MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Travelzy

PROJECT REPORT

Web Technologies Lab (DSE 3163)

Department of Data Science & Computer Applications

B. Tech Data Science
5th Semester - Batch: B1
Submitted By

Lingala Bayanna Siddhartha Reddy	200968016
Katasani Nikhith Reddy	200968044
Thiruveedula Balaji Kartheek	200968080
Atyam V V R Manoj	200968108
Madamanchi Chandra Vardhan	200968112

Mentored By:

Tojo Thomas
Assistant Professor-Senior
DSCA, MIT

Akshay Bhat
Assistant Professor-Senior
DSCA, MIT

DATE :

CERTIFICATE

This is to certify that Lingala Bayanna Siddhartha Reddy (200968016), Katasani Nikhith Reddy (200968044), T Balaji Kartheek (200968080), Atyam V V R Manoj (200968108), and M Chandra Vardhan (200968112), have completed a mini project titled “Travelzy” rightly bringing forth the competencies and skill sets they gained during the course - Web Tech Lab (DSE 3163) and thereby resulting in the culmination of this project.

Tojo Thomas	Akshay Bhat
Assistant Professor-Senior	Assistant Professor-Senior
DSCA, MIT	DSCA, MIT

Table of Contents

INTRODUCTION

This project aims to build a website where users can find restaurants in any location along with many additional details like address, contact number, email ID, location, cuisine, rating, and customer review/feedback about that particular restaurant. But we have limited the scope of this project to only 3 locations i.e., Manipal, Mangalore, and Bangalore. It also allows users to share images and details of any other place (not limited only to restaurants) that they have visited and had a wonderful experience with and would like to recommend that place to others to visit. Travelzy is an online service that provides a one-stop local platform for users to discover and visit multiple restaurants and places nearby their location by making it easy for them to explore more conveniently. Our main goal is to reach the maximum number of users and let them make use of this online platform.

MOTIVATION

The economic standing of the travel destination is improved by tourism. There is enormous potential for the development of the entire city or town with the money flowing in for the services sold in these regions.

People all over the world are fond of travel and tourism. People often find it difficult to search for the best places to travel and enjoy their vacations. To address this issue , we adopt a traveling website through which people can find places which some other people had already visited, had an amazing experience and want to suggest that place to others to visit and let them have the same experience.

Not only recommending the best places, but also a list of hygiene restaurants to stop by and enjoy tasty food. This online platform would let users check for nearby top restaurants and places by just registering themselves on the website.

OBJECTIVES

- To let people, add their recently visited places which they would like to suggest others to visit.
- To let users, discover destinations that are not shown online and share images and details of those places so that interested travelers can visit those locations.
- Also letting other users to view the exact location of that place by pointing that location in a map.
- To provide the users with a list of the finest restaurants near them.
- Enabling them to search by name/place/cuisine for their desired restaurant.
- To let users, give review/feedback for a place shared by other users.
- Create necessary APIs to integrate with the database of website.
- Build the website using popular web development frameworks like NodeJS, ExpressJS, MongoDB etc.

METHODOLOGY

AGILE

Agile is a structured and iterative approach to project management and product development, an alternative to Waterfall or traditional sequential development. It is a time-focused philosophy that allows creating a project incrementally, dividing it into small pieces. One of its main benefits is the ability to adapt and change at any step.

When we use this system, we work on "sprints". They are basically set amounts of time that can be used on certain tasks and objectives. At the end of each sprint, we'll have to analyze and assess where changes can be made and new ideas can be introduced ahead of the next sprint.

This methodology is suitable for this project because of its features to make immediate necessary changes and improvements over time.

