**Project Synopsis**

1. **Title**

RateMyFood

1. **Statement about the problem**

To find hygienic local food stalls and small-sized restaurants which provide quality food and services when visiting other localities or area is a common problem people often face due to lack of knowledge about that area. A lot of time is invested to find such places and even though we do find such places the results are quite disappointing.

1. **Why this topic?**

People often suffer from the problem of finding local food stalls which provide quality food and services. With the help of system, good local food stalls and small restaurants will get recognition and it will help them to generate revenue. The users will save quite the time alongside with good quality food.

1. **Objective and Scope**
   1. **Objective**

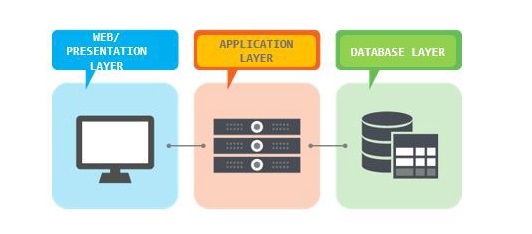
* To make accessible platform with good user experience to users which will help them to find quality local stalls and restaurants.
* Restaurants will be able to register on the platform.
* To make information of restaurants, local food stalls available to users for particular locality.
* Users can find quality restaurants and can review the restaurants.
  1. **Scope**
* The website will be available in Mumbai, later it can be expanded.
* The website will be available to local food stalls and small restaurants.

1. **Methodology**

To develop this system, I am going to use the Incremental model.

1. **Proposed Architecture**

This system will work on three-tier architecture, the web-based application.

****

***Figure 1.2 Proposed Architecture***

1. **Requirements**
   1. **Software Requirements**

* **Frontend -** HTML, CSS and ReactJS
* **Backend -** ExpressJS
* **Database –** MongoDB
* **Operating System -** Windows 7 or higher
  1. **Hardware Requirements**
* **Processor -** Intel Pentinum or higher, AMD Ryzen
* **RAM -** 1 GB or more
* Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

1. **Platform**

Visual Studio Code

1. **Contribution**

RateMyFood website helps in saving time and efforts to find hygienic and quality food. It also provides a marketing platform for local food stalls and small-sized restaurants. Both the customers and the shop keepers can use the website to shorten the distance between them and get feedbacks for their improvements.

**Chapter 1: Introduction**

* 1. **Background**

The inspiration is taken from yelp. Yelp is a review website for various services. User can see basic information about particular service provider. They can see various photos regarding that service provider’s shop. User can see reviews given by other users for a specific service. They can also add review from their experience about that shop.

* 1. **Objectives**
* Users can create an account and can see various local food stalls and restaurants of particular area.
* Restaurants can create an account and add their basic information.
* Users can review particular restaurants by giving them ratings.
* Users can also add text, photos based reviews about restaurants.
* Restaurants with maximum good reviews will top the list.
* Based on the reviews given by users, website will provide some points to users which will get converted into coupons.
  1. **Purpose, Scope and Availability**
     1. **Purpose**

Purpose of this system is to solve the problem of finding local food stalls and restaurants which provide quality food and services. This system will save time of users and they can enjoy quality food without too much work. To give recognition to restaurants which provide quality food and services is also one of the purposes of this system.

* + 1. **Scope**
* The website will be available to local food stalls and small restaurants.
* The website will be available in Mumbai, later it can be expanded.
* This website will be compatible to all digital devices like phone, laptop, tablet, etc.
  + 1. **Applicability**

RateMyFood website would make finding for street food easier. Anyone with devices like smartphone, tablets, laptops and decent internet connection can use this website. This website is more applicable in cities and populated places around center of attractions.

**Chapter 2: Survey Of Technologies**

1. **AngularJS**

AngularJS is a free and open source JavaScript -based web framework for developing single page applications. It was maintained mainly by Google and a community of individuals and corporations. It aimed to simplify both the development and the testing of such applications by providing a framework for client-side model-view-controller (MVC) and model-view-viewmodel (MVVM) architectures, along with components commonly used in web applications and Progressive web applications.

