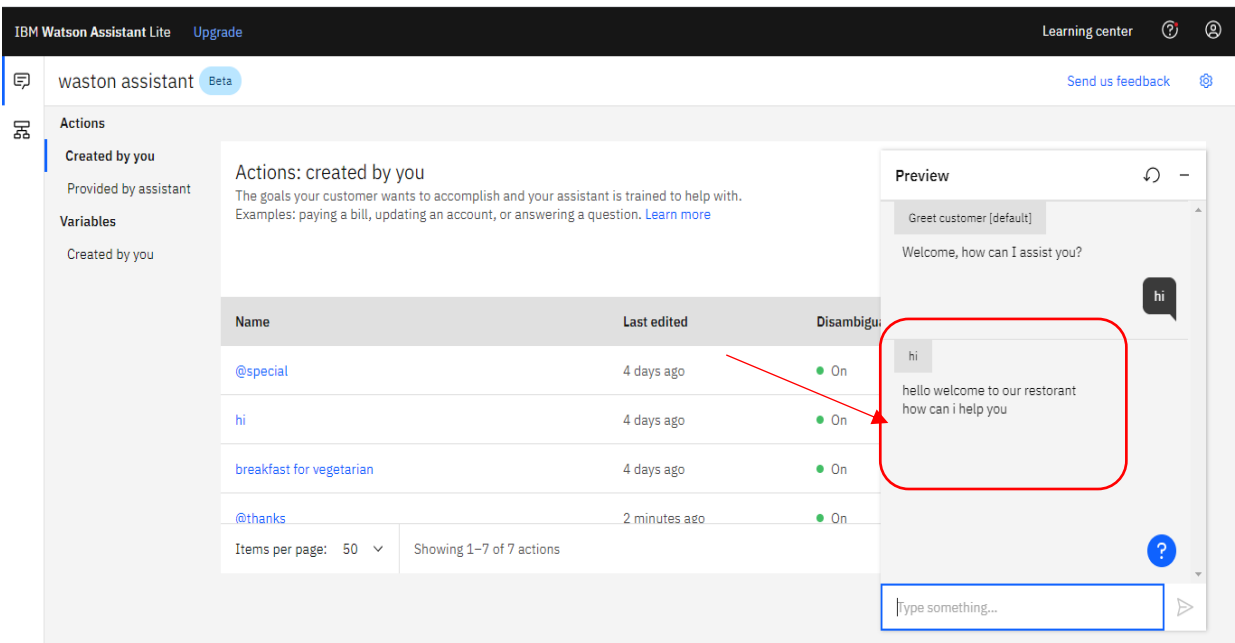
Write the user question

I have provided hi in that chatbot



You can see in image chatbot give the user response

That means our chatbot successfully created



## Step 4

### How to add waston assistant in a static webpage

Go to web chat

IBM Watson Assistant Lite Upgrade Learning center ?

← Assistants

### My first assistant

Built for you to explore and learn.

Preview ▶

**Actions**

**My first skill**

LANGUAGE	VERSION	CREATED	UPDATED
English (US)	Development	Jun 23, 2021 2:47 PM IST	Jun 23, 2021 2:47 PM IST

LINKED ASSISTANTS (1): My first assistant

**Search** Plus

**Turn any content into answers**

- Create Q&A experiences in minutes
- Sync with websites and data sources for always up-to-date answers
- Handle even complex questions with inclusive, contextual responses

Try Plus plan

Watch a brief demonstration

**Integrations**

Web chat

Choose a channel to deploy your assistant. Add integration

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

IBM Watson Assistant Lite Upgrade Learning center ?

### Web chat

Close Saved

**Style** Home screen Live agent Suggestions Security Embed

**Customize your chat UI**

Update the style to match your brand and your website. A developer can also add more advanced styling changes with code. [Learn more](#)

Assistant's name as known by customers

Watson Assistant

Primary color Secondary color

Chat header User message bubble

#FFFFFF #3D3D3D

Accent color

Significant and interactive

Restart conversation ↺

**Watson Assistant**

Hi! I'm a virtual assistant. How can I help you today?

Type something...

## 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

### Used Technology

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

### Run time Environment

Node.js

### Database

Mongo dB compass



# 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.

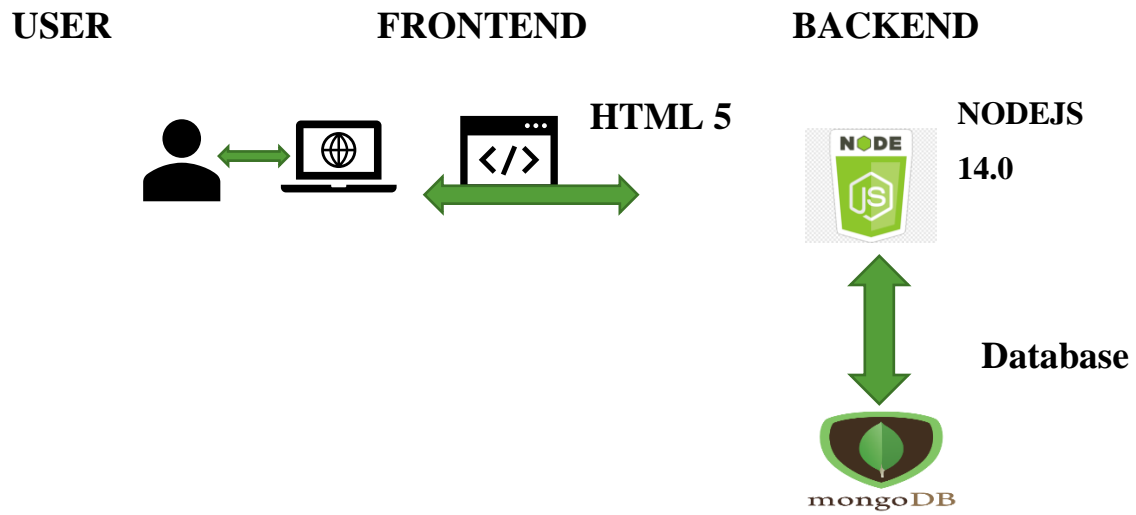
## VII. Mongo dB Atlas

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

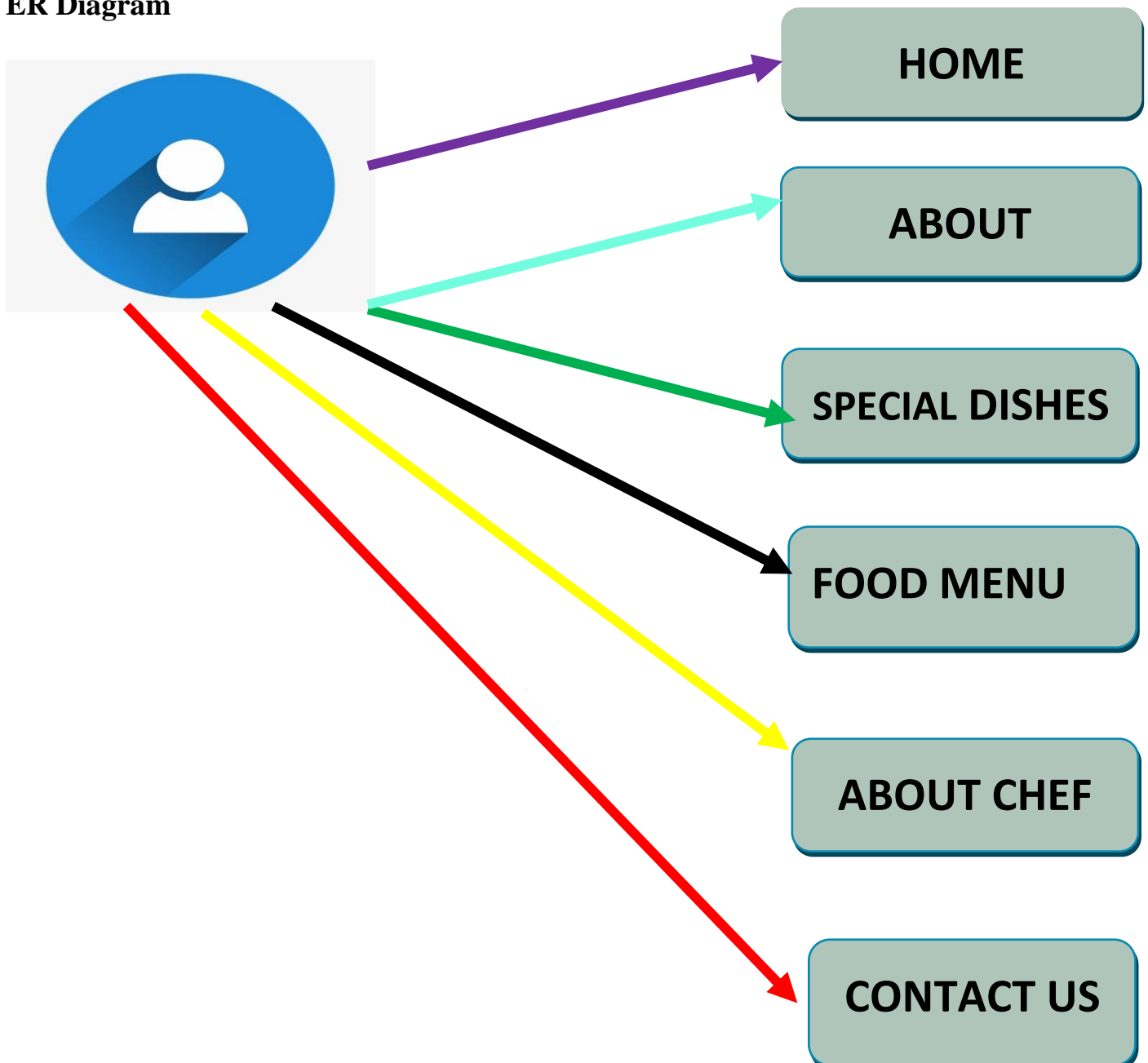


## CHAPTER 4

### ARCHITECTURE BLOCKS DETAIL WORKING

#### 4.1 Blocks

##### ER Diagram



## CHAPTER 5

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

<b><i>S.I No</i></b>	<b><i>Topics</i></b>	<b><i>Page No</i></b>
<i>1.a</i>	Creating files	<i>22</i>
<i>b</i>	Install <i>dependencies</i> modules for <i>NodeJS</i> server	<i>23</i>
<i>2</i>	<i>Connect our data with mongo dB Atlas</i>	<i>24-38</i>
<i>a</i>	Create an account in mongo dB	<i>25-27</i>
<i>b</i>	<i>Account setup</i>	<i>28</i>
<i>c</i>	<i>Create cluster</i>	<i>29-30</i>
<i>d</i>	<i>Add data with mongo dB database</i>	<i>30-38</i>