SOFTWARE REQUIREMENTS

- **VS Code**
  - VS Code is an open-source, cross-platform source code editor, particularly in the web development community. It's fast, extensible, customizable, and has tons of features. This editor is used to write, run and debug the HTML, CSS and JavaScript code required for building our project
- **Figma**
  - Figma is a free web-based designing tool that consists of powerful and exciting features for creating UI/UX design in web/app development. It is used to create User Interface for our project.
- **Bootstrap**
  - Bootstrap is a free, open-source front-end development framework for the creation of websites and web apps. Designed to enable responsive development of mobile-first websites, Bootstrap provides a collection of syntax for template designs that are customizable and easy to use.
- **MongoDB**
  - MongoDB is a database for storing structured or unstructured data in the form of collections/documents. It uses a JSON-like format to store documents.
- **NodeJS & ExpressJS**
  - Node.js is a JavaScript runtime environment that allows JS to run outside a web browser. It is one of the most efficient cross-platform JavaScript environments that can help you build robust and effective REST APIs, mobile applications, and web applications. While Express is a flexible backend framework of Node.js

that supports the development of API by creating POST and GET requests to fetch the data from the database or an external server.

- **PassportJS**
  - Passport is authentication middleware for Node.js. Extremely flexible and modular, Passport can be dropped in to any Express-based web application. A comprehensive set of strategies support authentication using a username and password, Google, Facebook, Twitter, and more. In this project, we authenticate users by using only username and password.
- **EmbeddedJS**
  - E' stands for 'effective'. EJS is a simple templating language that lets you generate HTML markup with plain JavaScript. (Basically meaning, injecting JavaScript code in between HTML markup text).
- **Git & GitHub**

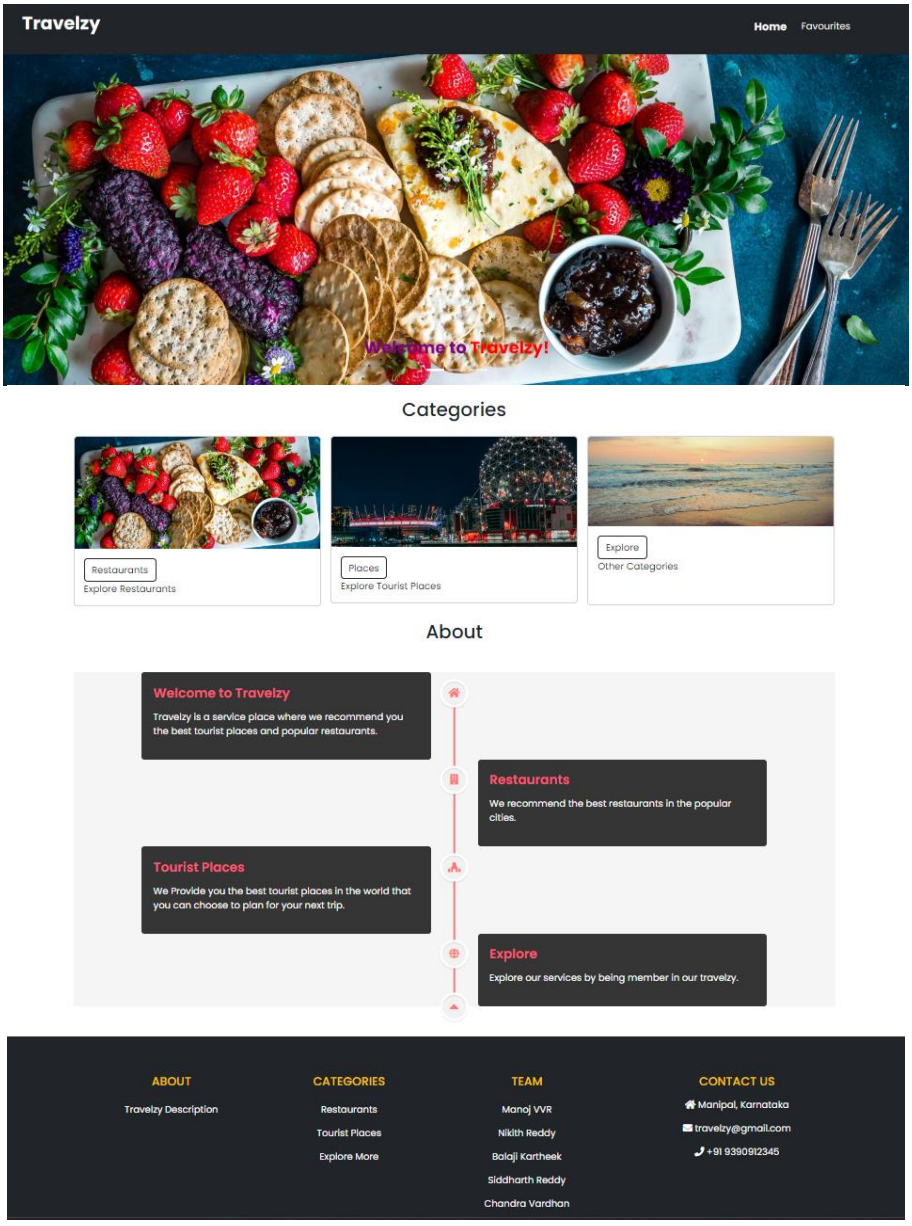
  GitHub is a website and cloud-based platform used for storing, tracking, and collaborating on software projects. It enables developers to upload their own code files and to collaborate with fellow developers on open-source projects. It assists developers in storing and managing code, as well as tracking and controlling changes to their code. While Git is a version control system that helps in managing files on the local system and cloud-based platforms like GitHub and GitLab in project development and management.
- **Cloudinary platform**
  - Cloudinary is an end-to-end image- and video-management solution for websites and mobile apps, covering everything from image and video uploads, storage, manipulations, optimizations to delivery. In this project, we use this platform to store the images uploaded by the users.