1. **PHP**

PHP is a general-purpose scripting language geared toward web development. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Additionally, PHP can be used for many programming tasks outside the web context, such as standalone graphical applications and robotic drone control. PHP code can also be directly executed from the command line.

1. **PostgreSQL**

PostgreSQL is an advanced, enterprise class open source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is a highly stable database management system, backed by more than 20 years of community development which has contributed to its high levels of resilience, integrity, and correctness. PostgreSQL is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications. PostgreSQL has a rich history for support of advanced data types, and supports a level of performance optimization that is common across its commercial database counterparts, like Oracle and SQL Server. PostgreSQL's dedicated community of contributors and enthusiasts regularly find bugs and fixes, contributing to the overall security of the database system.

1. **Svelte**

Svelte is a free and open source front-end compiler. Svelte is not a monolithic JavaScript library imported by applications instead, Svelte compiles HTML templates to specialized code that manipulates the DOM directly, which may reduce the size of transferred files and give better client performance, application code is also processed by the compiler, inserting calls to automatically recomputed dataand re-render UI elements when the data they depend on is modified.This also avoids the overhead associated with *runtime* intermediate representations, such as virtual DOM unlike traditional frameworks (such as React and Vue) which carry out the bulk of their work at runtime, *i.e.* in the browser. The compiler itself is written in TypeScript.

1. **Node.js**

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on a JavaScript Engine (i.e. V8 Engine) and executes JavaScript code outside a web-browser, which was designed to build scalable network applications. 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-side and client-side scripts.

**Why these technologies?**

* **Frontend – ReactJS (JavaScript Framework)**

ReactJS is flexible as we can use it on a vast variety of platforms to build quality user interfaces. ReactJS has a great developer experience. RaectJS has META’s support as well as broader community support which will help this system solve many problems easily. ReactJS has great performance and it is easy to test.

* **Backend – ExpressJS (Node.js Framework)**

ExpressJS acts only as a thin layer of core web application features. ExpressJS has no out of the box object relational mapping or templating engine. It strives to put control in the developer’s hands and make web application development for Node.js easier. This freedom, coupled with lighting fast setup and the pure JavaScript environment of Node.js makes ExpressJS a strong candidate for development.

**Chapter 3: Requirement And Analysis**

* 1. **Problem Definition**

The common problem people face is to find hygienic food stalls and restaurants which provide quality food and services when visiting other localities. RateMyFood will make finding for street food easier. User will be able to see basic information about local stalls. They can also add review from their experience.

**Sub Problems –**

1. People face problem in finding good food stalls.

It is a common problem to find hygienic, small-sized restaurants and local food stalls providing quality food and services. While visiting new places people often face this problem. This problem costs lot of user’s quality time and efforts. The system will solve this problem by suggesting user a good local food stall according to their location and cuisine, based on other people’s reviews.

1. Good local food stalls does not get recognition.

There are lot of good local food stalls and small-sized restaurants available in each and every locality. But these stalls often miss out on opportunities due to lack of recognition. The system will solve this problem by providing recognition to such stalls and restaurants. Local stalls with maximum good reviews from customers will top the list. This will help good food stalls get recognition among people.

* 1. **Requirement Specification**
     1. **Requirement Gathering**
* **One-on-one interview**

This is one of the most effective techniques for requirement gathering. In this method, the system creator talks to the users who are very close to the problem of the project system creator working on. It is the responsibility of the interviewer to extract relevant information. Below are the tips for conducting productive interviews –

* Ask open-ended questions
* Interview the right person
* Share the question ahead of time

For this system to gather requirements, one-on-one interviews have been done.

* **Brainstorming**

Thisisa common technique used early in a project, because it acts as a starting point of a project. With brainstorming, we gather as many ideas as possible to identify, categorize and assign tasks, opportunities and potential solutions quickly. In brainstorming sessions, it is important to take notes on generated problems and ideas.