## A. Creating files

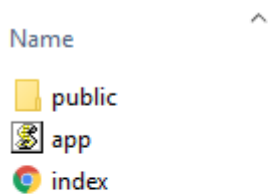
Create folder

I have created a project folder

Inside project folder I have created those 3 files & folder

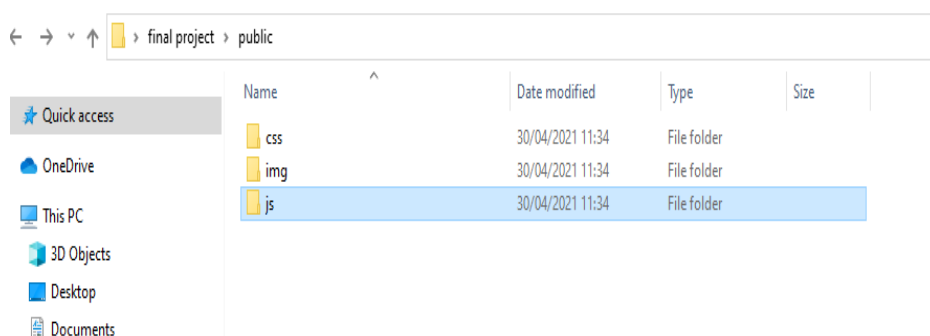
1. index.html
2. Public
3. App.js

You can see in image those folders



Public folder

You can see in image



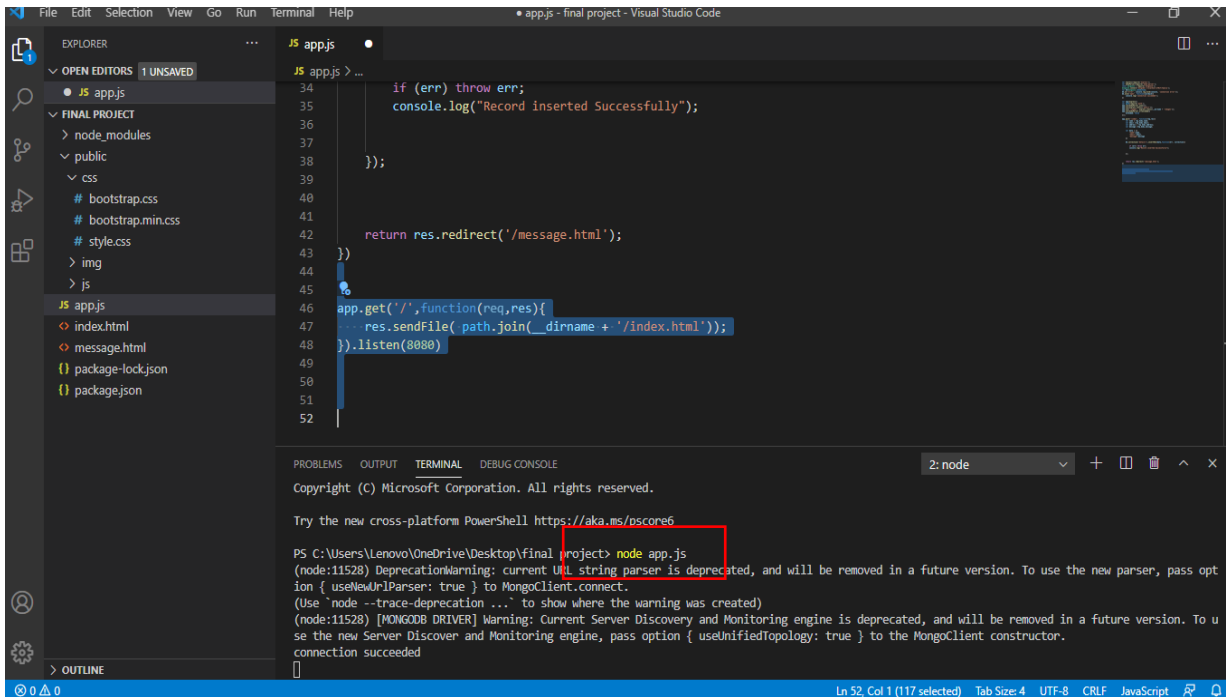
## B. Install dependencies modules

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 displays the project structure, including files like `app.js`, `index.html`, `message.html`, `package-lock.json`, and `package.json`. The main editor area shows the content of `app.js`, which includes a route for `/message.html` and a default route for `/` that serves `index.html`. The terminal at the bottom shows the command `node app.js` being executed, with a red box highlighting the command. The terminal output includes a deprecation warning about the URL string parser and a message from the MongoDB driver, followed by the text "connection succeeded".

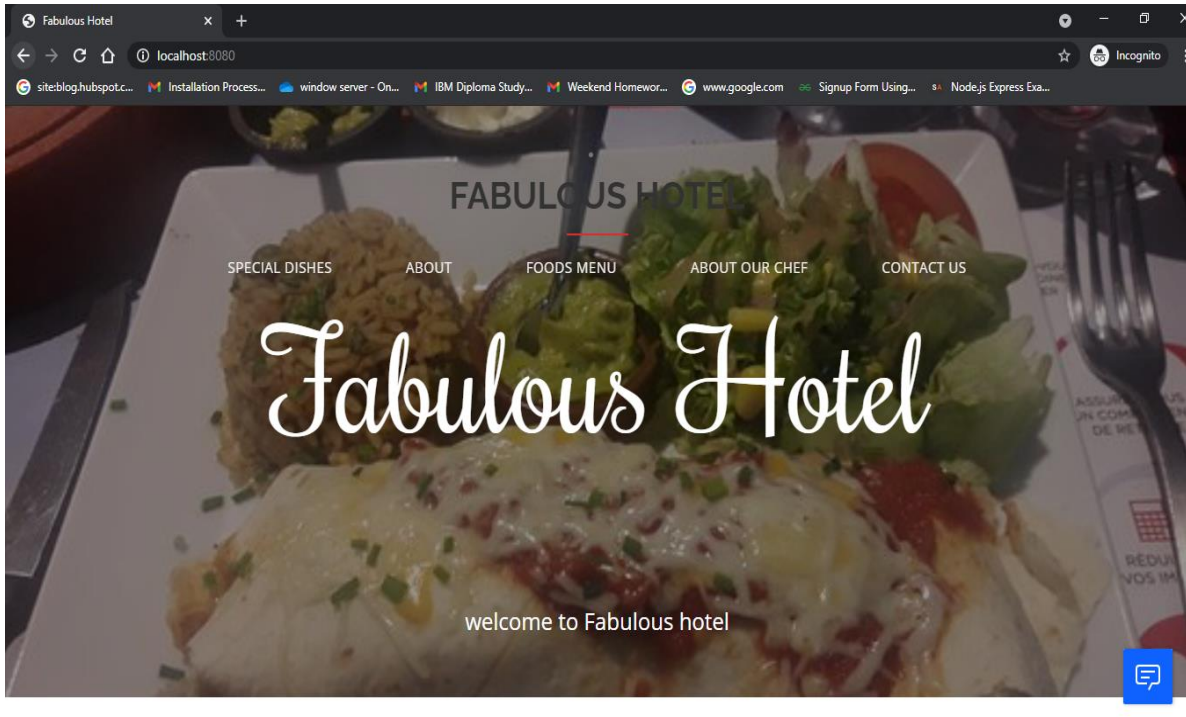
## C. Run application NodeJS server

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

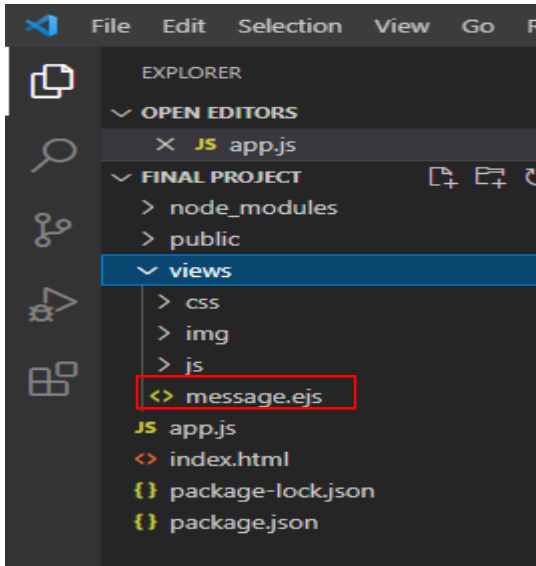
### 1. install dependencies modules

- 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
- ✓ Such as You see in image





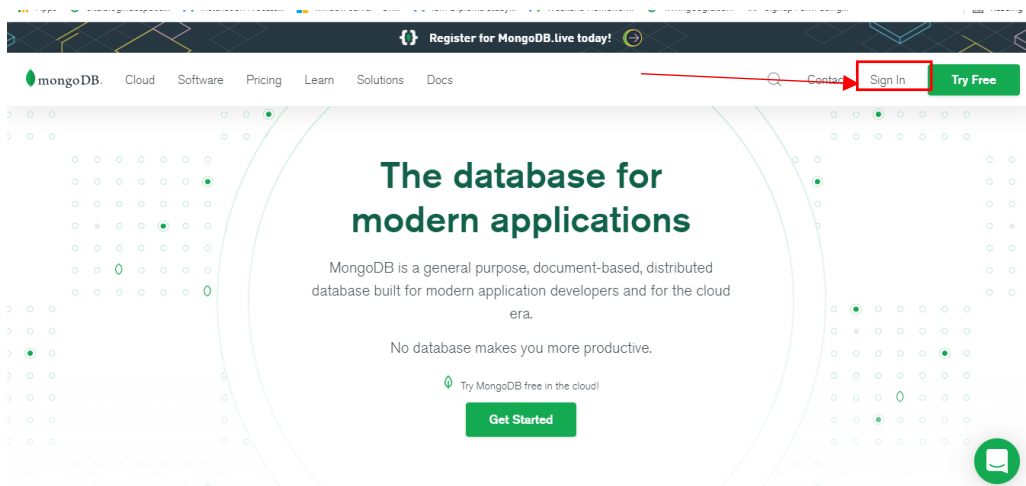
## 2. create an account in mongo dB Atlas

### step 1

go this link

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

click on sign in button



Click on sign up

The screenshot shows the MongoDB login page. On the left, there is a 'Log in to your account' section with a 'Log in with Google' button, an 'Email Address' input field, and a 'Next' button. A red box highlights the 'Sign Up' link next to the 'Next' button. On the right, there is a large banner for 'MongoDB.live!' with the text 'Register now for MongoDB.live!' and 'Hear what's new with MongoDB and deep dive into the future of data with dozens of sessions, live panels, and direct access to experts.' The banner also includes a 'Learn more' link and a 'Sign Up' button.

