GRILL & CHILL

Submitted in partial fulfilment of the
Requirements for the award of the degree

of
Bachelor of Technology

in
Computer Science & Engineering

By:

Vidisha Grover (202768027187/CSE-3/2018)

Department of Computer Science & Engineering
Guru Tegh Bahadur Institute of Technology

Guru Gobind Singh Indraprastha University
Dwarka, New Delhi

Year 2018-2022

GRILL & CHILL

Duration

12th May, 2020 – 15th June, 2020

By:

Vidisha Grover (20276802718/CSE-3/2018)

From

Udemy, Inc

At

www.udemy.com

DECLARATION

I hereby declare that all the work presented in this Industrial Training Report for the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science & Engineering**, Guru Tegh Bahadur Institute of Technology, affiliated to Guru Gobind Singh Indraprastha University Delhi is an authentic record of our own work carried out at Udemy from 12th May, 2020 to 15th June, 2020.

Date:

Vidisha Grover (20276802718 / CSE-3/2018)

CERTIFICATE

ACKNOWLEDGEMENT

I express my sincere gratitude to Ms. Angela Yu for her valuable guidance and timely suggestions during the entire duration of my dissertation work, without which this work would not have been possible. I also take this opportunity to give thanks to all others who gave their support for the project or in other aspects of study at Guru Tegh Bahadur Institute of Technology. Finally, I would also like to thank my friends for their advice and pointing out my mistakes.

Vidisha Grover (20276802718 /CSE-3/2018) Date:
grovervidisha8@gmail.com

ABSTRACT

Problem:

People often find it difficult to promote their brands and business. Due to this they lack publicity in the market. Not getting enough audience might result in loss and leads to shutting of business. Not having enough knowledge of marketing is also one of the reasons why business suffers losses. In Business you need to have one to one interaction with your audience. But due to less interaction with the users and less availability of information on Internet businesses are unable to create good impact on their audience.

Solution:

One of the most effective ways to get your business online is getting a website for it. A website not only helps small businesses to promote and sell their product and services, but it also helps them to outshine from their competitors, especially nowadays as people mostly rely on Internet. Having website for business not only gives you publicity in market but also helps you to interact with customers easily. As having a website for business helps you to provide your services both offline and online. It is very easy for customer to learn about someone's business through website.

LIST OF FIGURES AND TABLES

CONTENTS

2. INTRODUCTION


GURU TEGH BAHADUR INSTITUTE OF TECHNOLOGY
DELHI
ਦੇਗ ਤੇਗ ਫਤਹਿ॥

As having business website helps your business to grow and creates a way to interact with your users. **Grill and Chill** is a web application that is desktop as well as phone responsive. It is user friendly and provides easy functionality. All the web pages are beautifully design with good typography and efficient color scheme. Main reason to have business website is people can easily know about your services. Just like that this website contains all relevant information regarding the restaurant like menu, reservation, about and contact. For reservation of table user has to input specific details which gets stored in database to maintain the record. Similarly Contact Us page is also created to take up user's suggestion regarding the restaurant. Menu items that are displayed on webpage are also stored in database to create dynamic data. Website is made interactive by adding icons and related images.

Frontend:

1. HTML
2. CSS
3. JS

For backend Nodejs is used and related dependencies are installed using npm. **Node.js** is an open-source, cross-platform, back-end, JavaScript runtime that executes JavaScript code outside a web browser.

Dependencies:

1. Express
2. Ejs
3. Express-ejs-layouts
4. Mongoose
5. Nodemon

2. REQUIREMENTS

2.1 What is Software Requirement Specification(SRS)

A Software Requirement Specification (SRS) is a description of a software system to be developed. It lays out functional requirements and may include a set of use cases that describe user interactions that the software must provide.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements, the developer needs to have a clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

The SRS may be one of a contact deliverable Data Item Description or have other forms of organizationally-mandated content.

2.2 Uses of SRS

The purpose of the document is to collect and analyse all assorted ideas that have come up to define the system, the requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product developers.

In short, the purpose of this SRS document is to provide a detailed overview of our

software product, its parameters and goals. This document describes the project's

target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

2.3 Software Requirements

This project was built on VS Code and can run on any operating system with code editor installed in it which can edit HTM, CSS , JS , EJS files . Rest of the requirements include: -

- Processor
  - o Windows: It can be run on any version of Windows Operating System with proper code editor installed in it and can be run using Command Prompt or by installing Hyper.
  - o Linux: You can run this project on any version of Linux using terminal or by installing Hyper..
  - o MacOS: This project can run on any version of MacOS using the terminal or by installing Hyper..
  - o Language: Javascript
  - o Framework: Express

2.4 Hardware Requirements

Processor: Standard processor with a minimum speed of 1.8 GHz RAM: minimum 256 MB of RAM, 512MB recommended.
Hard Disk: Minimum 3 GB

3. WEB DEVELOPMENT

Web development is the work involved in developing a Web site for the Internet (World Wide Web) or an intranet (a private network).Web development can range from developing a simple single static page of plain text to complex Web-based Internet applications (Web apps), electronic businesses , and social network services . A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering , Web design , Web content development , client liaison, client-side / server-side scripting , Web server and network security configuration, and e-commerce development. Every Web Developer must have basic understanding of HTML , CSS , JS.

3.1 FRONTEND

HTML	CSS	JAVASCRIPT
HTML	CSS	JavaScript
HTTP/XHR	CSS Responsive	ECMASCRIPT5

Fig 1: Frontend in web development

3.2 DIGGING DEEPER

HTML	CSS	JAVASCRIPT
HTML DOM	CSS Icons	XML
Google Maps		JSON
Google Fonts		AJAX
Google Charts		

Fig 2: Digging Deeper in web development

3.3 FRAMEWORKS

CSS	JAVASCRIPT	XML
Bootstrap	jQuery	XSLT
W3.css	AngularJS	XPath
	Vue.js	XQuery
	W3.JS	

Fig 3: Frameworks in web development

3.4 FULLSTACK JS

FULLSTACK JS
SQL
Node.js
MySQL
Mongo.db

Fig 4: FullStack JS

4. FRONTEND

4.1 HTML

Hypertext Markup Language (**HTML**) is the standard markup language for documents designed to be displayed in a web browser. 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 page semantically and originally included cues for the appearance of the document.HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Browsers do not display the HTML tags, but use them to interpret the content of the page. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

4.1.1 History of HTML

DEVELOPMENT

In 1980, physicist Tim Berners-Lee, a contractor at CERN, proposed and prototyped ENQUIRE, a system for CERN researchers to use and share documents. In 1989, Berners-Lee wrote a memo proposing an Internet-based hypertext system. Berners-Lee specified HTML and wrote the browser and server software in late 1990. That year, Berners-Lee and CERN data systems engineer Robert Cailliau collaborated on a joint request for funding, but the project was not formally adopted by CERN. In his personal notes from 1990 he listed "some of the many areas in which hypertext is used" and put an encyclopedia first.

The first publicly available description of HTML was a document called "HTML Tags", first mentioned on the Internet by Tim Berners-Lee in late 1991. It describes 18 elements comprising the initial, relatively simple design of HTML.
Many of the text elements are found in the 1988 ISO technical report TR 9537 Techniques for using SGML, which in turn covers the features of early text formatting languages such as that used by the RUNOFF command developed in the

early 1960s for the CTSS (Compatible Time-Sharing System) operating system: these formatting commands were derived from the commands used by typesetters to manually format documents.
Berners-Lee considered HTML to be an application of SGML. It was formally defined as such by the Internet Engineering Task Force (IETF) with the mid-1993 publication of the first proposal for an HTML specification, the "Hypertext Markup Language (HTML)" Internet Draft by Berners-Lee and Dan Connolly, which included an SGML Document type definition to define the grammar. The draft expired after six months, but was notable for its acknowledgment of the NCSA Mosaic browser's custom tag for embedding in-line images, reflecting the IETF's philosophy of basing standards on successful prototypes. Similarly, Dave Raggett's competing Internet-Draft, "HTML+ (Hypertext Markup Format)", from late 1993, suggested standardizing already-implemented features like tables and fill-out forms. After the HTML and HTML+ drafts expired in early 1994, the IETF created an HTML Working Group, which in 1995 completed "HTML 2.0", the first HTML specification intended to be treated as a standard against which future implementations should be based.

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
...	Set ... in boldface
...	Set ... in italics
...	Center ... on the page horizontally
...	Brackets an unordered (bulleted) list
...	Brackets a numbered list
...	Brackets an item in an ordered or numbered list
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a horizontal rule
<img src="...">	Displays an image here
 ... 	Defines a hyperlink

Fig 5: HTML Tags

4.1.2 HTML TIMELINE

1989	Tim Berners-Lee invented www
1991	Tim Berners-Lee invented HTML
1993	Dave Raggett drafted HTML+
1995	HTML Working Group defined HTML 2.0
1997	W3C Recommendation: HTML 3.2
1999	W3C Recommendation: HTML 4.01
2000	W3C Recommendation: XHTML 1.0
2008	WHATWG HTML5 First Public Draft
2012	WHATWG HTML5 Living Standard
2014	W3C Recommendation: HTML5
2016	W3C Candidate Recommendation: HTML 5.1
2017	W3C Recommendation: HTML5.1 2nd Edition
2017	W3C Recommendation: HTML5.2

Fig 6: HTML Timeline

4.2 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, colors, 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.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

4.2.1 History of CSS

CSS was first proposed by Håkon Wium Lie on October 10, 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside World Wide Web Consortium resulted in the first W3C CSS Recommendation (CSS1) being released in 1996. In particular, a proposal by Bert Bos was influential; he became co-author of CSS1, and is regarded as co-creator of CSS.