ROLES & RESPONSIBILITIES

Names	***Roles***
Siddhartha Reddy	Research & Documentation
Balaji Kartheek	Frontend Development
Nikhith Reddy	Backend Development
Madamanchi Chandra Vardhan	UI/UX Designing
Atyam V V R Manoj	Backend Development

FRONTEND

The entire frontend part of the website was built using Bootstrap (a framework of CSS). It is used in creation of websites and web apps, designed to enable responsive development of mobile-first websites and provides a collection of usable syntax for template designs that are customizable and easy to use and is based on a grid system. And its most files are pre-compiled meaning the user can just copy the required syntax from the Bootstrap documentation and use it in his/her website.

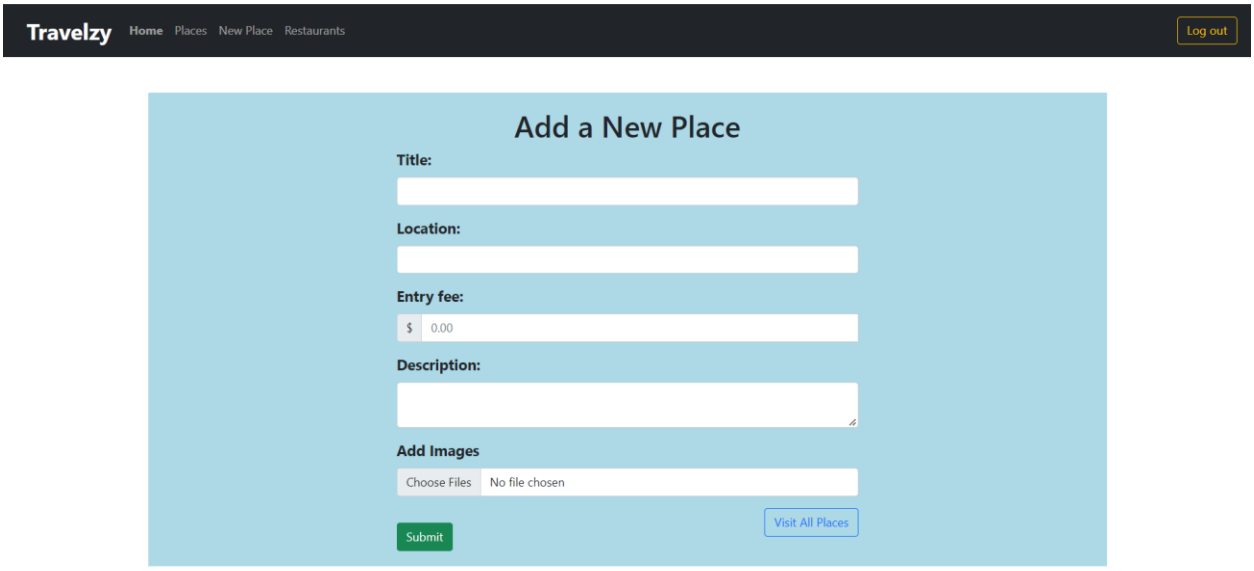