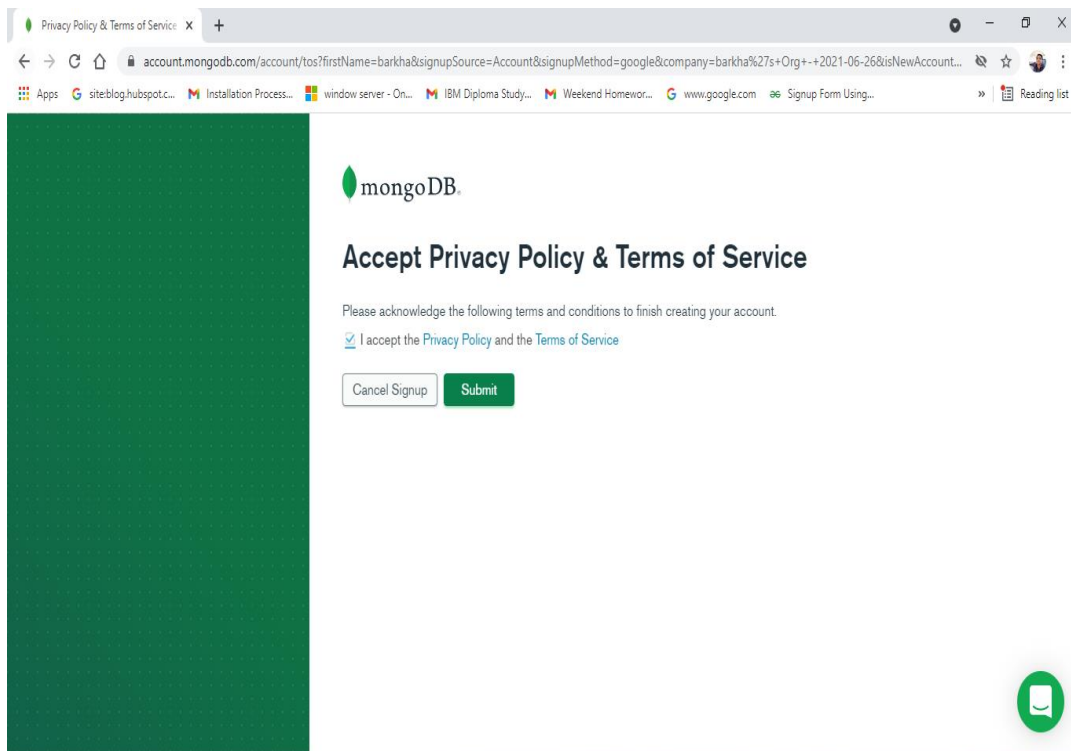
Click on sign up with google

You select your email id

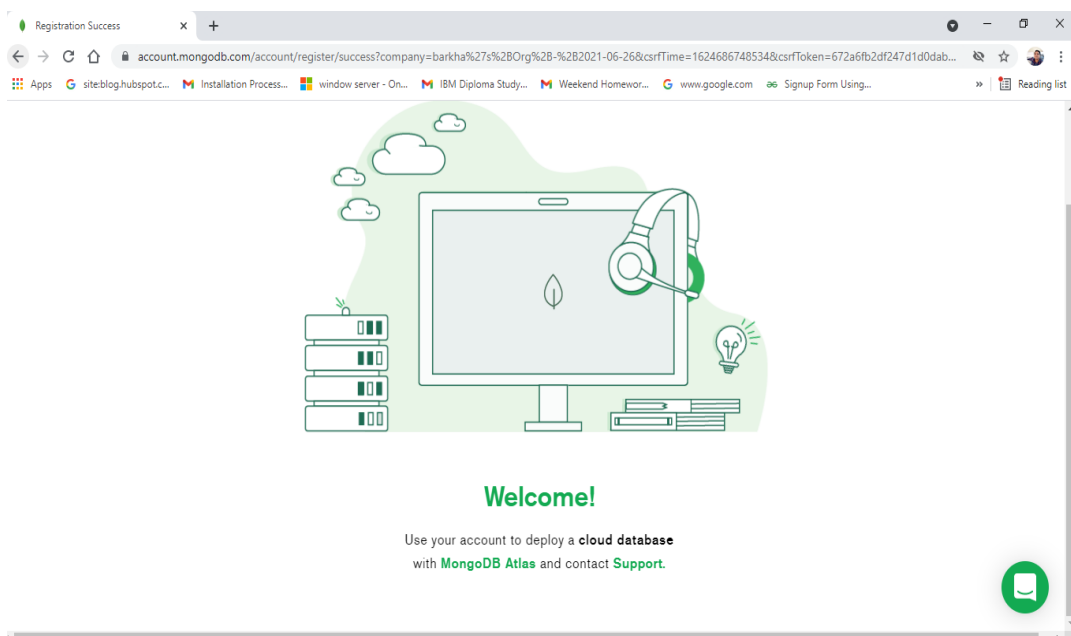
The screenshot shows the MongoDB registration page. On the left, there is a large green vertical bar. The main content area has the title 'Create your account' and a 'Sign up with Google' button, which is highlighted by a red box. Below this, there are input fields for 'Email Address', 'First Name', 'Last Name', and 'Password'. The 'Email Address' field has a note: 'We recommend using your work email'.

## Accept term & condition

Check the button click on submit



## You can create my account



# D.Account Setup

Setup your account or go ahead individual

Atlas Onboarding | MongoDB x mongodcloud\_new - Google S... +

cloud.mongodb.com/v2/60d6c0a13681da46abcc1a77#setup/onboarding

Apps siteblog.hubspot... Installation Process... window server - On... IBM Diploma Study... Weekend Homewor... www.google.com Signup Form Using... Reading list

MONGODB ATLAS

Let's get your account set up

**Name your organization and project**

**Organization**  
Your organization can be a business, team, or an individual

mongodbcloud\_new

**Project Name**  
Use projects to isolate different environments (development/testing/production)

Project 0

What is your preferred language?

Select java language

Click on continue

Atlas Onboarding | MongoDB x mongodcloud\_new - Google S... +

cloud.mongodb.com/v2/60d6c0a13681da46abcc1a77#setup/onboarding

Apps siteblog.hubspot... Installation Process... window server - On... IBM Diploma Study... Weekend Homewor... www.google.com Signup Form Using... Reading list

Project Name

Use projects to isolate different environments (development/testing/production)

Project 0

**What is your preferred language?**  
We'll use this to customize code samples and content we share with you. You can always change this later.