Style sheets have existed in one form or another since the beginnings of Standard Generalized Markup Language (SGML) in the 1980s, and CSS was developed to provide style sheets for the web. One requirement for a web style sheet language was for style sheets to come from different sources on the web. Therefore, existing style sheet languages like DSSSL and FOSI were not suitable. CSS, on the other hand, let a document's style be influenced by multiple style sheets by way of "cascading" styles.

Improving web presentation capabilities was a topic of interest to many in the web community and nine different style sheet languages were proposed on the www-style mailing list. Of these nine proposals, two were especially influential on what became CSS: Cascading HTML Style Sheets and Stream-based Style Sheet Proposal (SSP). Thereafter, Lie and Bos worked together to develop the CSS standard.

Lie's proposal was presented at the "Mosaic and the Web" conference (later called WWW2) in Chicago, Illinois in 1994, and again with Bert Bos in 1995. Around this

time the W3C was already being established, and took an interest in the development of CSS. In August 1996, Netscape Communication Corporation presented an alternative style sheet language called JavaScript Style Sheets (JSSS). The spec was never finished, and is deprecated. By the end of 1996, CSS was ready to become official, and the CSS level 1 Recommendation was published in December. Development of HTML, CSS, and the DOM had all been taking place in one group, the HTML Editorial Review Board (ERB).

4.2.2 CSS TIMELINE

1996	W3C Recommendation: CSS 1
1998	W3C Recommendation: CSS 2
1999	W3C revises CSS 1 Recommendation
1999	First 3 CSS 3 drafts

2002	First draft of CSS 2
2011	W3C Recommendation: CSS 2.1
2016	First draft of 2nd revision of CSS 2
2017	Final Draft of CSS3

Fig 7: CSS Timeline

4.3 JS

Alongside HTML and CSS, **JavaScript** is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behavior , and all major web browsers have a dedicated JavaScript engine to execute it.

JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

4.3.1 History of JS

CREATION AT NETSCAPE

During these formative years of the Web, web pages could only be static, lacking the capability for dynamic behavior after the page was loaded in the browser. So in 1995, Netscape decided to add a scripting language to Navigator. They pursued two routes to achieve this: collaborating with Sun Microsystems to embed the Javaprogramming language, while also hiring Brendan Eich to embed the Scheme language.

Netscape management soon decided that the best option was for Eich to devise a new language, with syntax similar to Java and less like Scheme or other extant scripting languages. Although the new language and its interpreter implementation were officially called LiveScript when first shipped as part of a Navigator release in September 1995, the name was changed to JavaScript three months later.

ADOPTION BY MICROSOFT

Microsoft debuted Internet Explorer in 1995, leading to a browser war with Netscape. On the JavaScript front, Microsoft reverse-engineered the Navigator interpreter to create its own, called JScript.

JScript was first released in 1996, alongside initial support for CSS and extensions to HTML. Each of these implementations was noticeably different from their counterparts in Navigator. These differences made it difficult for developers to make their websites work well in both browsers, leading to widespread use of "best viewed in Netscape" and "best viewed in Internet Explorer" logos for several years.

GROWTH AND STANDARDIZATION

In July 2008, these disparate parties came together for a conference in Oslo. This led to the eventual agreement in early 2009 to combine all relevant work and drive the language forward. The result was the ECMAScript 5 standard, released in December 2009.

REACHING MATURITY

From 2016 to 2019, a new version of the ECMAScript standard was published each year, but the scope of changes was much smaller than the 5th or 6th editions. Thus JavaScript can now be considered a mature language that has largely settled down. The current JavaScript ecosystem has many libraries and frameworks, established programming practices, and increased usage of JavaScript outside of web browsers. Plus, with the rise of single-page applications and other JavaScript-heavy websites, a number of transpilers have been created to aid the development process.

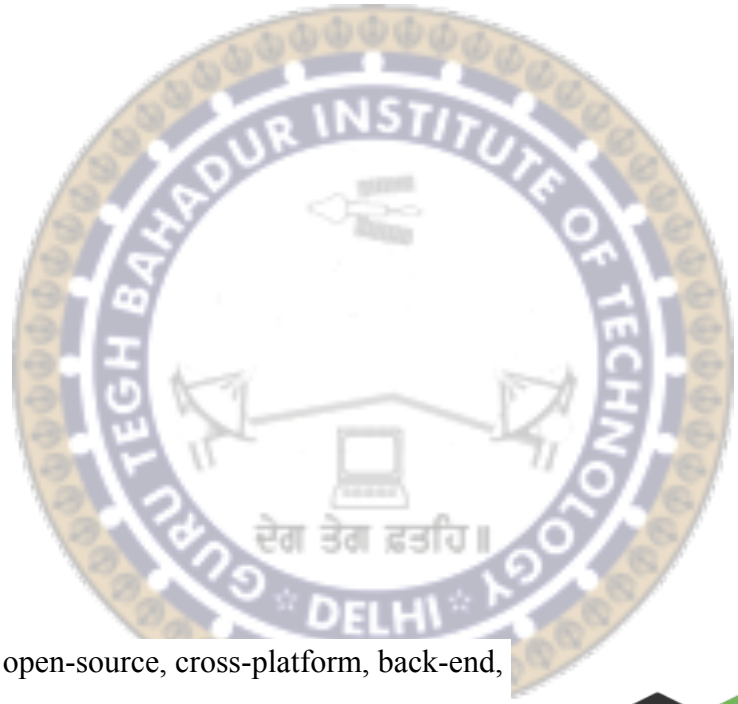
4.3.2 JS TIMELINE

ES1	ECMAScript 1 (1997)

ES2	ECMAScript 2 (1998)
ES3	ECMAScript 3 (1999)
ES4	ECMAScript 4
ES5	ECMAScript 5 (2009)
ES6	ECMAScript 2015
	ECMAScript 2016
	ECMAScript 2017 ECMAScript 2018

Fig 8: JS Timeline

5. NODE.js



Node.js is an open-source, cross-platform, back-end, JavaScript runtime that executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS

Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

WHAT DOES Node.js DO?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database.

5.1 History of Node.js

Node.js was written initially by Ryan Dahl in 2009, about thirteen years after the introduction of the first server-side JavaScript environment, Netscape's LiveWire Pro Web. The initial release supported only Linux and Mac OS X. Its development and maintenance was led by Dahl and later sponsored by Joyent.

In January 2010, a package manager was introduced for the Node.js environment called npm. The package manager makes it easier for programmers to publish and share source code of Node.js packages and is designed to simplify installation, updating, and un-installation of packages.

In June 2011, Microsoft and Joyent implemented a native Windows version of Node.js. The first Node.js build supporting Windows was released in July 2011.

In January 2012, Dahl stepped aside, promoting coworker and *npm* creator Isaac Schlueter to manage the project. In January 2014, Schlueter announced that Timothy J. Fontaine would lead the project.

In December 2014, Fedor Indutny started io.js, a fork of Node.js. Due to the internal conflict over Joyent's governance, io.js was created as an open governance alternative with a separate technical committee. In February 2015, the intent to form a neutral Node.js Foundation was announced. By June 2015, the Node.js and io.js communities voted to work together under the Node.js Foundation.

In September 2015, Node.js v0.12 and io.js v3.3 were merged back together into Node v4.0. This merge brought V8 ES6 features into Node.js and a long-term support release cycle. As of 2016, the io.js website recommends that developers switch back

to Node.js and that no further releases of io.js are planned due to the merge. In 2019, the JS Foundation and Node.js Foundation merged to form the OpenJS Foundation.

5.2 Node.js TIMELINE

Release	Code name	Release date
v0.10.x	End-of-Life	2013-03-11
v0.12.x	End-of-Life	2015-02-06
4.x	End-of-Life	2015-09-08
5.x	End-of-Life	2015-10-29
6.x	End-of-Life	2016-04-26
7.x	End-of-Life	2016-10-25
8.x	End-of-Life	2017-05-30
9.x	End-of-Life	2017-10-01
10.x	Maintenance LTS	2018-04-24
11.x	End-of-Life	2018-10-23
12.x	Active LTS	2019-04-23
13.x	End-of-Life	2019-10-22
14.x	Active LTS	2020-04-21
15.x	**Current**	2020-10-20
16.x	Pending	2021-04-20

	Old version
	Older version, still maintained
	Latest version
	Future release

Fig 9: Nodejs Timeline

5.3 Node.js NPM

NPM is a package manager for Node.js packages, or modules if you like. The NPM program is installed on your computer when you install Node.js. A package in Node.js contains all the files you need for a module. Modules are JavaScript libraries you can include in your project.