And this code is integrated with the backend by just connecting the html files with the anchor tags in the html code in bootstrap. The frontend work is done as soon as the UI/UX designs are finished. However, the website may not look exactly as designed in the UI/UX designs but as close as possible.

DATABASE

The database used in this project is MongoDB. It is a document database used to build highly available and scalable internet applications and store unstructured data in JSON-like format. We tried to integrate our database with our website using the NodeJS framework. We first created a server on Mongo and then used the Mongoose package which creates a connection to the database.

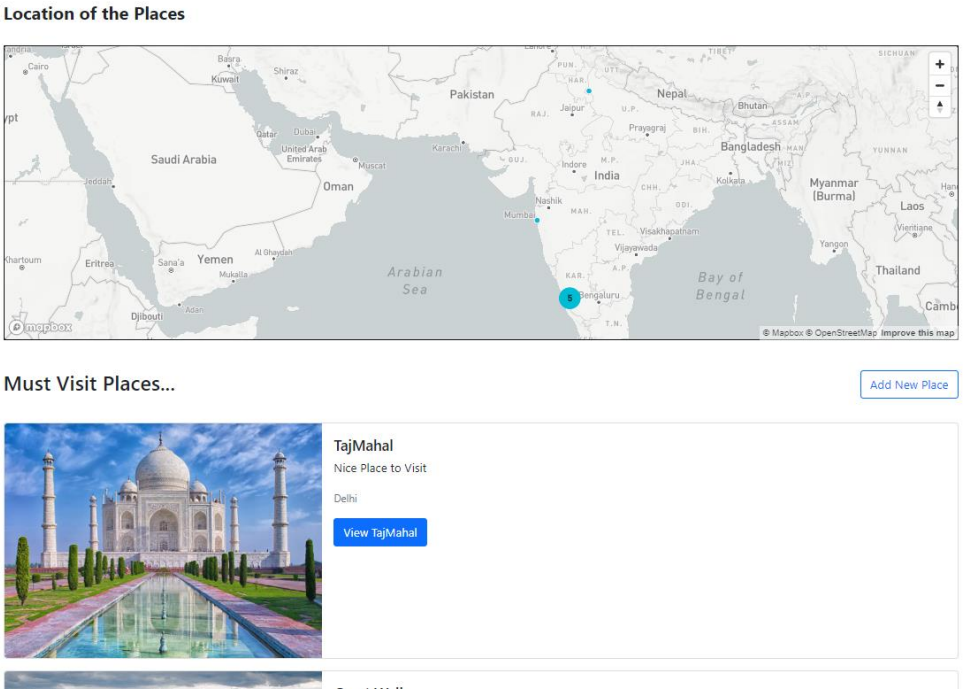
Mongoose is a node.js based ODM (Object Document Mapper) module that adds MongoDB support to the Express application. It manages relationships between data, provides schema validation and is used to translate between objects in code and representations of those objects in MongoDB.