JavaScript C++ C# / .NET Go

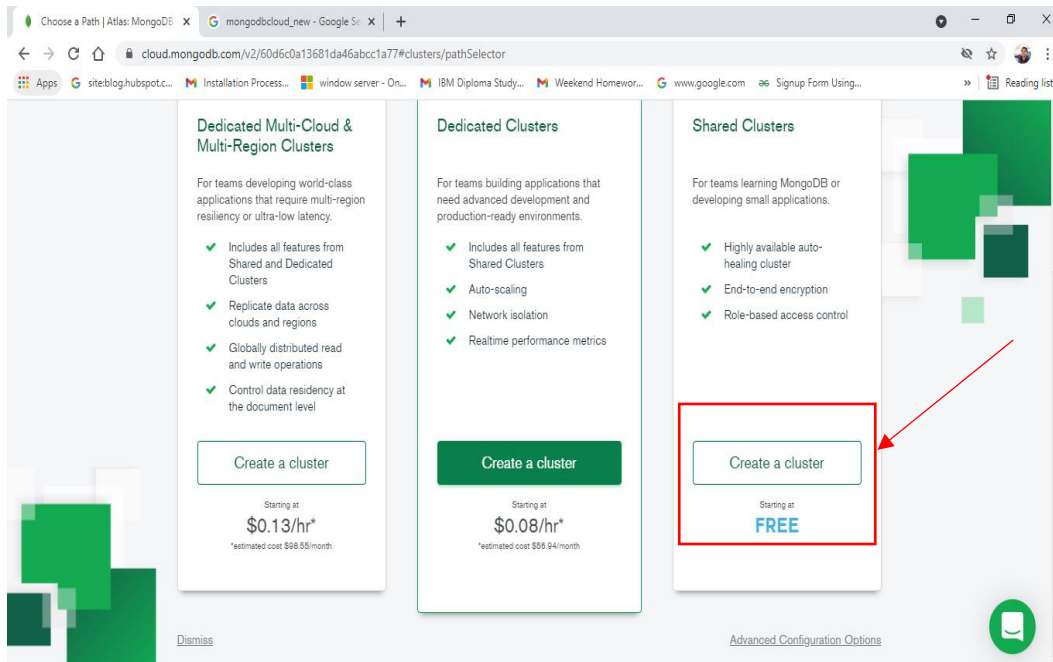
Java C Perl PHP

Python Ruby Scala Other

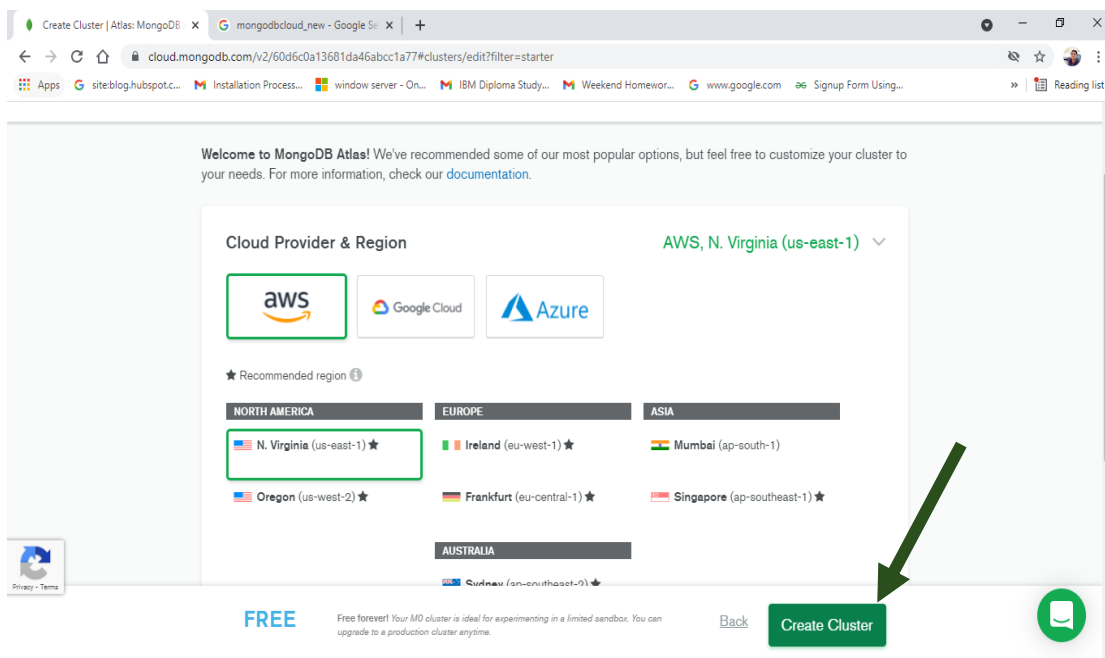
Skip Continue

## E. Create a cluster

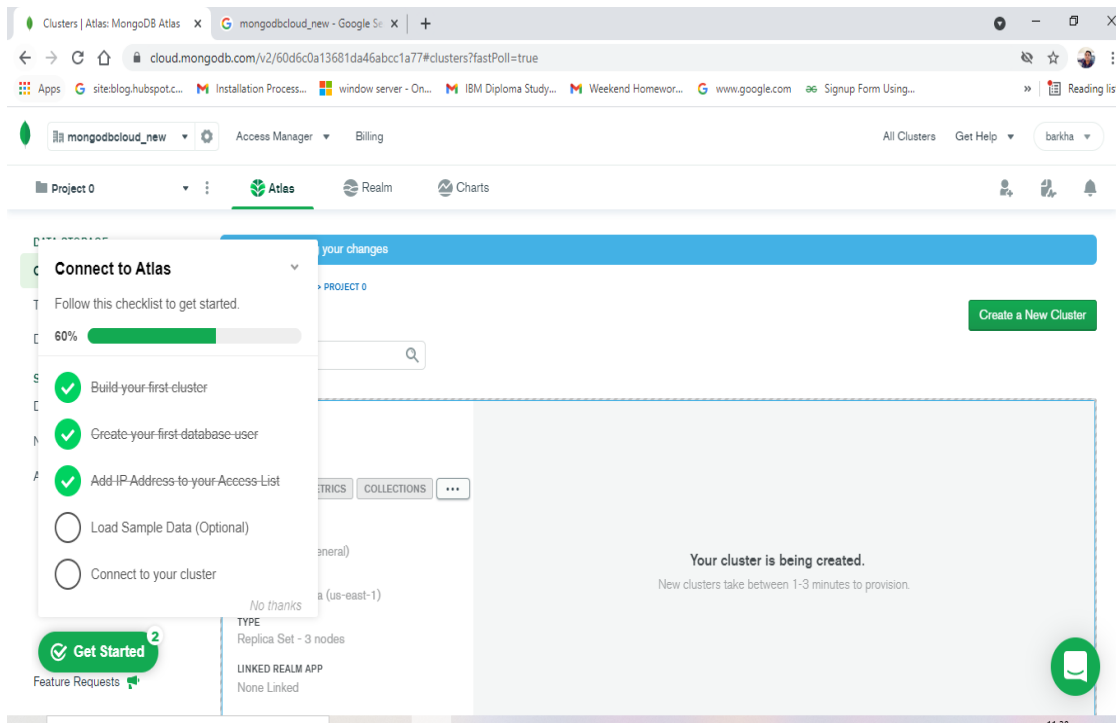
### Select free plan



### Click on create cluster

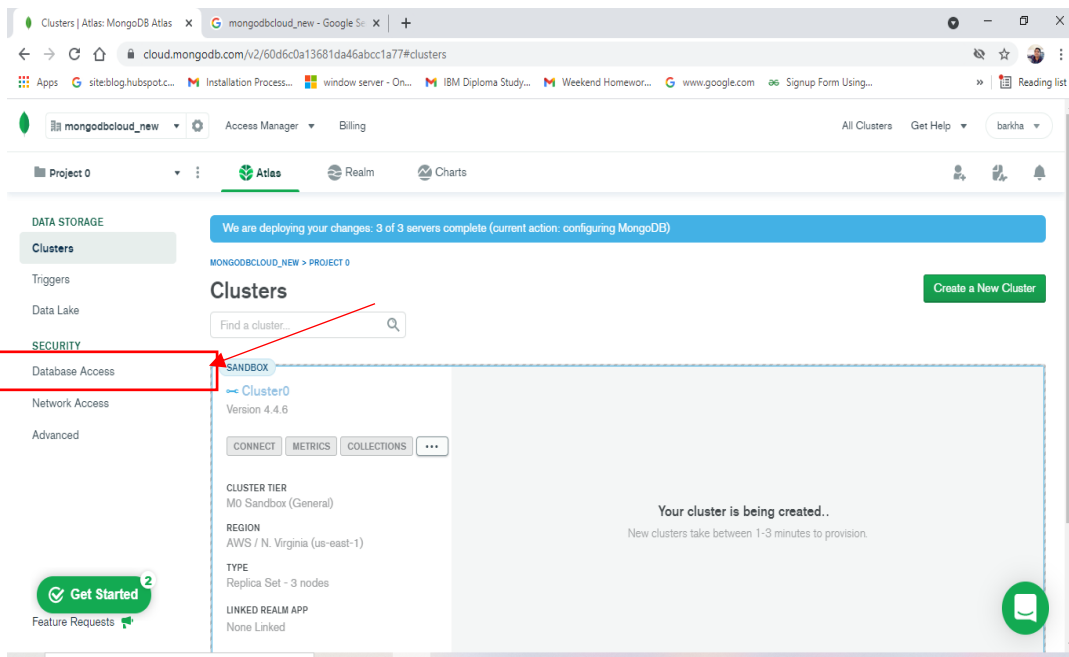


You can see open this page

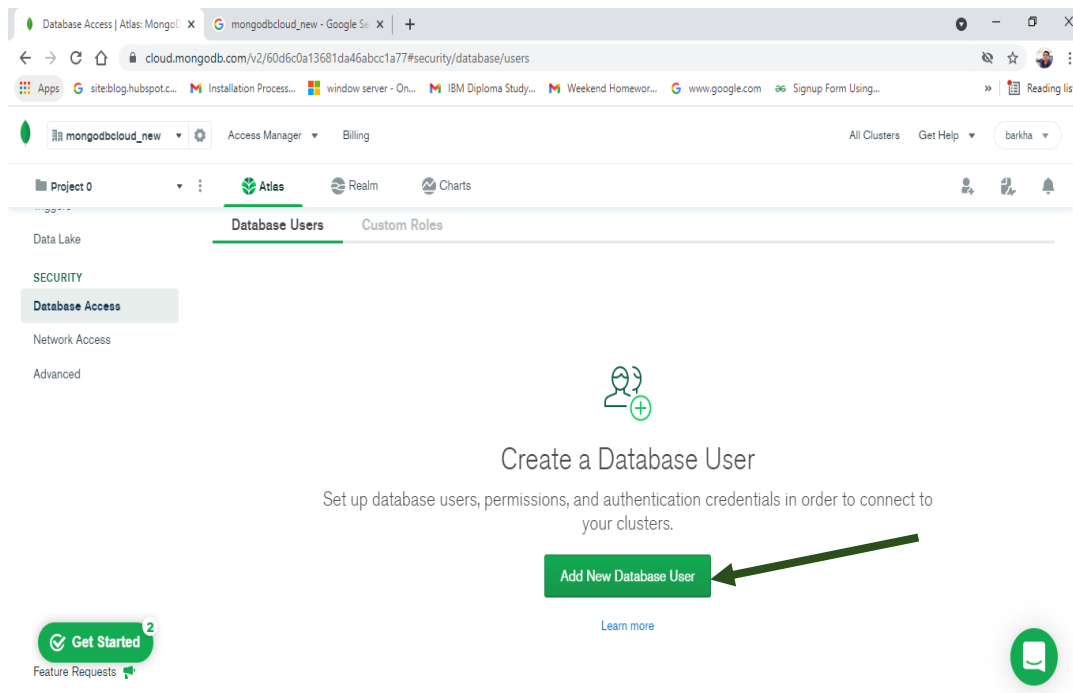


## D.Add Data with Mongo dB

Click on data access

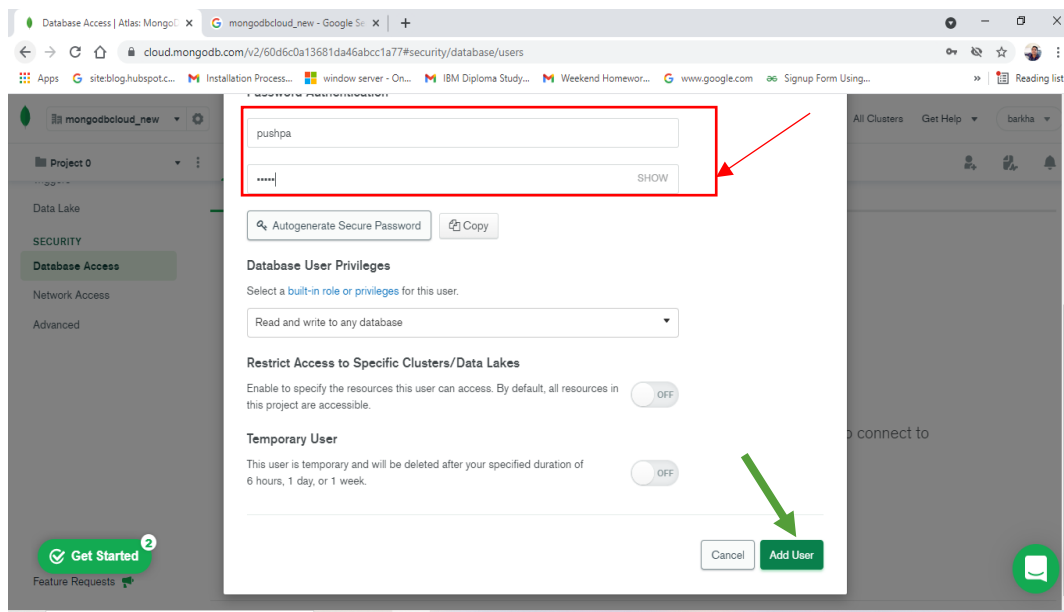


Click on add new user



Give the user's name & password

Click on add user



You able to see added your user

Database Access | Atlas: MongoD... x mongodcloud\_new - Google S... x +

cloud.mongodb.com/v2/60d6c0a13681da46abcc1a77#security/database/users

mongodcloud\_new Access Manager Billing All Clusters Get Help barkha

Project 0 Atlas Realm Charts

DATA STORAGE

Clusters

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

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

MONGODCLOUD\_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 2

System Status: All Good

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

Click on network access

Add Ip address

Network Access | Atlas: MongoD... x mongodcloud\_new - Google S... x +

cloud.mongodb.com/v2/60d6c0a13681da46abcc1a77#security/network/accesslist

mongodcloud\_new Access Manager Billing All Clusters Get Help barkha

Project 0 Atlas Realm Charts

DATA STORAGE

Clusters

Triggers

Data Lake

SECURITY

Database Access

Network Access

Advanced

Network Access

IP Access List Peering Private Endpoint

Add an IP address

Configure which IP addresses can access your cluster.