ALL BELOW PACKAGES / DEPENDENCIES ARE INSTALLED USING NPM

Fig 10: Dependencies

5.4 EXPRESS

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Installing Express :

$ npm install express

5.5 EJS

EJS stand for **Embedded JavaScript**. EJS is a simple templating language that lets you generate HTML markup with plain JavaScript. No religiousness about how to organize things. No reinvention of iteration and control-flow. It's just plain JavaScript. JavaScript code in simple, straightforward scriptlet tags. Just write JavaScript that emits the HTML you want. Expess-ejs-layouts is the layout used for web application.

Installing Ejs and Express-ejs-layouts :

$ npm install ejs express-ejs-layouts

5.6 MONGOOSE

Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB. MongoDB is a schema-less NoSQL document database.

Installing Express :

$ npm install mongoose

5.6 NODEMON

Nodemon is a tool that helps develop node.js based application automatically restarting the node application when file changes.

Installing Express :

$ npm install nodemon

6. DATABASE

Express apps can use many different databases, and there are several approaches you can use for performing **C**reate, **R**ead, **U**pdate and **D**elete (CRUD) operations. This tutorial provides a brief overview of some of the available options and then goes on to show in detail the particular mechanisms selected.

6.1 DATABASES THAT CAN BE USED

Express apps can use any database supported by Node (Express itself doesn't define any specific additional behavior/requirements for database management). There are many popular options, including PostgreSQL, MySQL, Redis, SQLite, and MongoDB.

6.2 INTERACTING WITH DATABASES

- Using the databases' native query language (e.g. SQL)
- Using an Object Data Model ("ODM") or an Object Relational Model ("ORM"). An ODM/ORM represents the website's data as JavaScript objects, which are then mapped to the underlying database. Some ORMs are tied to a specific database, while others provide a database-agnostic backend.

6.3 USING MONGOOSE AND MONGODB

Mongoose is a MongoDB object modeling tool designed to work in an asynchronous environment.

Mongoose acts as a front end to MongoDB, an open source NoSQL database that uses a document-oriented data model. A “collection” of “documents” in a MongoDB database is analogous to a “table” of “rows” in a relational database.

This ODM and database combination is extremely popular in the Node community, partially because the document storage and query system looks very much like JSON, and is hence familiar to JavaScript developers.

6.4 CONNECTING MONGOOSE TO MONGODB DATABASE

Connecting to MongoDB :

Mongoose requires a connection to a MongoDB database. By **require()** and connect

to a locally hosted database with **mongoose.connect()**.

Defining schemas :

First you **require()** mongoose, then use the **Schema** constructor to create a new schema instance, defining the various fields inside it in the constructor's object parameter.

Creating a model :

Models are created from schemas using the **mongoose.model()** method.

Schema types (fields) :

A schema can have an arbitrary number of fields — each one represents a field in the documents stored in MongoDB.

6.5 MONGODB COMPASS

MongoDB Compass analyzes the documents and displays rich structures within the collections through an intuitive GUI. It allows to quickly visualize and explore the schema to understand the frequency, types and ranges of fields in the data set. Real-time server statistics let the view key server metrics and database operations. Drill down into database operations easily and understand the most active collections. Point and click to construct sophisticated queries, execute them with the push of a button and Compass will display the results both graphically and as sets of JSON documents. Modifying existing documents is easy using the intuitive visual editor, or by inserting new documents and cloning or deleting existing ones in just a few clicks. Add stages, remove them, or drag and drop to re-order in the pipeline. Once when done, export it to native code to use in your application.

Fig 11: MongoDB Compass

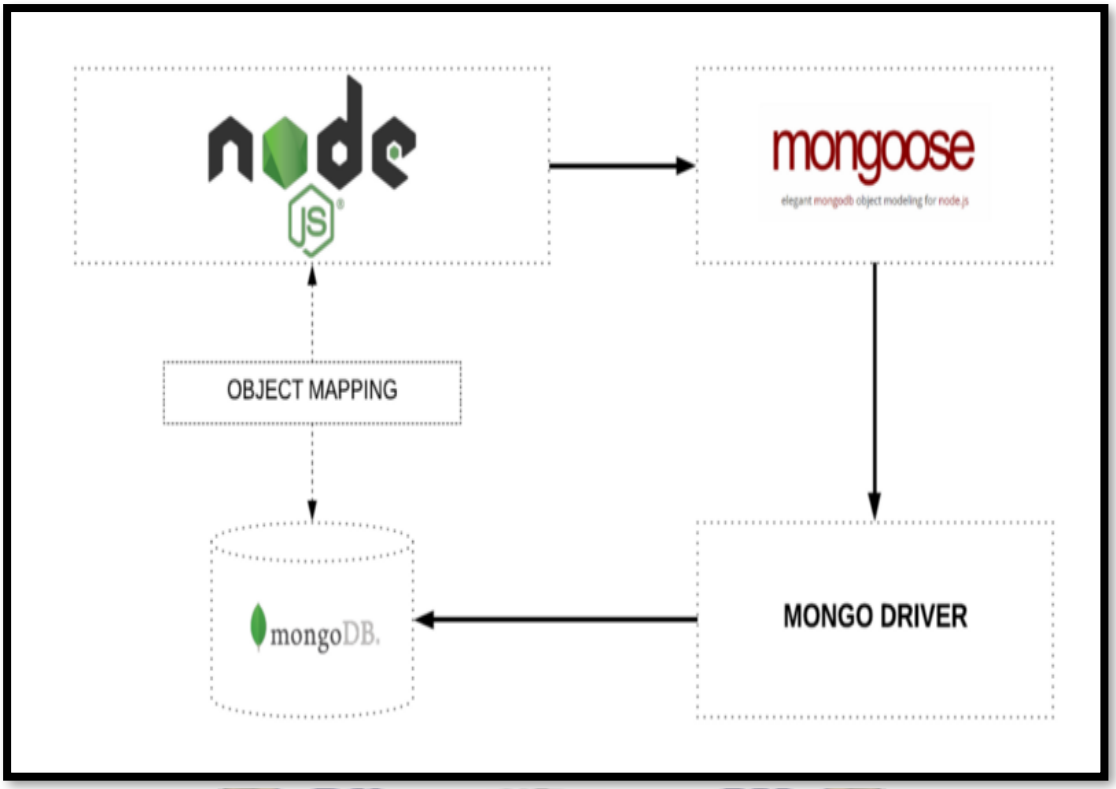


Fig 12: Illustration between Database and Nodejs

7. SUMMARY AND CONCLUSION

- Grill and Chill is a responsive website that is it can be accessed on any device be it desktop or phone. Webpages are beautifully designed. Website is user-friendly and provides easy functionality.

- It has good typography that is easily readable fonts and has efficient color scheme.

- It contains all the relevant information regarding the restaurant like menu, about, contact and reservation.

- It is made more interactive by adding icons and images of food and restaurant.

- Directions using Google Maps has also been added to make it more presentable.

- Reservation page takes all necessary information for reserving table and all data is stored in database.

- Menu items that are displayed on webpage are also stored in database to create dynamic data.

- Similarly Contact Us page is also created to take up user's suggestion regarding the restaurant.

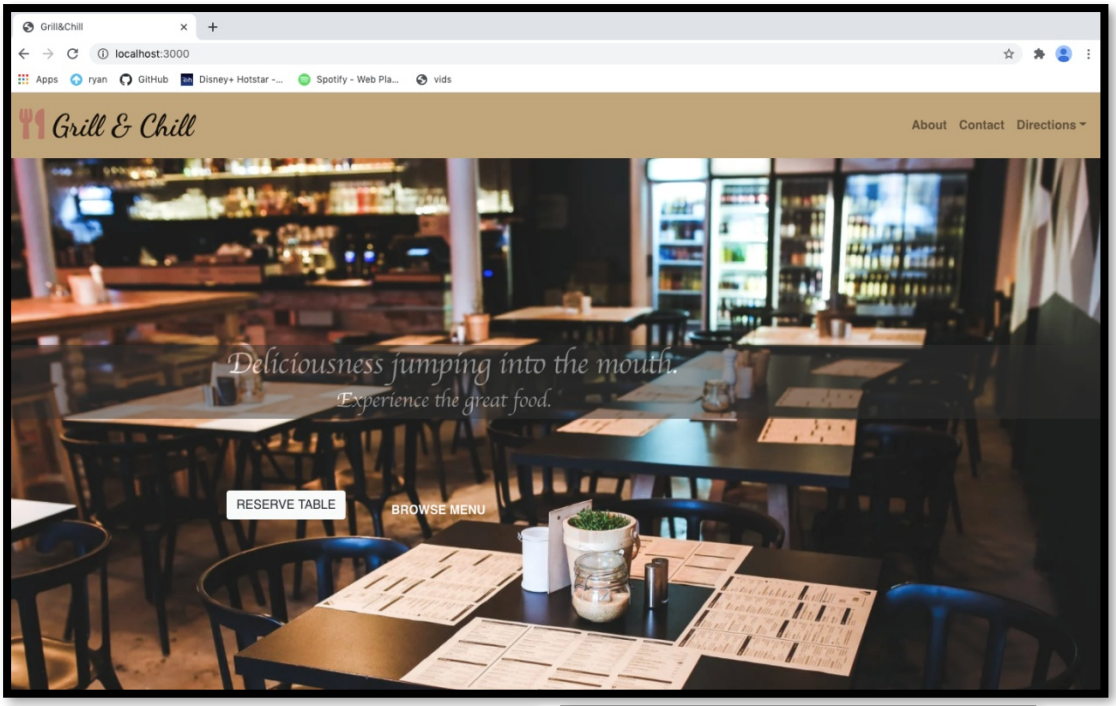
- For Future Scope, the website can also have feature of ordering online and real-time tracking their order.

