

CHAPTER 1:

INTRODUCTION

Now a day's a huge number of people are purchasing their products online. Therefore, it is necessary for companies to advertise their products online. Online advertising is a very complicated task we need a system to manage it. This project shows how companies like Google, Microsoft and amazon maintain databases of advertisers and users who browse the internet using their apps or websites. Usually, These Databases are used by companies to answer questions like 'Which product is selling the most?', 'Which category of users are most likely to buy the product?', 'What type of ads do most people like?', and so on.

1.1 PROBLEM STATEMENT

To design and develop a Ad's database management system that maintains and stores the information on websites, website owners, advertisers, ads, and users. It will handle all aspects of advertising on online websites.

1.2 OBJECTIVE

- To store the information and details related to the ad's
- To retrieve the stored data on the ad's.
- To update the stored data on the ad's.
- To provide effective and efficient management of ad's database.
- To delete the stored data on the ad's.

1.3 SCOPE OF PROJECT

It can be used by various companies to store and use the data associated with the ads being displayed on various online websites.

CHAPTER 2:

SYSTEM REQUIREMENTS AND SPECIFICATIONS

2.1 HARDWARE REQUIREMENTS

- Computers are intel, Ryzen or compatible Pentium 333 MHZ or higher.
- Memory (or RAM) has 64 MB minimum, 128MB recommended.
- Hard disk space should have 1GB for the database and the client software. this requirement may increase with the increase in records.
- The monitor has VGA or higher resolution required for graphical tools.
- The pointing device has Microsoft mouse or compatible.

2.2 SOFTWARE REQUIREMENTS

- Microsoft Windows XP/ Windows 7/ Windows Vista/ Windows 8/ Windows 10
- Jupyter Notebook should be installed.
- Oracle Database 10G Express Edition should be installed.
- Python Tkinter Module should be installed for GUI.

2.3 FUNCTIONAL REQUIREMENTS

- User Interface and User Experience: The system is easy to understand and use. A native user can also use the system effectively without any difficulties.
- Safety and robustness: The system is able to avoid or tackle disastrous actions. In other words, it should be fool proof.
- Error handling: Response to user errors and undesirable situations has been taken care of to ensure that the system operates without halting.
- Response time: The response time of all operations is good. This has been made possible by careful programming.