To gather requirements for RateMyFood website brainstorming sessions have been conducted.

* **Studying similar system (yelp.com, travelinsider.com, etc.)**

This technique also helps to gather requirements for the system. We can save time by analyzing existing similar systems which provide very valuable information. This technique gives what we need in our system by analyzing existing similar system’s pros and cons.

This technique has been used to gather information for RateMyFood.

**Some questions asked during interviews and brainstorming sessions –**

* + Where do you prefer to eat while you are on a trip?
  + Do you find any difficulties in searching for good food? If yes, then how you tackle this problem?
  + What do you consider a good food service in this industry?
  + Which format will you prefer for posting a review?
  + Do you prefer to register yourself on this type of promotion platform?
    1. **Requirement Analysis**
       1. **Functional Requirements**

1. Two types of accounts, one for common user and another for restaurants.
2. Ratings and Reviews

Users can add reviews in form of plain text, images, etc. and can give ratings to the various restaurants.

1. Form for Restaurants

A basic form for restaurants where they can add their basic information such as name, photos, address, contact info, restaurant menu, etc.

1. Like and Comments

Users can like and comment other users’ reviews.

1. Mobile Friendliness

Most of the users are mobile users so website should be mobile friendly.

* + - 1. **Non-Functional Requirements**

1. Security

Restaurant’s and user’s data should be safe. There must be authentication system for website to access.

1. Usability

Website’s interface should be clean, good-looking and most importantly easy to access even for non-technical users.

1. Scalability

Website should work as expected even when it scales.

1. Performance

Focus should be on loading every page of a site as fast as possible regardless of the number of integrations and traffic on the site.

* + - 1. **System Requirements**

1. **Login**

Input: Username and Password

Source: User

Output: Logged in successfully

Destination: Database

Action: After validation of the information provided by user, user will get access to their account on the website

Pre-Condition: User must provide username and password

Post-Condition: Redirected to home page if credentials are valid

1. **Register**

Input: Name, Email, Phone Number, Password, Confirm Password, etc.

Source: User

Output: registered successfully

Destination: Database

Action: After successful registration, separate account for user will be created

Pre-Condition: User must provide all the fields in the form for registration

Post-Condition: Redirected to Login if details are valid and correct

1. **Location Permission**

Input: Dialog box for GPS permission (enable/disable)

Source: User

Output: Location permission granted successfully

Destination: Database

Action: Restaurants in that particular location will be displayed

Pre-Condition: Internet connectivity, GPS enabled device

Post-Condition: Location permission granted successfully

1. **File Permission**

Input: Dialog box for file access permission (enable/disable)

Source: User

Output: File permission granted successfully

Destination: Database

Action: List of files in the device will be displayed

Pre-Condition: File format should be correct, depending on tasks

Post-Condition: File size

1. **Post a review**

Input: Text, images and ratings

Source: User

Output: Review posted successfully

Destination: Database

Action: Review posted by user will be displayed under that particular restaurant’s section

Pre-Condition: Review must be written in correct format

Post-Condition: Displaying the review under restaurant’s section

* 1. **Planning and Scheduling**
     1. **Activity Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Activity**  **No.** | **Chapter**  **No.** | **Chapter Name** | **Start Date** | **End Date** |
| A1 | 0 | Project Synopsis | 15-06-2022 | 18-06-2022 |
| A2 | 1 | Introduction | 20-06-2022 | 27-06-2022 |
| 2 | Survey of Technologies |
| A3 | 3 | Requirements and Analysis | 27-06-2022 | 04-07-2022 |
| 3.1 | Problem Definition |
| A4 | 3.2 | Requirement Specification | 04-07-2022 | 11-07-2022 |
| 3.2.1 | Requirement Gathering |
| 3.2.2 | Requirement Analysis |
| A5 | 3.2.3 | System Requirements | 11-07-2022 | 18-07-2022 |
| A6 | 3.3 | Planning and Scheduling | 18-07-2022 | 29-07-2022 |
| 3.4 | Software and Hardware Requirements |
| 3.5 | Conceptual Models |
| 3.5.1 | Data Model |
| A7 | 3.5.2 | Data Flow Diagram | 01-08-2022 | 18-08-2022 |
| A8 | 3.5.3 | Class Diagram | 22-08-2022 | 25-08-2022 |
| A9 | 3.5.4 | Use Case Diagram | 29-08-2022 | 08-09-2022 |
| A10 | 3.5.5 | Sequence Diagram | 12-09-2022 | 15-09-2022 |
| A11 | 3.5.6 | Activity Diagram | 14-09-2022 | 17-09-2022 |
| 3.5.7 | State Diagram |