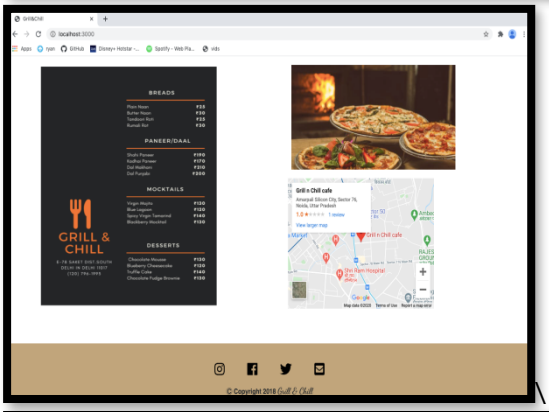
8. REFERENCES

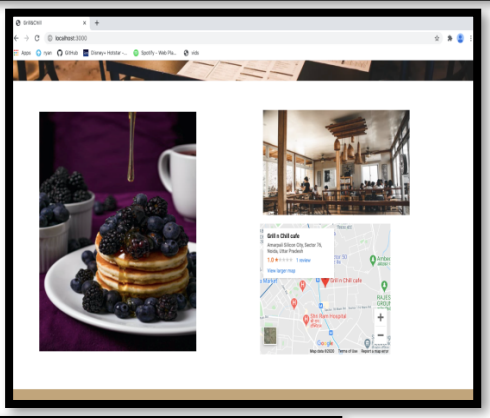
1. Bootstrap : https://getbootstrap.com/docs/4.0/getting-started/introduction/
2. Ejs : https://ejs.co/#promo

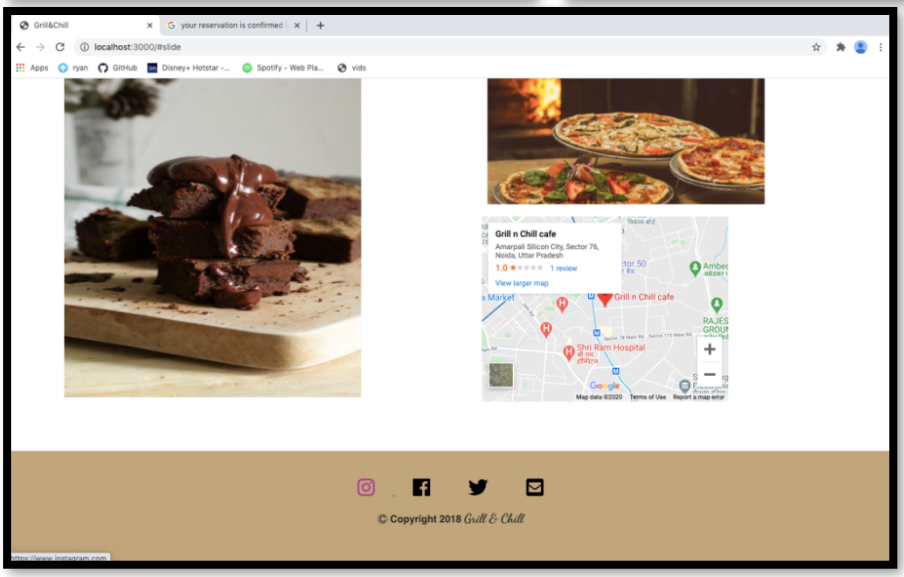
3. Express : https://expressjs.com/
4. Font Awesome Icons : https://fontawesome.com/v4.7.0/icons/
5. Geek for Geeks : https://www.geeksforgeeks.org/html-tutorials/
6. Google : https://www.google.com/
7. Google Maps : https://www.google.com/maps
8. Google Fonts : https://fonts.google.com/
9. JSON : https://www.w3schools.com/js/js_json_intro.asp
10. Laravel Mix : https://laravel-mix.com/
11. Mongoose : https://mongoosejs.com/docs/
12. Node.js : https://nodejs.org/en/
13. Pexels : https://www.pexels.com/
14. SASS : https://sass-lang.com/
15. Stack Overflow : https://stackoverflow.com/questions/tagged/javascript
16. Udemy : https://www.udemy.com/course/the-complete-web-development-bootcamp/learn/lecture/13028262#overview
17. Wikipedia : https://www.wikipedia.org/
18. W3 Schols : https://www.w3schools.com/css/default.asp
19. Youtube : https://www.youtube.com/playlist?list=PLu0W_9lII9agiCUZYRsvtGTXdxkzPyltg
20. Youtube : https://www.youtube.com/watch?v=lKNrkB6Kmgc

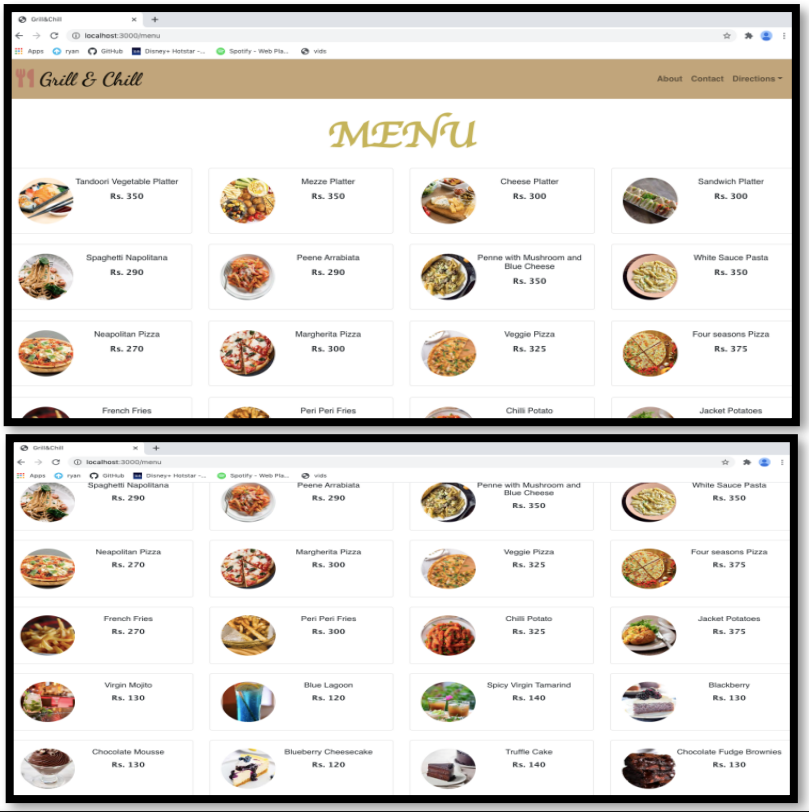
9. APPENDIX A (Screenshots)


Grill&Chill
localhost:3000
Apps
ryan
GitHub
Disney+ Hotstar -...
Spotify - Web Pla...
vids
Grill & Chill
About
Contact
Directions
Deliciousness jumping into the mouth.
Experience the great food.
RESERVE TABLE
BROWSE MENU

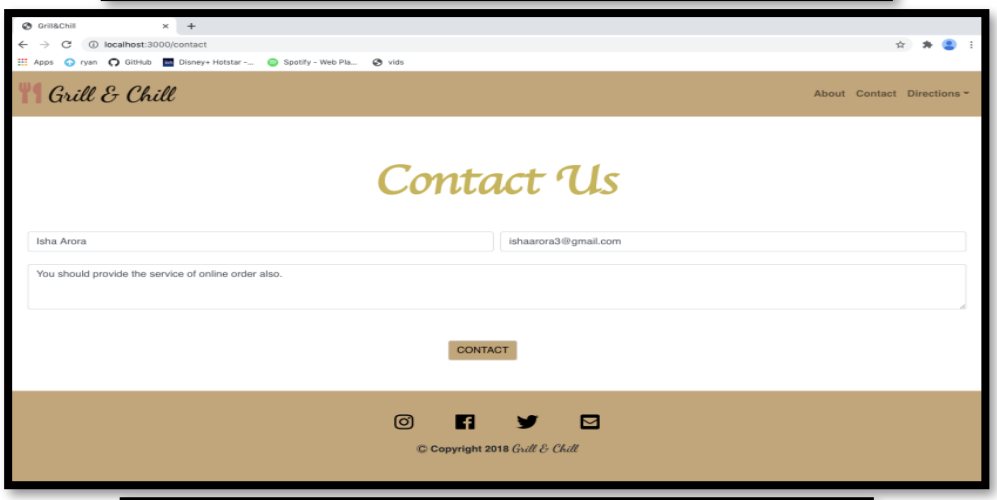

BREADS
PANEER/DAAL
MOCKTAILS
DESSERTS
GRILL & CHILL
Grill n Chill cafe
© Copyright 2018 Grill & Chill