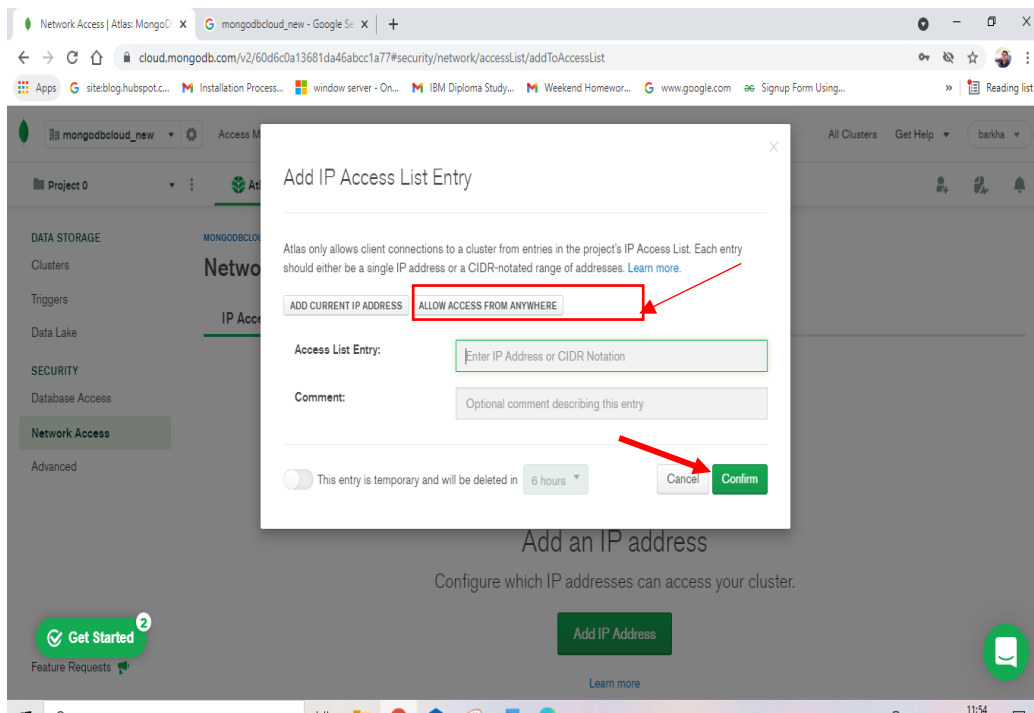
Add IP Address

Learn more

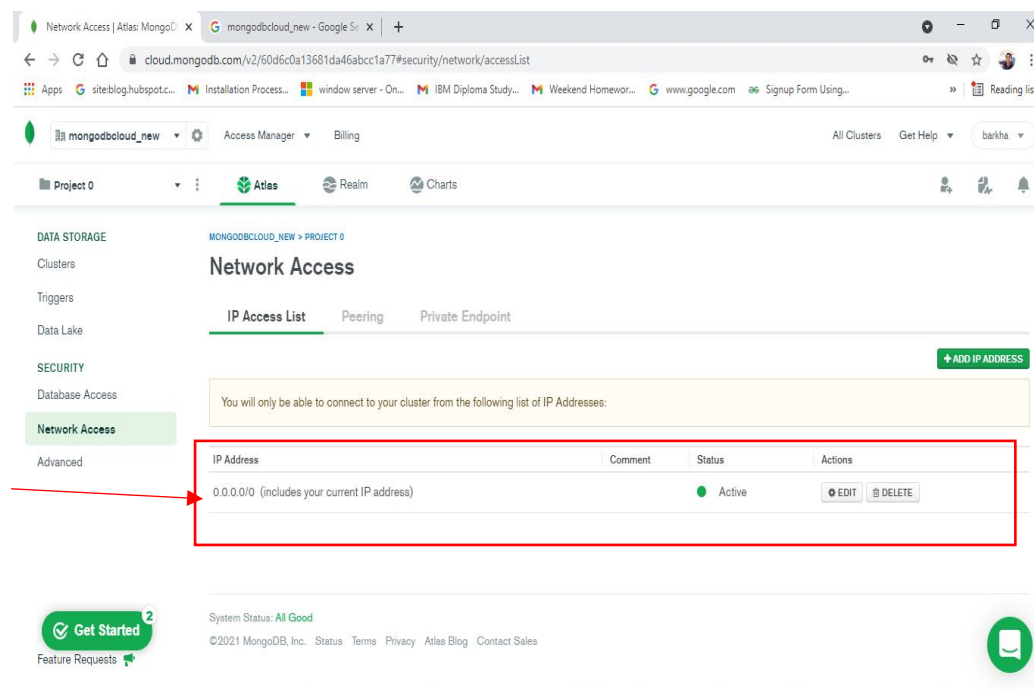


Select allow anywhere

Click on confirm

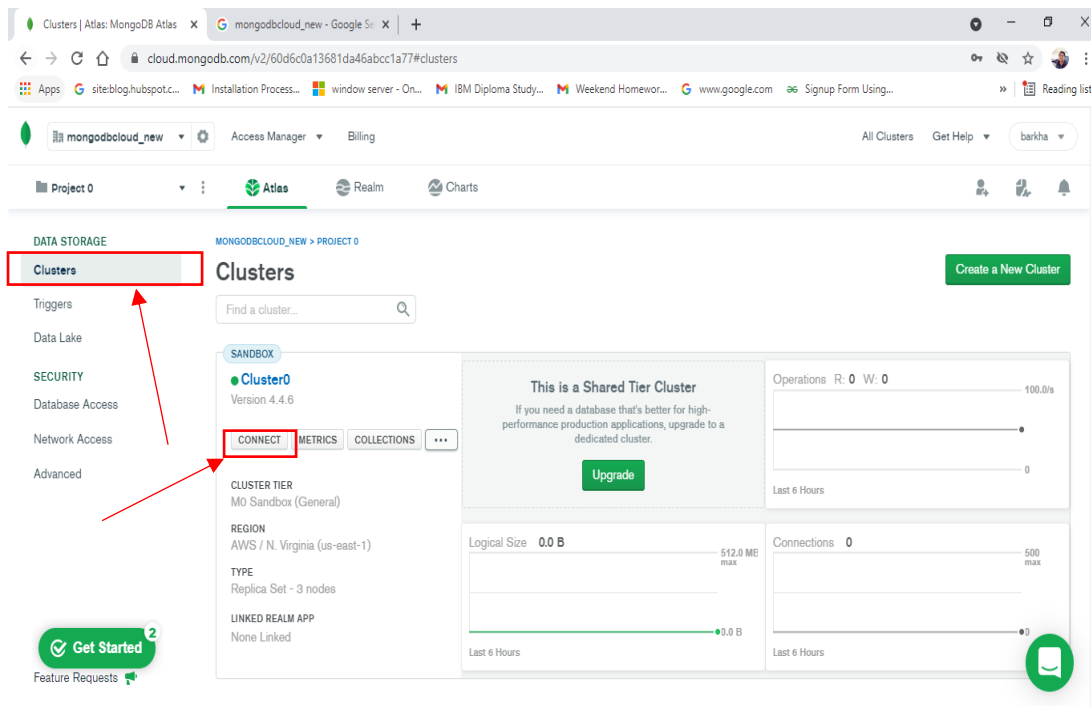


You can see active in status IP

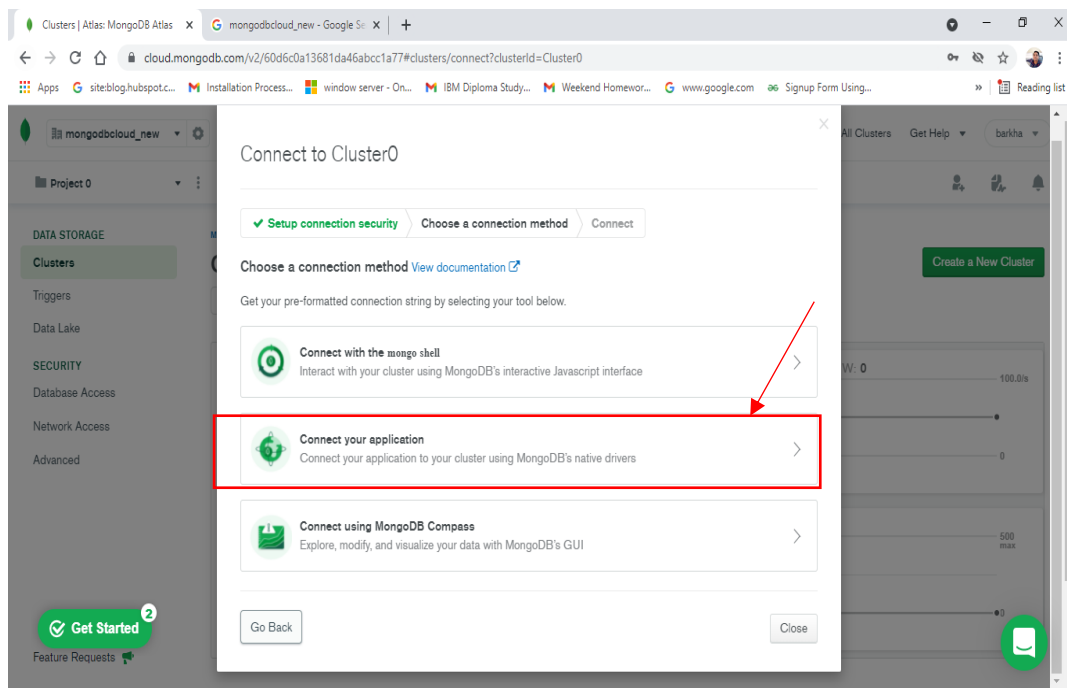


Go to cluster

## Click on connect

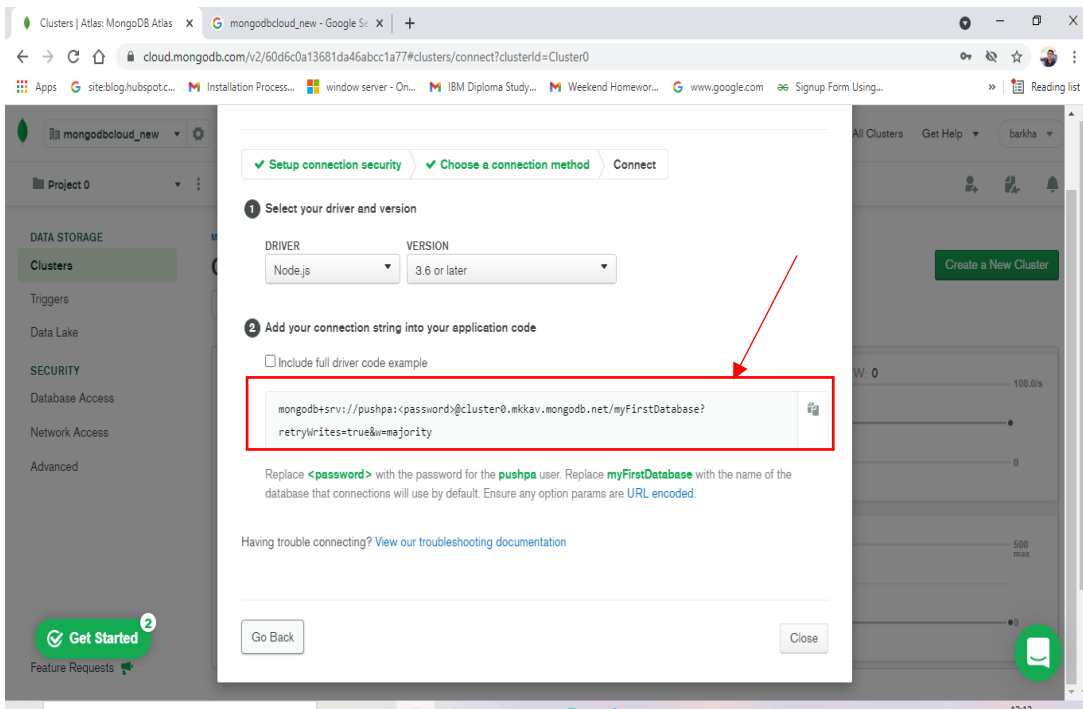


## Select connect your application



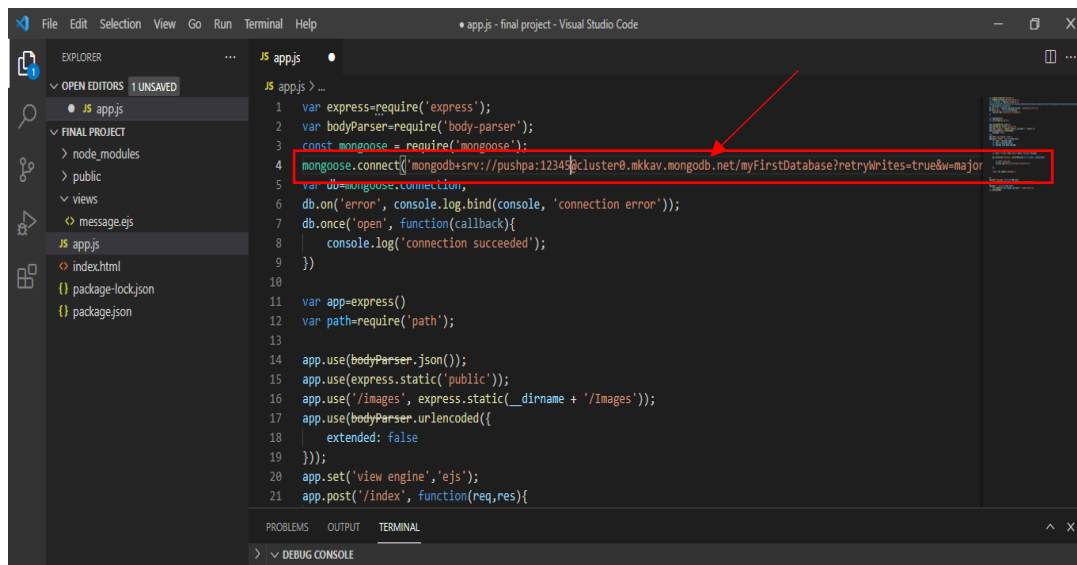
## Copy the link

## Paste it your app.js file



You can see it image

Save it



Run the command node app.js in terminal

Press enter

The screenshot shows the Visual Studio Code interface with the 'app.js' file open in the editor. The code is a Node.js application using Express.js and MongoDB. The terminal at the bottom shows the command 'node app.js' being executed in a PowerShell window. A red box highlights the terminal command, and a red arrow points to 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){

```

Terminal output:

```

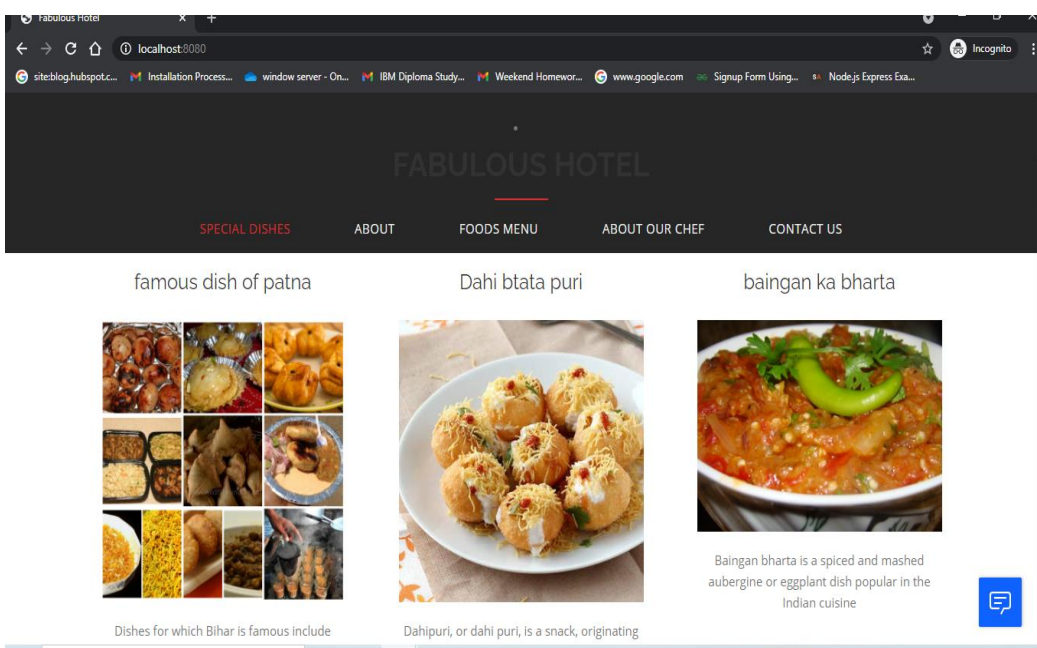
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS D:\final project> node app.js

```

My webpage run on 8080 port no



## User sending his feedback

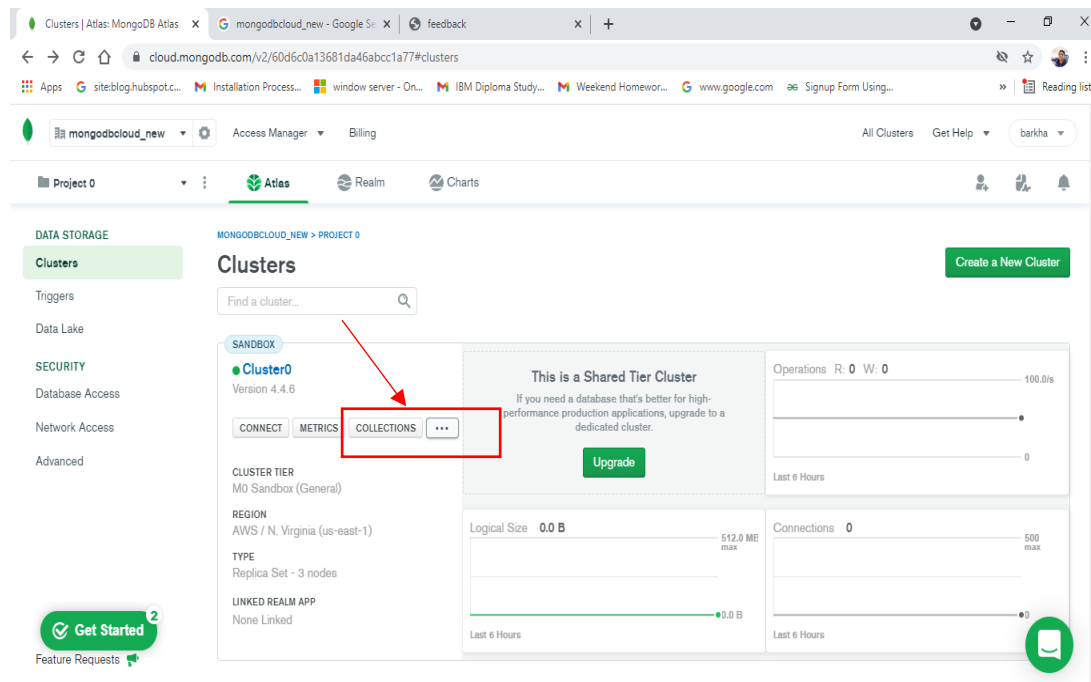
The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The website has a dark theme with the title 'FABULOUS HOTEL' at the top. A navigation menu includes 'SPECIAL DISHES', 'ABOUT', 'FOODS MENU', 'ABOUT OUR CHEF', and 'CONTACT US'. The 'CONTACT US' link is highlighted in red. Below the navigation menu, there is a section titled 'SEND US A MESSAGE' with a 'Page 22' indicator. The form contains three input fields: a name field with 'sita singh', an email field with 'ssfsa@gmail.com', and a message field with 'thank u'. A 'SEND MESSAGE' button is located at the bottom of the form.