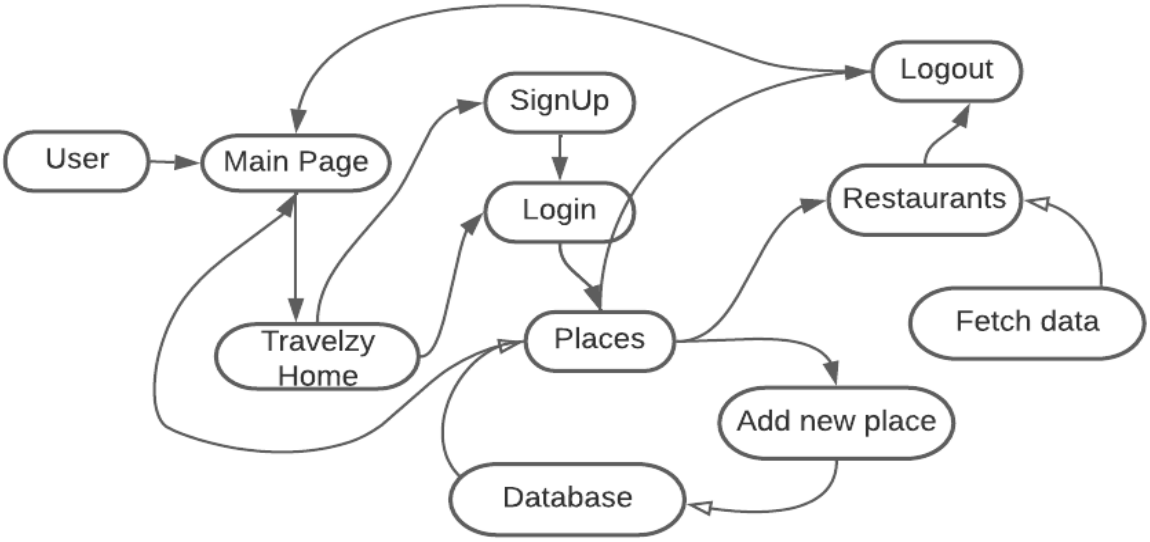
In this project, we mainly used MongoDB for storing the details of the place that the users want to share with others i.e., Title of the place, location, entry fee(if there is any), description (anything that they want mention about that place), an image of that place (only supported image formats are: ‘jpeg’, ‘png’, ‘jpg’).

Now, when other users view those places in the website and they might have already visited the same place in the past or in the present, they can give review/feedback along with a rating. And the users also have the option to delete the place they shared or the review they gave to some place.

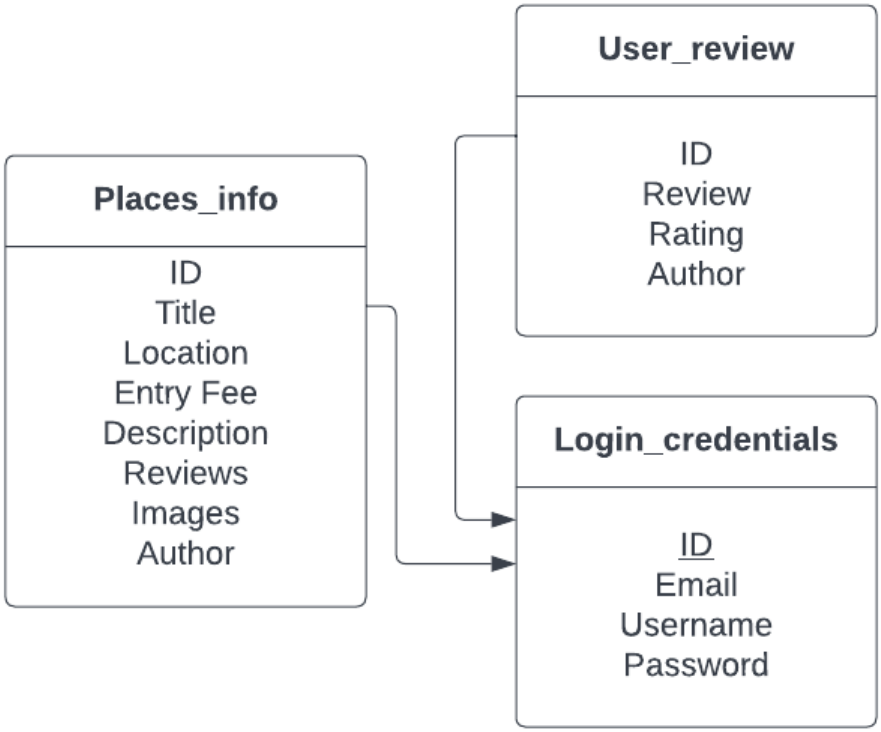
We also used Mapbox.js in our website. It provides many tools to build maps into our websites. So, the users can also view the geographical location of the places other users shared in our website.