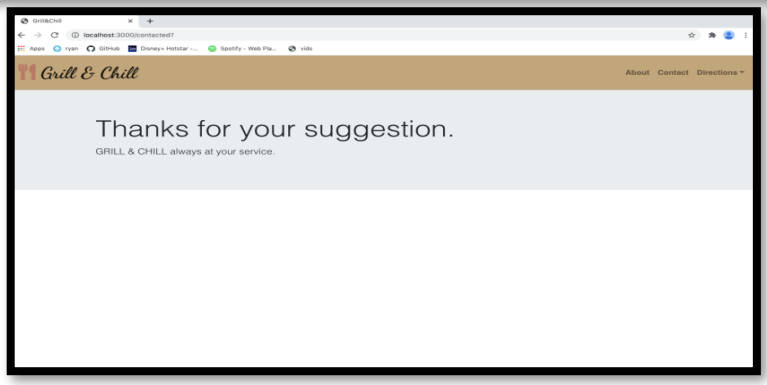


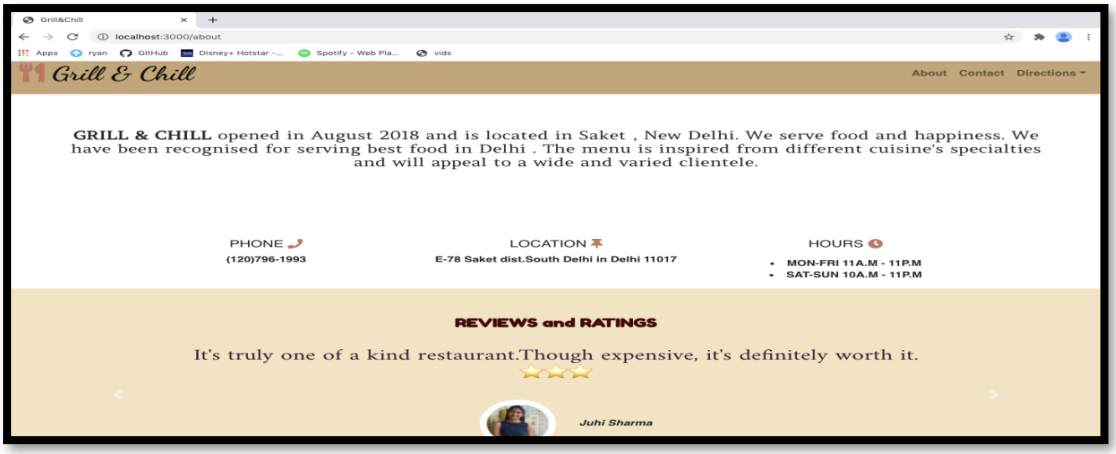

Grill&Chill
your reservation is confirmed
localhost:3000/#slide
Apps
ryan
GitHub
Disney+ Hotstar -...
Spotify - Web Pla...
vids
Grill n Chill cafe
Amarpali Silicon City, Sector 76, Noida, Uttar Pradesh
1.0
1 review
View larger map
Shri Ram Hospital
© Copyright 2018 Grill & Chill

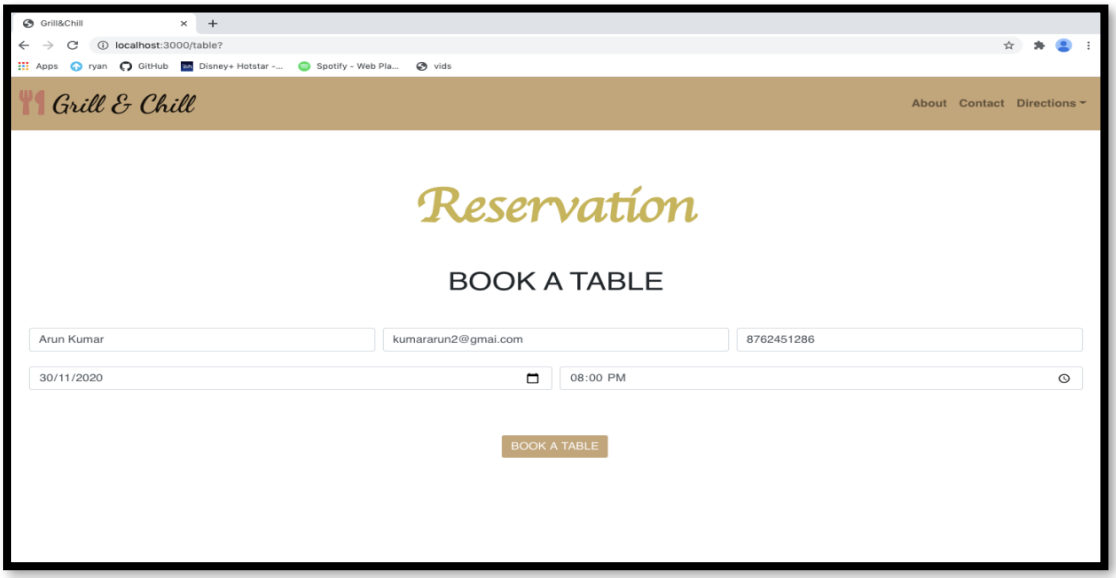

Grill & Chill
About Contact Directions
MENU
Tandoori Vegetable Platter Rs. 350
Mezze Platter Rs. 350
Cheese Platter Rs. 300
Sandwich Platter Rs. 300
Spaghetti Napolitana Rs. 290
Peene Arrabiata Rs. 290
Penne with Mushroom and Blue Cheese Rs. 350
White Sauce Pasta Rs. 350
Neapolitan Pizza Rs. 270
Margherita Pizza Rs. 300
Veggie Pizza Rs. 325
Four seasons Pizza Rs. 375
French Fries
Peri Peri Fries
Chilli Potato
Jacket Potatoes

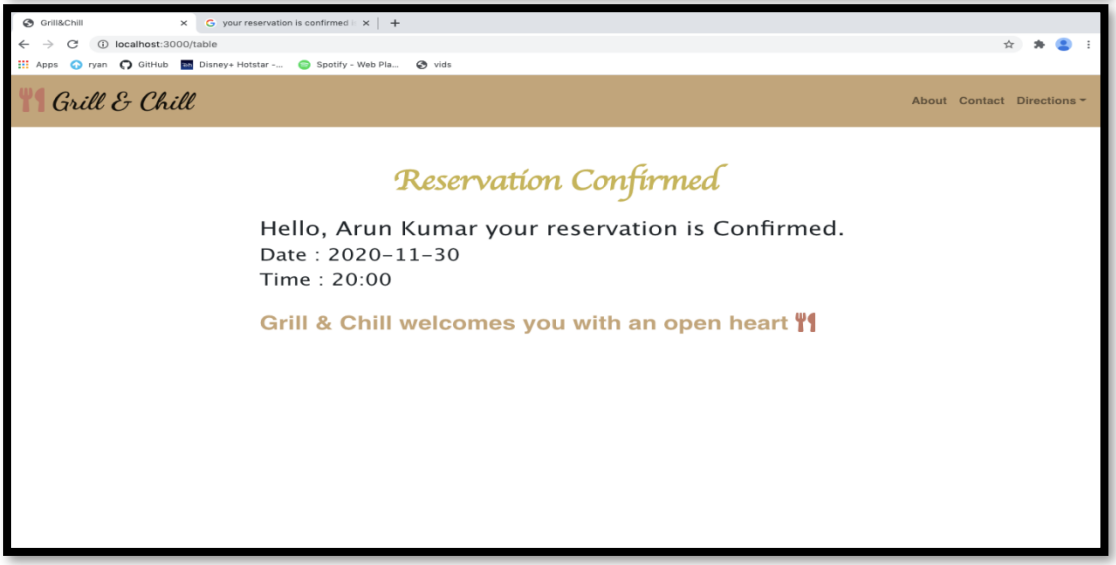
Spaghetti Napolitana Rs. 290
Peene Arrabiata Rs. 290
Penne with Mushroom and Blue Cheese Rs. 350
White Sauce Pasta Rs. 350
Neapolitan Pizza Rs. 270
Margherita Pizza Rs. 300
Veggie Pizza Rs. 325
Four seasons Pizza Rs. 375
French Fries Rs. 270
Peri Peri Fries Rs. 300
Chilli Potato Rs. 325
Jacket Potatoes Rs. 375
Virgin Mojito Rs. 130
Blue Lagoon Rs. 120
Spicy Virgin Tamarind Rs. 140
Blackberry Rs. 130
Chocolate Mousse Rs. 130
Blueberry Cheesecake Rs. 120
Truffle Cake Rs. 140
Chocolate Fudge Brownies Rs. 130


Grill & Chill
About Contact Directions
Contact Us
Isha Arora
ishaarora3@gmail.com
You should provide the service of online order also.
CONTACT
© Copyright 2018 Grill & Chill


Grill & Chill
About Contact Directions
Thanks for your suggestion.
GRILL & CHILL always at your service.