## Open thank u page

The screenshot shows a web browser window with the address bar displaying 'localhost:3333/index'. The page features a large, bold heading that reads 'Thank you for Using my Website'. Below this heading, there is a link labeled 'Home' in a purple color. The browser's taskbar at the bottom shows several open applications, including 'Apps', 'siteblog.hubspot...', 'Installation Process...', 'window server - On...', 'IBM Diploma Study...', 'Weekend Homewor...', 'www.google.com', and 'Signup Form Using...'. The system clock in the bottom right corner indicates the time is 16:53.

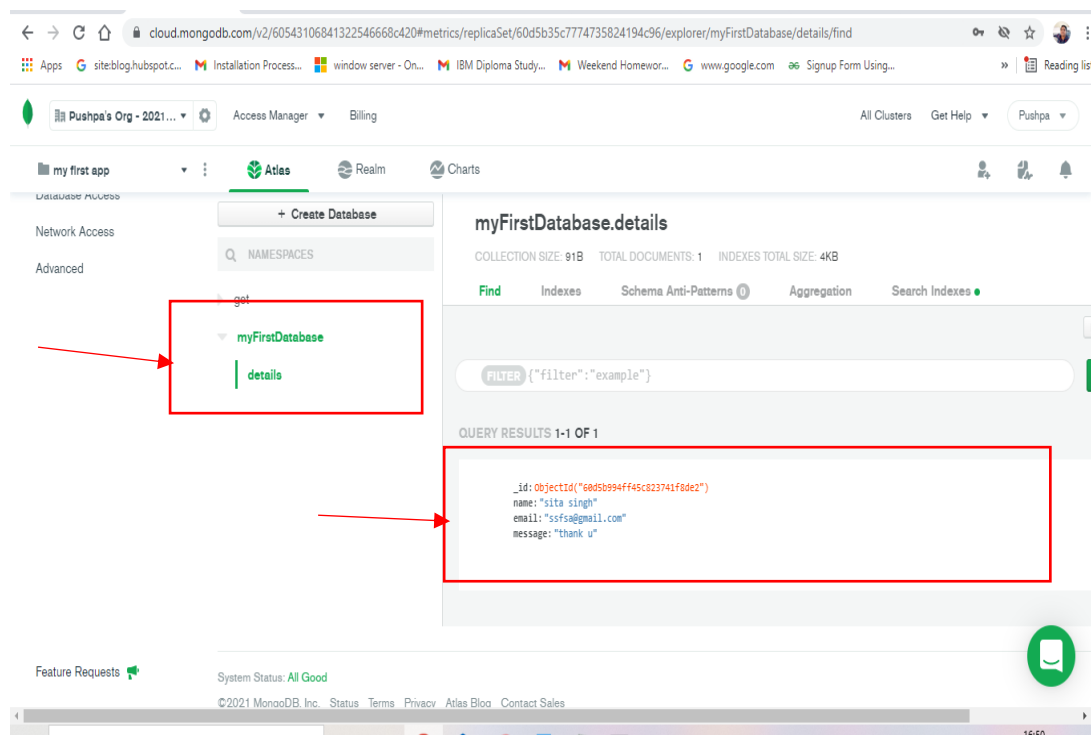
# See where store your data in atlas

Click on collections



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

Store user data in details



# 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>

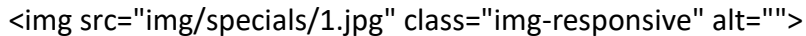
    </div>

    <div class="row">

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

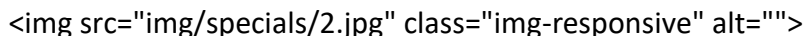
        <div class="features-item">
```

### famous dish of patna



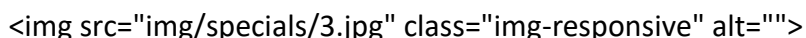
Dishes for which Bihar is famous include Bihari kebabs, litti chokha, Bihari boti, Bihari chicken masala, sattu paratha (parathas stuffed with roasted gram flour), chokha (spicy mashed eggplant and potatoes), fish curry and posta-dana ka halwa.