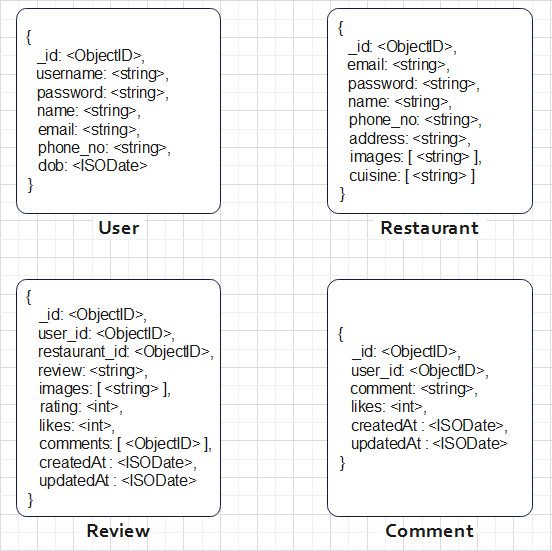
***Table 3.1 Activity Table***

* + 1. **Gantt Chart**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activity No.** | **June** | | | | | | | **July** | | | | | | | | | | **August** | | | | | | | | **September** | | | | | | | |
| 15 | | 20 | | 25 | | 30 | 05 | 10 | | 15 | | 20 | | 25 | 30 | | 05 | 10 | 15 | 20 | | 25 | | 30 | 05 | 10 | | 15 | | 20 | 25 | 30 |
| A1 |  |  |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
|  |  |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
| A2 |  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
|  | |  |  |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
| A3 |  | |  | |  |  |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
|  | |  | |  |  |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
| A4 |  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
|  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
| A5 |  | |  | |  | |  |  |  |  |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
|  | |  | |  | |  |  |  |  |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
| A6 |  | |  | |  | |  |  |  | |  |  | |  |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
|  | |  | |  | |  |  |  | |  |  | | | |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |
| A7 |  | |  | |  | |  |  |  | |  | |  | |  |  |  | | | |  | |  | |  |  |  | |  | |  |  |  |
|  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  | |  | |  | |  |  |  | |  | |  |  |  |
| A8 |  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  |  | |  |  |  |  | |  | |  |  |  |
|  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  |  |  | |  |  |  | |  | |  |  |  |
| A9 |  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  | |  | |  | |  |  |  |
|  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  | |  | |  | |  |  |  |
| A10 |  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  |  |  | |  |  |  |
|  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  |  |  | |  |  |  |
| A11 |  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  |  |  |  |  |
|  | |  | |  | |  |  |  | |  | |  | |  |  | |  |  |  |  | |  | |  |  |  | |  | |  |  |  |

***Figure 3.1 Gantt Chart***

* 1. **Software and Hardware Requirements**
     1. **Software Requirements**
* **Frontend -** HTML, CSS, ReactJS
* **Backend -** ExpressJS
* **Database -** MongoDB
* **Operating System** -Windows 7 or higher, Linux or Mac OS
  + 1. **Hardware Requirements**
* **Processor -** Intel Pentinum or higher
* **RAM -** 1GB or more
* **Monitor** **-** 17 CRT or LCD, Plasma etc.
* **Hard-Disk** **-** 256 or more (SSD preferable)
  1. **Conceptual Models**
     1. **Data Model**

****