Grill&Chill
localhost:3000/about
Grill & Chill
About Contact Directions
GRILL & CHILL opened in August 2018 and is located in Saket , New Delhi. We serve food and happiness. We have been recognised for serving best food in Delhi . The menu is inspired from different cuisine's specialties and will appeal to a wide and varied clientele.
PHONE
(120)796-1993
LOCATION
E-78 Saket dist.South Delhi in Delhi 11017
HOURS
MON-FRI 11A.M - 11P.M
SAT-SUN 10A.M - 11P.M
REVIEWS and RATINGS
It's truly one of a kind restaurant.Though expensive, it's definitely worth it.
Juhi Sharma


Grill&Chill
localhost:3000/table?
Grill & Chill
About Contact Directions
Reservation
BOOK A TABLE
Arun Kumar
kumararun2@gmai.com
8762451286
30/11/2020
08:00 PM
BOOK A TABLE


Grill&Chill
your reservation is confirmed
localhost:3000/table
Grill & Chill
About Contact Directions
Reservation Confirmed
Hello, Arun Kumar your reservation is Confirmed.
Date : 2020-11-30
Time : 20:00
Grill & Chill welcomes you with an open heart

9. APPENDIX B (Source Code)

- training

Server.js

```
const express = require('express');
const app = express();
const ejs = require('ejs');
const path = require('path');
const expressLayout = require('express-ejs-layouts');
const PORT = process.env.PORT || 3000;
const mongoose = require('mongoose');


//Database Coonection
const url = 'mongodb://localhost/restro';
mongoose.connect(url, { useNewUrlParser : true,
useCreateIndex: true, useUnifiedTopology : true, useFindAndModify : true});
const connection = mongoose.connection;
connection.once('open', () => {
   console.log('Database connected');
}).catch(err => {
   console.log('Connection failed')

});

// Assets
app.use(express.static('public'))
app.use(express.urlencoded({ extended: false}))
// set Template engine
app.use(expressLayout);
app.set('views' , path.join(__dirname, '/resources/views'));
app.set('view engine', 'ejs');

//routes
require('./routes/web')(app) ;
app.use((req, res) => {
   res.status(404).send('<h1>404 page not found </h1>')
})
app.listen(PORT, () =>{
   console.log(`running on ${PORT}`)
})
```

Menu.json

```
[{

  "_id": {
    "$oid": "5eee651f739f8c674fd736ee"
  },
  "name": "Tandoori Vegetable Platter",
  "image": "",
  "price": "350"
},{
  "_id": {
    "$oid": "5eee6671a27a66807cf2bea3"
  },
  "name": "Mezze Platter",
  "image": "",
  "price": "350"
},{
  "_id": {
    "$oid": "5eee6692a27a66807cf2bea4"
  },
```

```
 "name": "Cheese Platter",
 "image": "",
 "price": "300"
},{
 "_id": {
  "$oid": "5eee66a5a27a66807cf2bea5"
 },
 "name": "Sandwich Platter",
 "image": "",
 "price": "300"
},{
 "_id": {
  "$oid": "5eee66c4a27a66807cf2bea6"
 },
 "name": "Spaghetti Napolitan",
 "image": "",
 "price": "290"
},{
 "_id": {
  "$oid": "5eee66cfa27a66807cf2bea7"
 },
 "name": "Peene Arrabbiat",
 "image": "",
 "price": "290"
},{
 "_id": {
  "$oid": "5eee66eea27a66807cf2bea8"
 },
 "name": "Penne with Mushroom and Blue Cheese  ",
 "image": "",
 "price": "350"
},{
 "_id": {
  "$oid": "5eee6717a27a66807cf2bea9"
 },
 "name": "White Sauce Pasta",
 "image": "",
 "price": "350"
},{
 "_id": {
  "$oid": "5fba36ae93382e024d722ae8"
 },
 "name": "Napolitana",
 "image": "",
 "price": "270"
},{
 "_id": {
  "$oid": "5fba36c893382e024d722ae9"
 },
 "name": "Margherita",
 "image": "",
```



```
 "price": "300"
},{
 "_id": {
  "$oid": "5fba36db93382e024d722aea"
 },
 "name": "Veggie Pizza",
 "image": "",
 "price": "325"
},{
 "_id": {
  "$oid": "5fba36f193382e024d722aeb"
```

```
  },
  "name": "Four seasons Pizza",
  "image": "",
  "price": "375"
},{
  "_id": {
    "$oid": "5fba370f93382e024d722aec"
  },
  "name": "French Fries",
  "image": "",
  "price": "270"
},{
  "_id": {
    "$oid": "5fba372293382e024d722aed"
  },
  "name": "Peri Peri Fries",
  "image": "",
  "price": "300"
},{
  "_id": {
    "$oid": "5fba373993382e024d722aee"
  },
  "name": "Chilli Potato",
  "image": "",
  "price": "325"
},{
  "_id": {
    "$oid": "5fba374d93382e024d722aef"
  },
  "name": "Jacket Potatoes",
  "image": "",
  "price": "375"
},{
  "_id": {
    "$oid": "5fba442093382e024d722af0"
  },
  "name": "Plain Naan",
  "image": "",
  "price": "25"
},{
  "_id": {
    "$oid": "5fba444293382e024d722af1"
  },
  "name": "Butter Naan",
  "image": "",
  "price": "30"
},{
  "_id": {
```



```
    "$oid": "5fba445e93382e024d722af2"
  },
  "name": "Tandoori Roti",
  "image": "",
  "price": "25 "
},{
  "_id": {
    "$oid": "5fba446f93382e024d722af3"
  },
  "name": "Rumali Roti",
  "image": "",
  "price": "30"
},{
  "_id": {
```

```
    "$oid": "5fba448393382e024d722af4"
   },
   "name": "Shahi Paneer",
   "image": "",
   "price": "190"
  },{
   "_id": {
    "$oid": "5fba44d693382e024d722af5"
   },
   "name": "Kadhai Paneer",
   "image": "",
   "price": "170"
  },{
   "_id": {
    "$oid": "5fba44fd93382e024d722af6"
   },
   "name": "Dal Makhani",
   "image": "",
   "price": "210"
  },{
   "_id": {
    "$oid": "5fba450d93382e024d722af7"
   },
   "name": "Dal Punjabi",
   "image": "",
   "price": "200"
  },{
   "_id": {
    "$oid": "5fba462293382e024d722af8"
   },
   "name": "Virgin Mojito",
   "image": "",
   "price": "130"
  },{
   "_id": {
    "$oid": "5fba463893382e024d722af9"
   },
   "name": "Blue Lagoon",
   "image": "",
   "price": "120"
  },{
   "_id": {
    "$oid": "5fba466e93382e024d722afa"
   },
   "name": "Spicy Virgin Tamarind",
```



```
   "image": "",
   "price": "140"
  },{
   "_id": {
    "$oid": "5fba469393382e024d722afb"
   },
   "name": "Blackberry",
   "image": "",
   "price": "130"
  },{
   "_id": {
    "$oid": "5fba46ab93382e024d722afc"
   },
   "name": "Chocolate Mousse",
   "image": "",
   "price": "130"
  },{
```

```
  "_id": {
    "$oid": "5fba46c093382e024d722afd"
  },
  "name": "Blueberry Cheesecake",
  "image": "",
  "price": "120"
},{
  "_id": {
    "$oid": "5fba46d593382e024d722afe"
  },
  "name": "Truffle Cake ",
  "image": "",
  "price": "140"
},{
  "_id": {
    "$oid": "5fba46e893382e024d722aff"
  },
  "name": "Chocolate Fudge Brownies",
  "image": "",
  "price": "130"
}]
```


➢ training/app/http/controllers

basicController.js

```
function homeController(){

return{
home(req, res){
res.render('home')
},
about(req, res){
res.render('about')
},
contact(req, res){
res.render('contact')
},
postContact(req, res){
```



```
res.render('suggestion')
}
}
}
module.exports = homeController
```

menuController.js

```
const Menu = require('../../models/menu')
function menuController(){
   return {
      menu(req, res){
         Menu.find().then(function(food){
            // console.log(food)
           return res.render('menu',{food : food})
         })
      }
   }
```

```
}

module.exports = menuController
```

reserveController.js

```
const Info = require("../../models/info")

function reserveController() {
   return {
      table(req, res) {
         res.render('table')
      },
      tableConfirm(req, res){
         res.render('tableconfirm')
      },
      tablenotConfirm(req, res){
         res.render('tablenotconfirm')
      },
      postTable(req, res) {
         const { name, email, phone, date, time } = req.body
         // Validate Request
         if (!name || !email || !phone || !date || !time) {
            return res.redirect('/table')
         }

         const info = new Info({
           name,
           email,
           phone,
            date,
            time
         })

         info.save().then((info) => {
            // return res.redirect('/tableconfirm')
            Info.findOne({email}).then(function(user){
               console.log(user)
               return res.render('tableconfirm', {user : user})
            })
```



```
}).catch(err => {
            return res.redirect('/tablenotconfirm')
         })
      }
   }
}
module.exports = reserveController
```

➢ training/app/models

info.js

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema

const infoSchema = new Schema({
      name: { type: String, required: true},
      email: { type: String, required: true, unique: true},
      phone: { type: Number, required: true},
      date: { type: String, required: true},
      time: { type: String, required: true}
```

```
}, { timestamps: true})

module.exports = mongoose.model('Info', infoSchema)
```

menu.js

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema

const menuSchema = new Schema({
   name: { type: String, required: true},
   image: { type: String, required: true},
  price: { type: Number, required: true}

})
```