### Dahi btata puri



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

### baingan ka bharta



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

```
<!-- 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>
```

```
</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/ Neymeen 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 ... Goody 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" place-
  holder="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://localhost:27017/fruits');

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": ""  
}
```

## 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

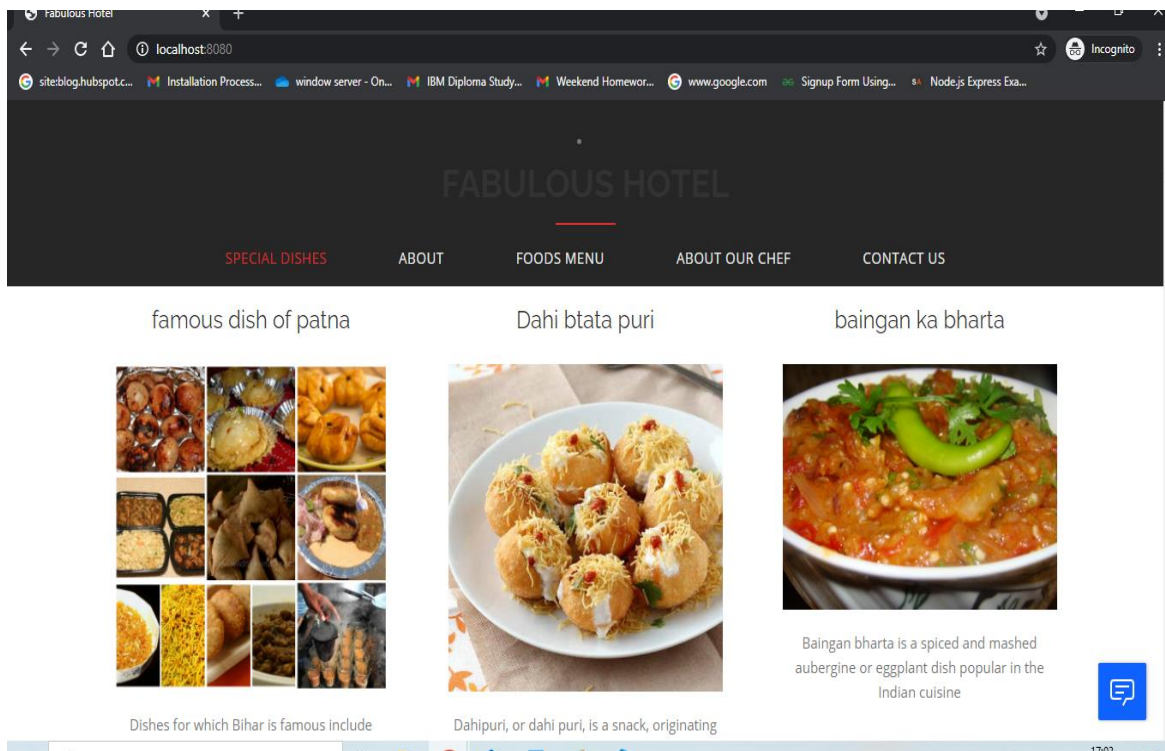
<https://www.mongodb.com/>

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

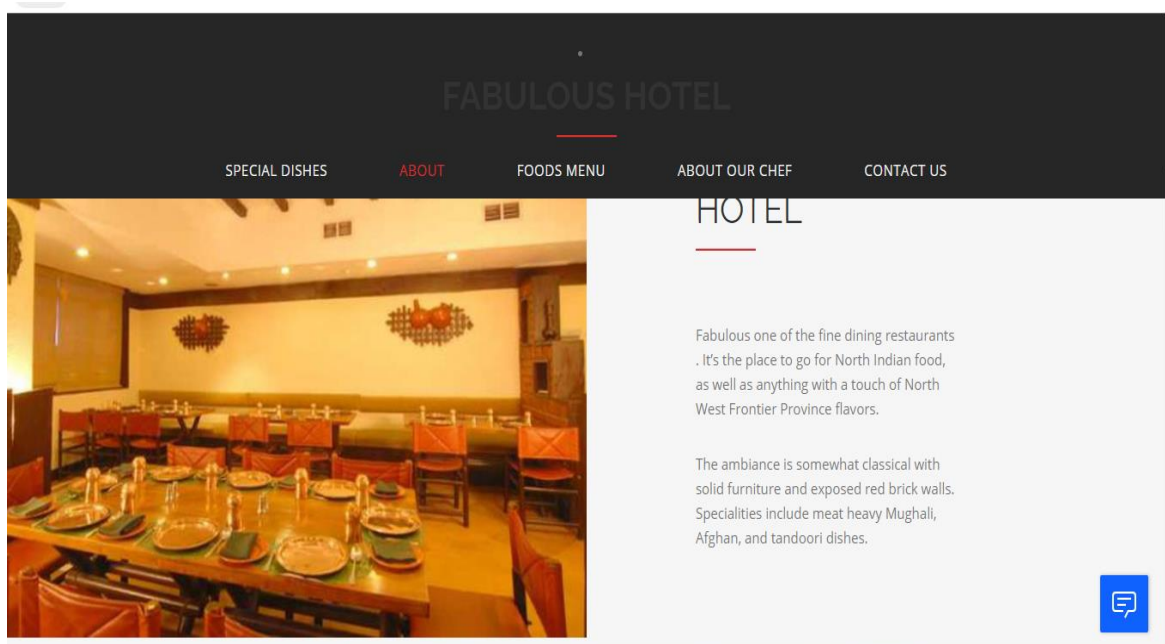
<https://www.mongodb.com/blog/post/quick-start-nodejs-mongodb--how-to-get-connected-to-your-database>

# Screenshots

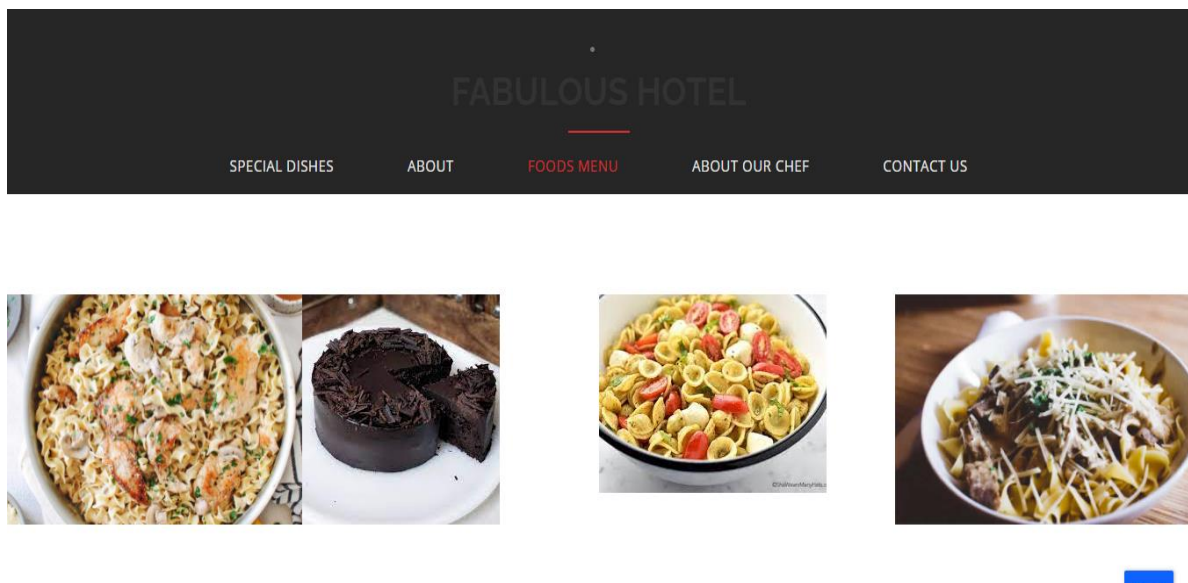
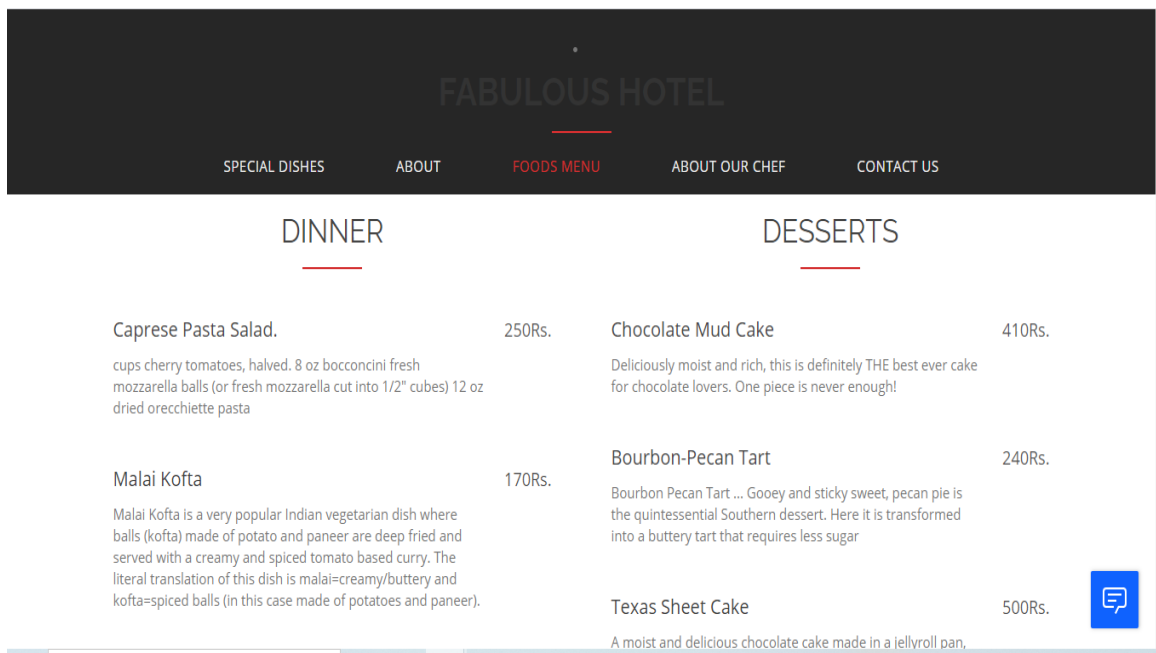
## Home page