UML Diagram



SCHEMA DIAGRAM



BACKEND

NodeJS and ExpressJS are the frameworks used in the backend for most of the part, for creating the APIs for fetching data and integrating other with other frameworks and tools wherever required. For the restaurants page, we created APIs using express and created runtime environments to run our code in the browser using node. We fetched the data from online websites and displayed them in our website by applying styling using CSS. For the remaining part of the website, we used node.js to integrate MongoDB into our website for storing the data and make connections. And we used another middleware, passport.js for the user validation and authentication. We then used a tool called Mapbox for displaying the geographical locations of the places entered by the users in maps by generating an API key and integrated it using node.js.

The following are few packages that need to be installed:

- Nodejs – runtime environment
- Express - backend framework for node js.
- cheerio - package that can be used to extract and manipulate the html tags from a website
- axios - promise based http client for the browser and node.js, makes it easy to send http requests to rest endpoints and perform crud operations (Meaning - we can use it to get post put and delete data).
- cors - node.js package that provides express.js middleware that can be used to enable calls with various options

MODULES

- **Home –** This is the main section which consists of navbar link to other sections
- **Signup/Login** - This section allows user to create an account and only valid users can add places and explore restaurants.
- **Places** – here the places are displayed along with the geographical location in the map.
- **Add places –** a user can share his experience about the place by giving the details of that place and a snapshot if possible.
- **Restaurants** – this page displays restaurants and can be filtered by searching using name or location or cuisine based on user choices.
- **Give a review** – the users can give a feedback and rating if they already visited a place shared by some other user on our website.
- **Delete a place** – the users can delete a place if they no longer want to share it or might have entered incorrect details. They also have the facility to delete a review they gave to a place shared by other user.

CONCLUSION

The platform serves as a one-stop destination for the users to find nearby restaurants and explore places giving the best tourism experience as well as hygiene restaurants located near them to have an appetite.

The final outcomes are:

- Successfully established a connection with the MongoDB database.
- Created APIs using Express and Node frameworks to fetch restaurant data directly from the online servers to display them in our website.
- Created a search module to filter restaurants by name/location/cuisine.
- Implemented successful login authentication using passport.js.
- Added an additional functionality of showing the geographical locations of the user input places using Mapbox.
- Came across many new tools and technologies as part of this project and learnt to work collaboratively as a team by splitting total work among ourselves and communicate with each other trying to achieve the same final output.

FUTURE SCOPE FOR IMPROVEMENT

- Can try to improve the user experience by displaying restaurants from all the places by not limiting the locations to only 3.
- Can try to make appropriate changes to the authentication page and correct a few mistakes while displaying the messages.
- Try to use more meaningful variable names while writing the code.
- Recommend restaurants not just by place but by popularity by considering the rating and feedback given by users.
- Try to implement the authentication using other options like using Google, Twitter, Facebook etc.

REFERENCES

- https://nodejs.org/en/docs/
- https://expressjs.com/
- https://www.mongodb.com/docs/
- https://figma.com/
- https://www.youtube.com/watch?v=1wXYg8Eslnc&list=WL&index=8&t=855s
- https://www.youtube.com/watch?v=9FQrFah9rnc
- https://www.youtube.com/watch?v=lYVKbAn5Od0
- https://www.youtube.com/watch?v=o256D3tztV8