➢ training/resources/scss

_variables.scss

```
module.exports = mongoose.model('Menu', menuSchema)


// Colours
$basic : white;
$dark : black;
$logocolour : #ba7967;
$heading: #c5b35b;
$button: #c1a57b;

// Transitions
$smooth : all .3s ease-in-out;
```

app.scss


```
@import './variables';

body{

font-family: 'Roboto', sans-serif;

 font-weight: bold;

-webkit-font-smoothing : antialiased ;
}
.header-img {
height: 700px;
background: url("https://images.pexels.com/photos/6267/menu-restaurant-vintage-table.jpg?
auto=compress&cs=tinysrgb&dpr=2&h=650&w=940");
background-size: cover;
}
.img{
color: $logocolour;
}
.logo{
font-weight: bold;
font-size: 2.5em;
```

```
font-family: 'Dancing Script', cursive;
}
.bar:hover{
transition: $smooth;
background-color: $logocolour;
opacity: 0.5;
color: $dark;
}
.title{
font-family: cursive;
font-weight: bolder;
position: absolute;
color: white;
left: 0%;
right : 0%;
top: 42%;
background-color: #323232;
opacity: 0.5;
&:hover{
opacity: 1;
}
}
.content{
margin-left: 20%;
font-family: cursive;
}
.content1{
margin-left: 30%;
font-family: cursive;
}
.reserve{
text-align: center;
position: absolute;
left: 20%;
bottom : 29% ;
&:hover{
background-color: transparent;
color: $basic;
}
}
.menu{
text-align: center;
position:absolute;
left: 35%;
bottom : 29% ;
color: $basic;
&:hover{
color: black;
background-color:$basic;
}
}
.intro{
text-align: center;
margin: 5%;
font-family: 'Amethysta', serif;
}
.details{
margin-top: 10%;
text-align: center;
}
.reviews{
padding: 4% 3%;
```

```
padding-bottom: 0.8%;
text-align: center;
background-color: #f0e3c4;
}
.text{
font-family: 'Fredoka One', cursive;
text-align: center;
color: #420000;
}
.reviews-text{
font-family: 'Amiri', serif;
color: #382039;
}
.reviews-image
{
border-radius:100% ;
width: 100px;
margin: 2%;
}
.carousel-item{
padding:2% 2%;
}
#reserve{
color: $heading;
font-family: cursive;
font-weight: bold;
font-size: 5em;
padding-top: 5%;
text-align: center;
}
.heading{
padding-top: 3%;
padding-bottom: 3%;
text-align: center;

}
.book{
margin-left: 45%;
margin-top: 5%;
background-color: $button;
color: $basic;
&:hover{
color: $dark;
background-color: $button;
}
}
.form-row, .form-group{
padding-left: 2%;
padding-right: 2%;
}
#contact{
color: $heading;
font-family: cursive;
font-weight: bold;
font-size: 5em;
text-align: center;
padding-top: 5%;
padding-bottom: 3%;
}
.cb{
margin-left: 45%;
margin-top: 3%;
background-color: $button;
```

```
color: $basic;
&:hover{
color: $dark;
background-color: $button;
}
}
#last{
text-align: center;
margin-top: 4%;
padding-top:3%;
padding-bottom: 3%;
background-color: $button;
}
.fa-2x{
margin-left: 2%;
margin-right: 2%;
color: $dark;
}
.fa-instagram:hover{
color: rgb(173, 28, 149);
}
.fa-facebook:hover {
color: blue;
}
.fa-twitter:hover{
color: #2bb2bb;
}
.fa-envelope-square:hover{
color: red;
}
```



```
#grill{
font-family: 'Dancing Script', cursive;
font-size: 1.3em;
}
#menu{
color: $heading;
font-family: cursive;
font-weight: bold;
font-size: 5em;
text-align: center;
padding-top: 2%;
}
.mcont{
text-align: center;
margin-top: 5%;
margin-bottom: 2%;
}
.card-title{
font-weight: normal;
font-size: medium;
}
.price{
font-weight: bolder;
font-size: medium;
}
.food-image{
border-radius:50% ;
float: left;
width: 100px;
height: 100px;
object-fit: cover;
```

```
}
.alert{
margin-top: 2%;
}
.notconfirm{
margin-left: 20%;
}
.box{
margin-top: 3%;
margin-left: 23%;
margin-right: 23%;
}
.box h2,h3{
font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva,
Verdana, sans-serif;
}
.box p{
color: $button;
font-weight: bolder;
font-size: 2em;
}
#confirm {
color: $heading;
font-family: cursive;
font-weight: bold;
font-size: 3em;
text-align: center;
padding-top: 2%;

}

#slide{
padding-top: 4%;
padding-left: 5%;
}
.testimonial-image
{
width: 75%;
height: 550px;
margin-left: 8px;
margin-right: none;
}
.space{
margin-bottom: 3%;
}
.place{
width: 70%;
height: 250px;
}
.map{
width: 20%;
}

@media (max-width: 769px){
.header-img {
height: 600px; //672px
}
.logo{
font-size: medium;
}
.bar{
font-size: small;
}
```

```
.navbar-expand-lg{
padding-left: 5%;
padding-bottom: 1%;
padding-top: 1%;
}
.navbar-toggler{
font-size: 0.1rem;
}
.title{
top: 38%;
}
.title h1, h3{
font-size: large;
}
.reserve{
font-size: x-small;
left: 2%
}
.menu{
font-size: small;
left: 35%;
}
.intro{
font-size: medium;
}
```



```
.details h5,p {
font-size: medium;
}
.text{
font-size: medium;
}
.reviews-text{
font-size: small;
}
.reviews-image{
width: 80px;
}
.carousel-item{
padding:0% 0%;
}
em{
font-size: small;
}
#reserve, #contact{
font-size: 3em;
}
.heading{
font-size: small;
}
.book, .cb{
margin-left: 35%;
font-size: smaller;
}
.form-row, .form-group{
padding-left: 0%;
padding-right: 0%;
}
.card-title{
font-weight: normal;
font-size: small;
}
.price{
```

```
font-weight: bold;
font-size: smaller;
}
.food-image{
width: 70px;
height: 70px;
}
#menu, #confirm{
font-size: 2em;
}
.box h2{
font-size: larger;
}
.box h3{
font-size: small;
}
.box p{
font-weight: bolder;
font-size: 1.4em;
}
```



```
.testimonial-image
{

width: 90%;
height: 0px;
margin-left: 2%;
margin-bottom: 4%;
}
}
```

- training/resources/views

about.ejs

```
<!-- ABOUT -->
<section id = "about" class="about">
<h4 class="intro"><strong> GRILL & CHILL </strong> opened in August 2018 and is located in Saket , New Delhi. We
serve food and happiness. We have been recognised for serving best food in Delhi . The menu is inspired from
different cuisine's specialties and will appeal to a wide and varied clientele.</h4>
<div class="container details">
<div class="row">
<div class="col-sm">
<h5>PHONE <i class="fas fa-phone img"></i></h5>
<p>(120)796-1993</p>
</div>
<div class="col-sm">
<h5>LOCATION <i class="fas fa-thumbtack img"></i></h5>
<p>E-78 Saket dist.South Delhi in Delhi 11017 </p>
</div>
<div class="col-sm">
<h5> HOURS <i class="fas fa-clock img"></i></h5>
<p>

<li>MON-FRI 11A.M - 11P.M</li>
<li>SAT-SUN 10A.M - 11P.M</li>
```

```
</p>
</div>
</div>
</div>
</section>
<!-- REVIEWS -->
<section class="reviews">
<div id="carouselExampleControls" class="carousel slide" data-ride="false">
<div class="carousel-inner">
<h4 class="text"> <strong>REVIEWS and RATINGS</strong> </h4>
<div class="carousel-item active">
<h2 class="reviews-text">It's truly one of a kind restaurant.Though expensive, it's definitely worth
it.<br>
</h2>
<img class="reviews-image" src="img/juhi.png" alt="">
<em>Juhi Sharma </em>
</div>
<div class="carousel-item">
```



```
<h2 class="reviews-text">Love the wonderful ambience and food. The pasta and pastries are worth
dying.<br></h2>
<img class="reviews-image" src="img/kriti.png" alt="">
<em>Kriti Kalra</em>
</div>
<div class="carousel-item">
<h2 class="reviews-text">Best Restaurant in Delhi which doesn't compromise on food
quality.<br></h2>
<img class="reviews-image" src="img/eklavya.png" alt="">
<em>Eklavya Singh</em>
</div>
</div>
<a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="sr-only">Next</span>
</a>
</div>
</section>
```

contact.ejs

```
<h5 id="contact">Contact Us</h5>
<form>
<div class="form-row">
<div class="col">
<input type="text" class="form-control change" placeholder="Name">
</div>
<div class="col">
<input type="email" class="form-control" placeholder="xyz@gmail.com">
</div>

</div>
</form>
<form>
<div class="form-group">
<label for="exampleFormControlTextarea1"></label>
```

```
<textarea class="form-control change" id="exampleFormControlTextarea1" rows="3"
placeholder="any query/suggestion"></textarea>
</div>
</form>
<form method="GET" action="/contacted">
<button type="submit" class="btn btn-light cb">CONTACT</button>
</form>
<!-- </form> -->
<footer id="last">
<p>
<a href="https://www.instagram.com"><i class="fab fa-instagram fa-2x"></i> </a>
<a href="https://www.facebook.com"> <i class="fab fa-facebook fa-2x"></i> </a>
<a href="https://www.twitter.com"> <i class="fab fa-twitter fa-2x"></i> </a>
<a href="https://www.gmail.com"> <i class="fas fa-envelope-square fa-2x"></i> </a>
</p>
<p> © Copyright 2018 <span id="grill"> Grill & Chill</span></p>
</footer>
</div>
```



```
</div>
```