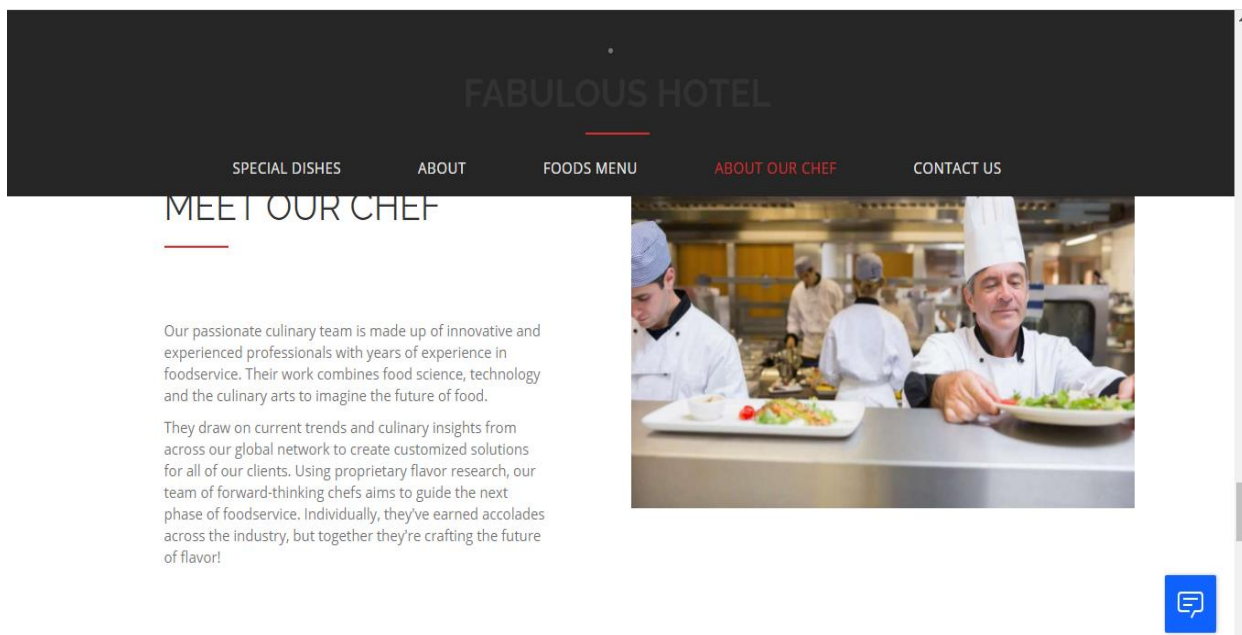
## About page



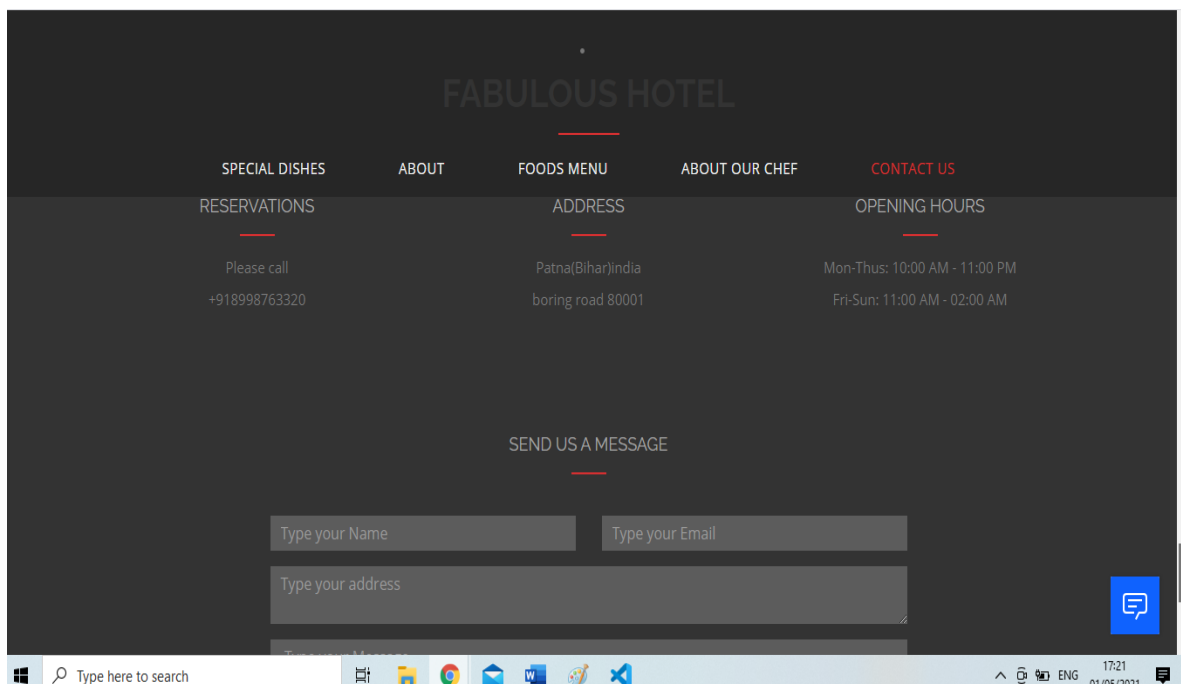
## food menu page



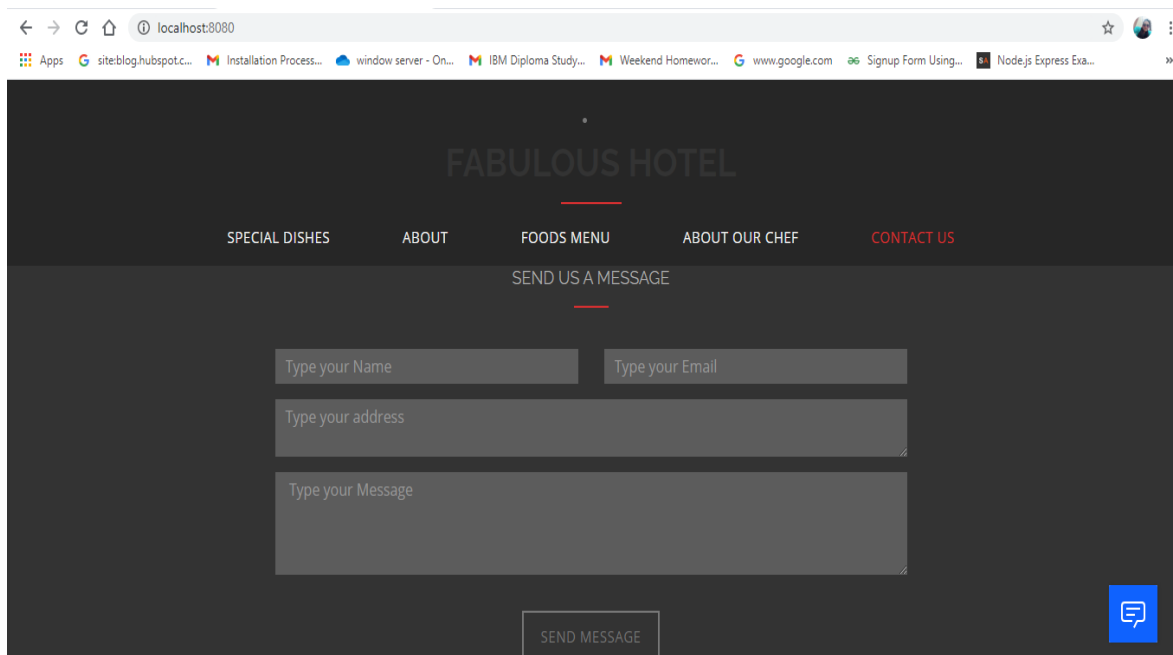
## our chef page page



## contact us page

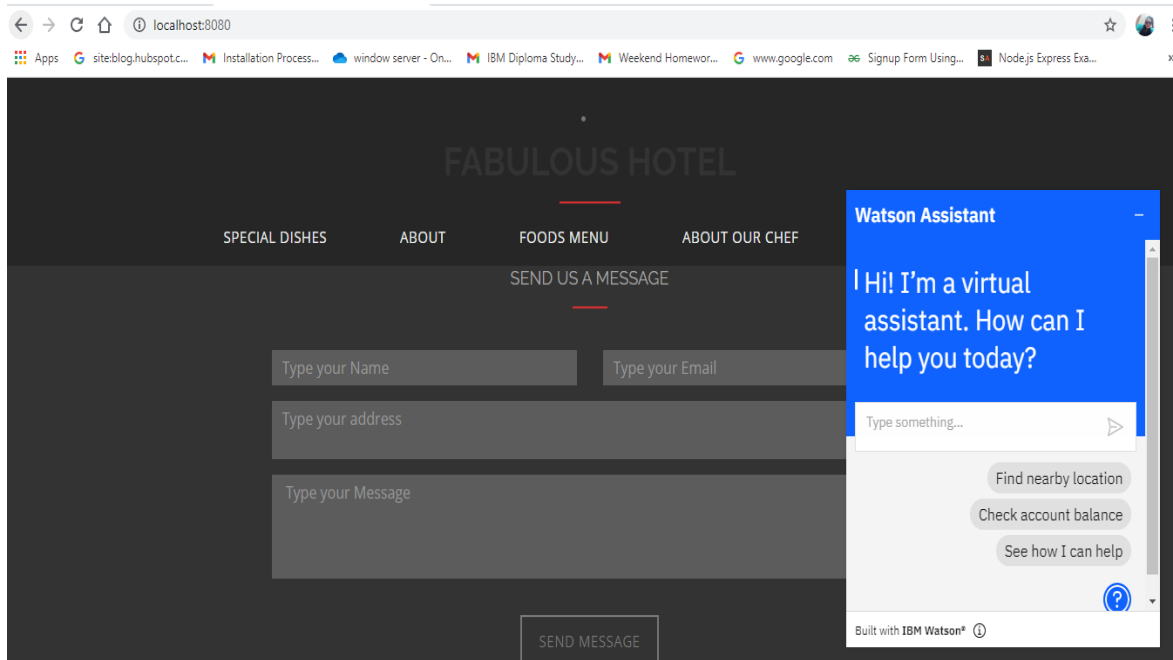


*In this field you can send your message*



The screenshot shows a web browser window with the address bar displaying 'localhost:8080'. The browser's tab bar includes several open tabs: 'Apps', 'site:blog.hubspot.c...', 'Installation Process...', 'window server - On...', 'IBM Diploma Study...', 'Weekend Homewor...', 'www.google.com', 'Signup Form Using...', and 'Node.js Express Exa...'. The main content area of the browser displays 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' (highlighted in red). Below the navigation menu, there is a section titled 'SEND US A MESSAGE' with a red underline. 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



This screenshot shows the same web browser window as the previous one, but with a 'Watson Assistant' chat widget overlaid on the right side. The widget has a blue header with the text 'Watson Assistant'. Below the header, it says 'Hi! I'm a virtual assistant. How can I help you today?'. There is a text input field with the placeholder 'Type something...' and a send button. Below the input field, there are three suggested actions: 'Find nearby location', 'Check account balance', and 'See how I can help'. At the bottom of the widget, it says 'Built with IBM Watson®' and includes a small information icon.



**THANK YOU...**