**Technical Requirement Document for Magic Website for business**

**Purpose and Aim:-**

The aim is to build a simple-to-use website which is visually appealing and ready to use for a small scale business who have a very small base of users (upto couple of thousands) and people who have just come to the Internet to start their small businesses amid the Pandemic.

**Technical Requirements:-**

|  |  |  |
| --- | --- | --- |
| SL No. | Requirements | Specific Description |
|  |  |  |
| 1 | Python Language | The Website is coded with it |
| 2 | Django Framework | This Framework is used for a lot of features in the website |
| 3 | Bootstrap | The Frontend is handled by Bootstrap to make it look Appealing |
| 4 | Sqlite3db | This simple database is used to handle the user’s data |
| 5 | Django’s Admin | Django’s Built-in Admin panel is used in this website |
| 6 | Django’s Message Framework | Django’s Message Framework is used to give alert messages |
|  |  |  |

**Detailed Technical Requirements :-**

**Python Language:-**

The reason for using Python is that it’s syntax is easy to handle and maintain which makes it more readable and maintainable.

Python is a**high-level programming language** that has English-like syntax.

It is a very **productive language.** Due to the simplicity of Python, developers can focus on solving the problem.

They don’t need to spend too much time in understanding the **syntax** or **behaviour** of the programming language. You write less code and get more things done

* **Django Framework :-**

Django makes it easier to build better Web apps more quickly and with less code.

It is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. It was designed to help developers take applications from concept to completion as quickly as possible. It takes security seriously and helps developers avoid many common security mistakes. Some of the busiest sites on the Web leverage Django’s ability to quickly and flexibly scale. It comes with built-in Modules of Authentication , Admin etc.

* **Bootstrap :-**

Bootstrap helps in Quickly design and customize responsive mobile-first sites with Bootstrap, the world’s most popular front-end open source toolkit, featuring Sass variables and mixins , responsive grid system, extensive prebuilt components, and powerful JavaScript plugins. Most of the Frontend work has been handled by the bootstrap snippets . Some of the features coded using Bootstrap in this Website are Navigation Bar , Carousel , form, custom albums etc. The Snippets contain HTML, CSS and Java-scripts to pull out the required frontend of the website.

* **Sqlite3 database by Python :-**

SQLite is a C library that provides a lightweight disk-based database that doesn’t require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It’s also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle. This lightweight database has been used for handling the Data of the users/customers of the small scale industries .

* **Django’s Admin :-**

One of the most powerful parts of Django is the automatic admin interface. It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. The admin’s recommended use is limited to an organization’s internal management tool. It’s not intended for building your entire front end around. The admin has many hooks for customization, but beware of trying to use those hooks exclusively. If you need to provide a more process-centric interface that abstracts away the implementation details of database tables and fields, then it’s probably one has to write one’s own views. We have taken help of this Admin site that comes in-built in Django.

* **Django’s Message :-**

Quite commonly in web applications, you need to display a one-time notification message (also known as “flash message”) to the user after processing a form or some other types of user input. For this, Django provides full support for cookie- and session-based messaging, for both anonymous and authenticated users. The messages framework allows you to temporarily store messages in one request and retrieve them for display in a subsequent request (usually the next one). Every message is tagged with a specific **level** that determines its priority (e.g., **info**, **warning**, or **error**). Here we have used this feature of Django to provide us the Successful Submission Flash Message after every Form Submission.

**Website Overview :-**

The Websites has the following Features :-

Home Page :- Displays a Carousel with 3 images of the business. The Homepage also features a lot of other options to navigate to , with a nav bar on the top which has a Home button , About us , Services and Contact us. The page also shows up 3 thumbnails of Major Category of anything the business wants .

About us :- This Page contains all info about the business starting from the products , services extended to the details of the business and basically how beneficial people find it and other important social media icons and links with thumbnail and most importantly the contact details as well as the office address and email and phone number .

Services :- The Services page shows the extended services by the Business . Each service (thumbnail) is clickable which will again direct you to the showing the Real show Videos of a particular type for reference purposes before heading to contact page for booking/Purchase the Show/tickets as well as show up a form to do the same with a payment gateway.

Contact us :- This is the heart of the website . This page shows up a contact image on the top with a single image carousel and a Form that intakes the details of the customer and stores it to the database which can be CRUD by the Admin using the Django admin page that I have customized a bit for the Business . The data is stored to sqlite3 DB.

Payment Gateway :- The Contact Page is Equipped with a Payment Gateway fully secured by Razorpay . Customer can click on the Payment Button to Directly pay in an instant and book shows hassle-free . The payment details will be shown in the Razorpay Dashboard .

**Target Market:-**

This Website is intentionally made simple and with basic features as the aim is to make it as simple as possible for the user (lay-man) who has just come to the internet amid the Pandemic for extending his/her business opportunities and income.

Therefore the Target market is Small Scale Businesses all over the Country who need a Cost-friendly website with minimum basic functionality that the user can get used to from DAY 1 for just getting started with the online sector of his/her Business .