home.ejs

```
<header class="header-img">
<div class="title">
<h1 class="content">Deliciousness jumping into the mouth.</h1>
<h3 class="content1">Experience the great food.</h3>
</div>
<div>
<form method="GET" action="/table">
<button type="submit" class="btn btn-light reserve">RESERVE TABLE</button>
</form>
<a class="menu" href="/menu">BROWSE MENU</a>
</div>
</header>
<section id="slide">
<div class="row">
<div class="col-lg-6 ">
<div id="carouselExampleSlidesOnly" class="carousel slide" data-ride="carousel">
<div class="carousel-inner">
<div class="carousel-item active">
<image class="testimonial-image" src="img/menu1.jpeg" alt="food">
</div>
<div class="carousel-item ">
<img class="testimonial-image" src="img/menu2.jpeg" alt="food">
</div>
<div class="carousel-item ">
<img class="testimonial-image"
src="https://images.pexels.com/photos/1586942/pexels-photo-1586942.jpeg?
auto=compress&cs=tinysrgb&dpr=2&h=650&w=940"
alt="food">
</div>
<div class="carousel-item">
<img class="testimonial-image"
src="https://images.pexels.com/photos/1373915/pexels-photo-1373915.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500">
</div>
<div class="carousel-item">

<img class="testimonial-image"
```

```
src="https://images.pexels.com/photos/2067396/pexels-photo-2067396.jpeg?
auto=compress&cs=tinysrgb&dpr=2&h=650&w=940">
</div>
<div class="carousel-item">

<img class="testimonial-image"
src="https://images.pexels.com/photos/315755/pexels-photo-315755.jpeg?
auto=compress&cs=tinysrgb&dpr=2&h=650&w=940">
</div>

<div class="carousel-item">

<img class="testimonial-image"
src="https://images.pexels.com/photos/3780469/pexels-photo-3780469.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500">
</div>
```



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

<img class="testimonial-image"
src="https://images.pexels.com/photos/4457151/pexels-photo-4457151.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500">
</div>
</div>
</div>
</div>
<div class="col-lg-6 ">
<table>
<tr>
<div id="carouselExampleSlidesOnly" class="carousel slide space" data-ride="carousel">
<div class="carousel-inner place">
<div class="carousel-item active">
<img src="https://images.pexels.com/photos/1581554/pexels-photo-1581554.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500"
class="d-block w-100" alt="...">
</div>
<div class="carousel-item">
<img src="https://images.pexels.com/photos/1639565/pexels-photo-1639565.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500"
class="d-block w-100" alt="...">
</div>
<div class="carousel-item">
<img src="https://images.pexels.com/photos/1679825/pexels-photo-1679825.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=5000"
class="d-block w-100" alt="...">
</div>
<div class="carousel-item">
<img src="https://images.pexels.com/photos/1566837/pexels-photo-1566837.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500"
class="d-block w-100" alt="...">
</div>
<div class="carousel-item">
<img src="https://images.pexels.com/photos/1267321/pexels-photo-1267321.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500" class="d-block w-100" alt="...">
</div>
</div>
</div>
<!-- <img class=""
src="https://images.pexels.com/photos/1581554/pexels-photo-1581554.jpeg?
auto=compress&cs=tinysrgb&dpr=1&w=500"> -->
</tr>
<tr>
```

```
<p class="">
<iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d28031.61550760762!
2d77.36459313955076!3d28.5712058999999992!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!
1m2!1s0x390cefeac3b88b71%3A0x387b112c9e288da7!2sGrill%20n%20Chill%20cafe!5e0!3m2!
1sen!2sin!4v1606141834072!5m2!1sen!2sin"
width="400" height="300" frameborder="0" style="border:0;" allowfullscreen=""
aria-hidden="false" tabindex="0"></iframe>
</p>
</tr>
</table>
</div>
</div>
```



```
</section>


<footer id="last">
<p>
<a href="https://www.instagram.com"><i class="fab fa-instagram fa-2x"></i> </a>
<a href="https://www.facebook.com"> <i class="fab fa-facebook fa-2x"></i> </a>
<a href="https://www.twitter.com"> <i class="fab fa-twitter fa-2x"></i> </a>
<a href="https://www.gmail.com"> <i class="fas fa-envelope-square fa-2x"></i> </a>
</p>
<p> © Copyright 2018 <span id="grill"> Grill & Chill</span></p>
</footer>
```

layout.ejs

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Grill&Chill</title>
<!-- Google Fonts -->
<link href="https://fonts.googleapis.com/css2?family=Dancing+Script:wght@700&display=swap"
rel="stylesheet">
<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@300&display=swap"
rel="stylesheet">
<link href="https://fonts.googleapis.com/css2?family=Amethysta&display=swap" rel="stylesheet">
<link href="https://fonts.googleapis.com/css2?family=Amiri&family=Fredoka+One&display=swap"
rel="stylesheet">
<!-- Bootstrap css -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-TX8t27EcRE3e/
ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">
<!-- CSS -->
<link rel="stylesheet" href="/css/app.css">
<!-- Font Awesome -->
<!-- font awesome link -->
<script defer src="https://use.fontawesome.com/releases/v5.0.7/js/all.js"></script>
</head>
<body>

<header> </header>
<nav class="navbar navbar-expand-lg navbar-light img" style="background-color: #c1a57b;">
<a class="navbar-brand logo" href="/"> <i class="fas fa-utensils " style=" color: #ba7967;"></i> Grill
& Chill</a>
```

```
<button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarTogglerDemo01"
aria-controls="navbarTogglerDemo01" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarTogglerDemo01">


<ul class="navbar-nav ml-auto">
<li class="nav-item">
<a class="nav-link bar" href="/about">About</a>
```



```
</li>
<li class="nav-item">
<a class="nav-link bar" href="/contact">Contact</a>
</li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle bar" href="#slide">Directions</a>
</li>
</ul>
</div>
</nav>
<%- body %>
<!-- Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOn+Y7RsweNrtN/
tE3MoK7ZeZDyx" crossorigin="anonymous"></script>
<!-- JS -->
<script src="/js/app.js"></script>

</body>
</html>
```

menu.ejs

```
<section>
<h1 id="menu">MENU</h1>
<div class="row">
<% food.forEach(function(food){ %>
<div class="col-6 col-lg-3">
<div class="card mcont">
<div class="card-body ">
<img class="food-image" src="/img/<%= food.image %>">
<h5 class="card-title"><%= food.name %></h5>
<h3 class="price">Rs. <%= food.price %></h3>
</div>
</div>
</div>
<% }) %>
</section>
```

suggestion.ejs

```
<div class="jumbotron  jumbotron-fluid">
<div class="container">
<h1 class="display-4">Thanks for your suggestion.</h1>
<p class="lead">GRILL & CHILL always at your service.</p>
```

```
</div>
</div>
```

table.ejs

```
<div>
<h5 id="reserve"> Reservation</h5>
<h1 class="heading">BOOK A TABLE</h1>
```



```
<form method="POST" action="/table">
<!-- <form> -->
<div class="form-row">
<div class="col">
<input name="name" type="text" class="form-control change" placeholder="Name">
</div>
<div class="col">
<input name="email" type="email" class="form-control" placeholder="xyz@gmail.com">
</div>
<div class="col">
<input name="phone" type="number" class="form-control back" placeholder="Phone No.">
</div>
</div>
<br>
<div class="form-row">
<div class="col">
<input name="date" type="date" class="form-control">
</div>
<div class="col">
<input name="time" type="time" class="form-control">
</div>
</div>
</div>
<button type="submit" class="btn btn-light book">BOOK A TABLE</button>
</form>
</div>
```

tableconfirm.ejs

```
<div class="box">
<h1 id="confirm" > Reservation Confirmed</h1>
<br>
<h2>Hello, <%= user.name %> your reservation is Confirmed.</h2>
<h3> Date : <%= user.date %> </h3>
<h3>Time : <%= user.time %> </h3>
<br>
<p>Grill & Chill welcomes you with an open heart <i class="fas fa-utensils " style=" color:
#ba7967;"></i></p>
</div>
```

tablenotconfirm.ejs

```
<div class="alert alert-danger" role="alert">
Table by this email-id is already reserved try using different id or contact restaurant
</div>
<form method="GET" action="/table">
<button type="submit" class="btn btn-outline-secondary notconfirm">RESERVE TABLE</button>
</form>
```

- training/routes

web.js

```
const basicController = require("../app/http/controllers/basicController");
const reserveController = require("../app/http/controllers/reserveController");
const menuController = require("../app/http/controllers/menuController");
```



```
function initRoutes(app){
    app.get('/' , basicController().home);
    app.get('/about' , basicController().about);
    app.get('/contact' , basicController().contact);
    app.get('/contacted' , basicController().postContact);
    app.get('/table' , reserveController().table);
    app.post('/table', reserveController().postTable);
    app.get('/tableconfirm' ,reserveController().tableConfirm)
    app.get('/tablenotconfirm' ,reserveController().tablenotConfirm)
    app.get('/menu', menuController().menu);

   }


module.exports = initRoutes
```


GURU TEGH BAHADUR INSTITUTE OF TECHNOLOGY
DELHI