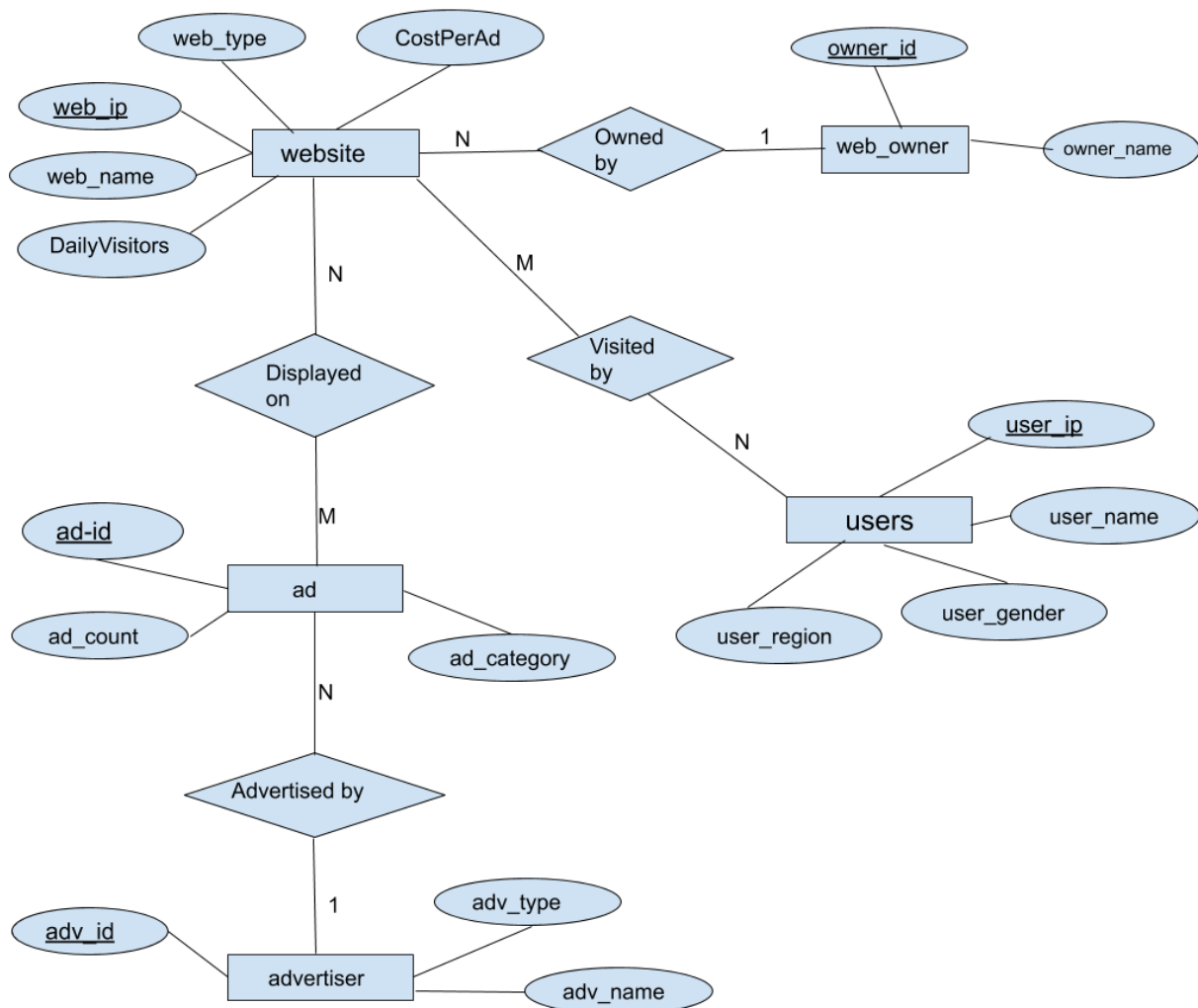
2.4 NON-FUNCTIONAL REQUIREMENTS

- Security: The system's backend server shall only be accessible to authenticated management.
- Reliability: The reliability of the overall project depends on the reliability of the separate components. The main pillar of reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes.
- Availability: The system should be available all the time. A user-friendly system should work 24 hours without any hindrances.
- Supportability: The code and supporting models of the system are well documented and easy to understand.

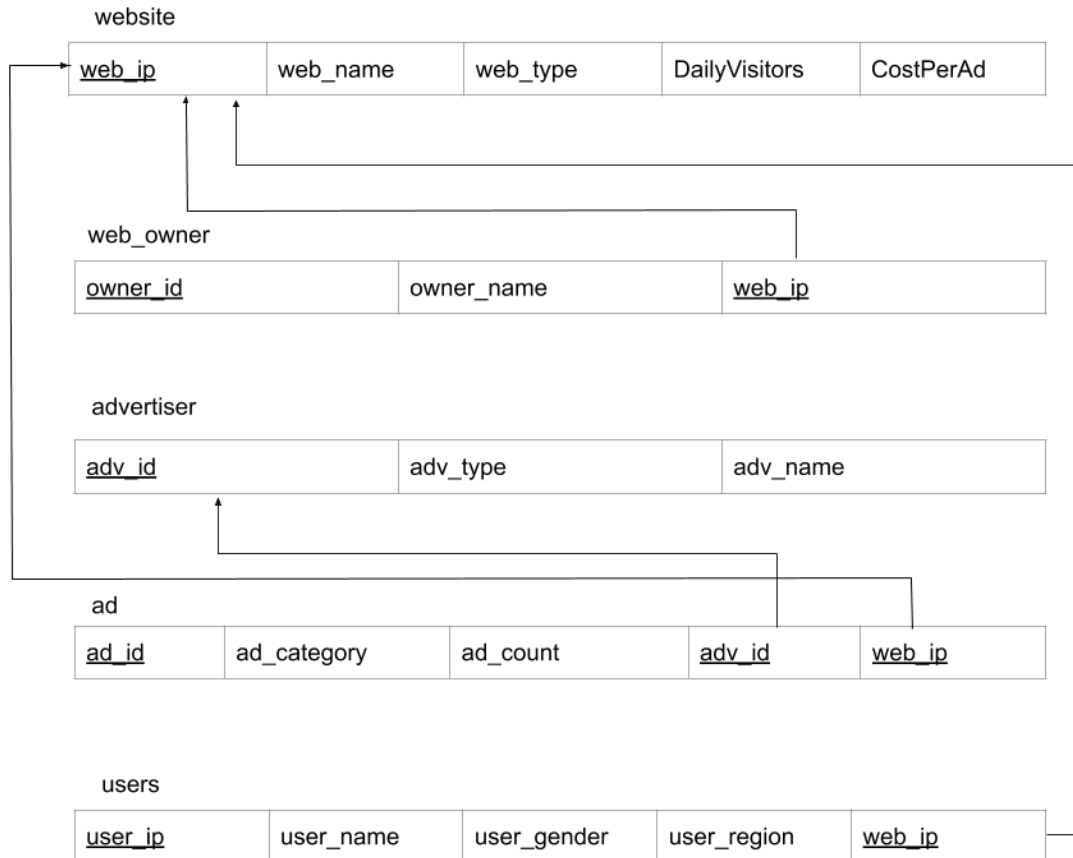
CHAPTER 3:

SYSTEM DESIGN

3.1 ENTITY-RELATIONSHIP DIAGRAM



3.2 SCHEMA DIAGRAM



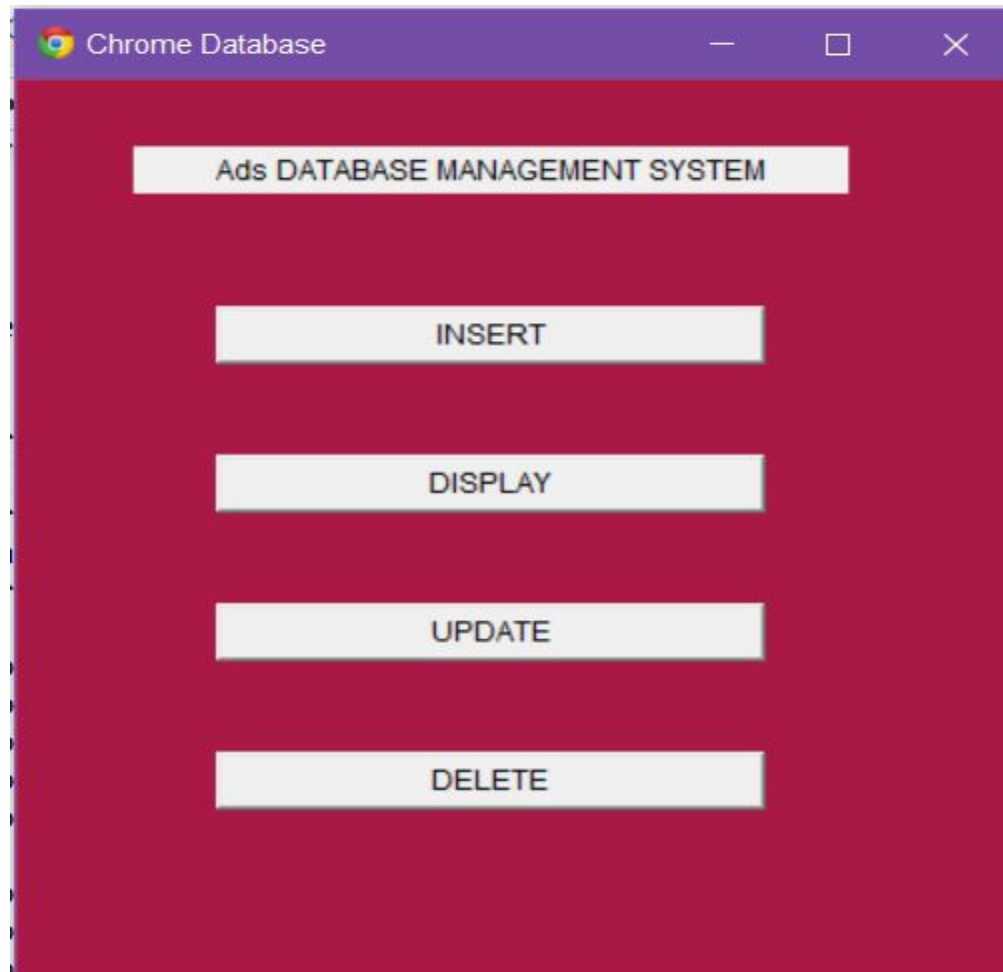
CHAPTER 4:

IMPLEMENTATION

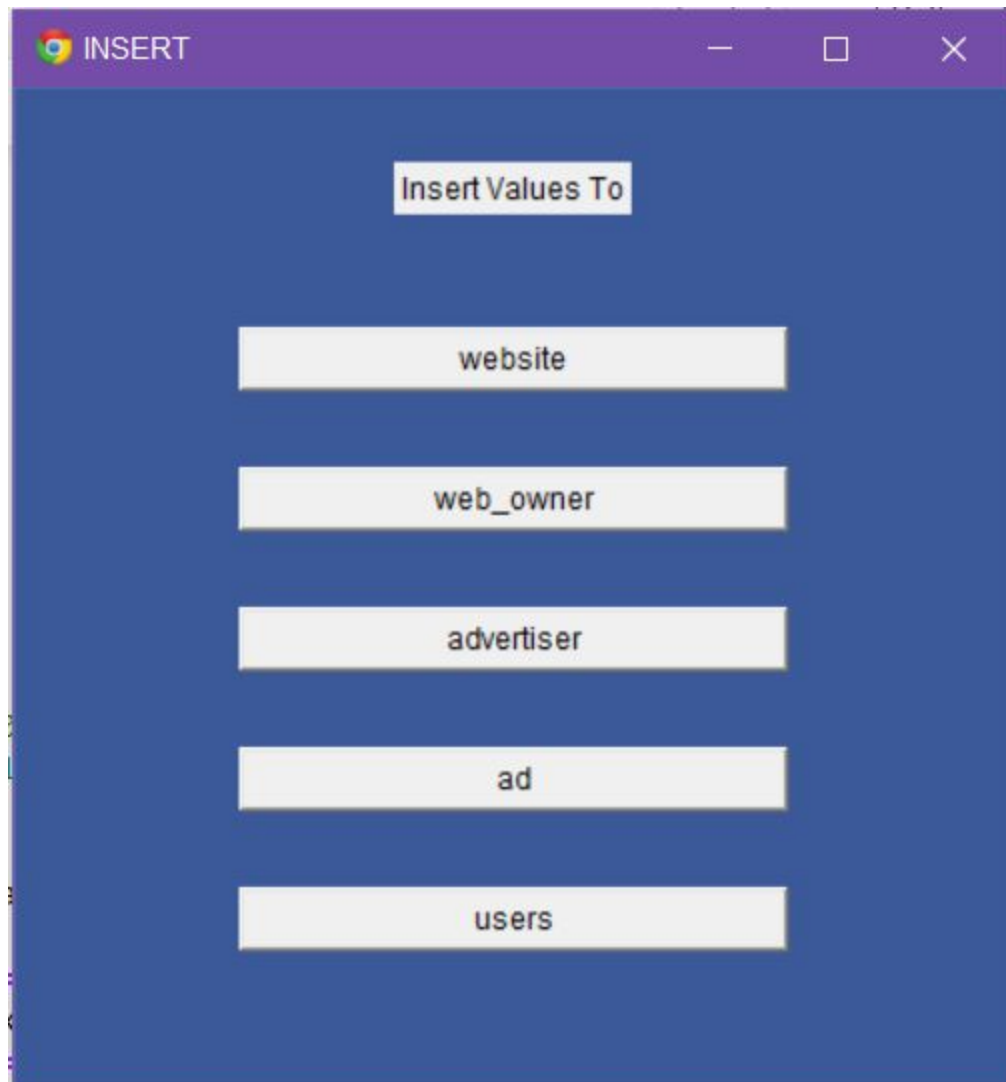
4.1 MODULE DESCRIPTION

- **website:** This module is used to store the information associated with a website. Here we maintain data such as web_ip, web_type, web_name, DailyVisitors, CostPerAd.
Input: web_ip, web_type, web_name, DailyVisitors, CostPerAd.
Output: We can insert, delete, update and view all the details of websites.
- **web_owner:** This module is used to store the information associated with a website owner. Here we maintain data such as owner_id, owner_name and respective websites he/she owns.
Input: owner_id, owner_name, web_ip.
Output: We can insert, delete, update and view all the details of web_owner.
- **advertiser:** This module is used to store the information associated with an advertiser. Here we maintain data such as adv_id, adv_name, adv_type.
Input: adv_id, adv_name, adv_type.
Output: We can insert, delete, update and view all the details of advertisers.
- **ad:** This module is used to store the information associated with an Ad. Here we maintain data such as ad_id, ad_category, ad_count, and respective adv_id and web_ip.
Input: ad_id, ad_category, ad_count, and respective adv_id and web_ip.
Output: We can insert, delete, update and view all the details of Ads.
- **users:** This module is used to store the information associated with a user. Here we maintain data such as user_ip, user_name, user_region, and respective web_ip.
Input: user_ip, user_name, user_region, web_ip.
Output: We can insert, delete, update and view all the details of users.

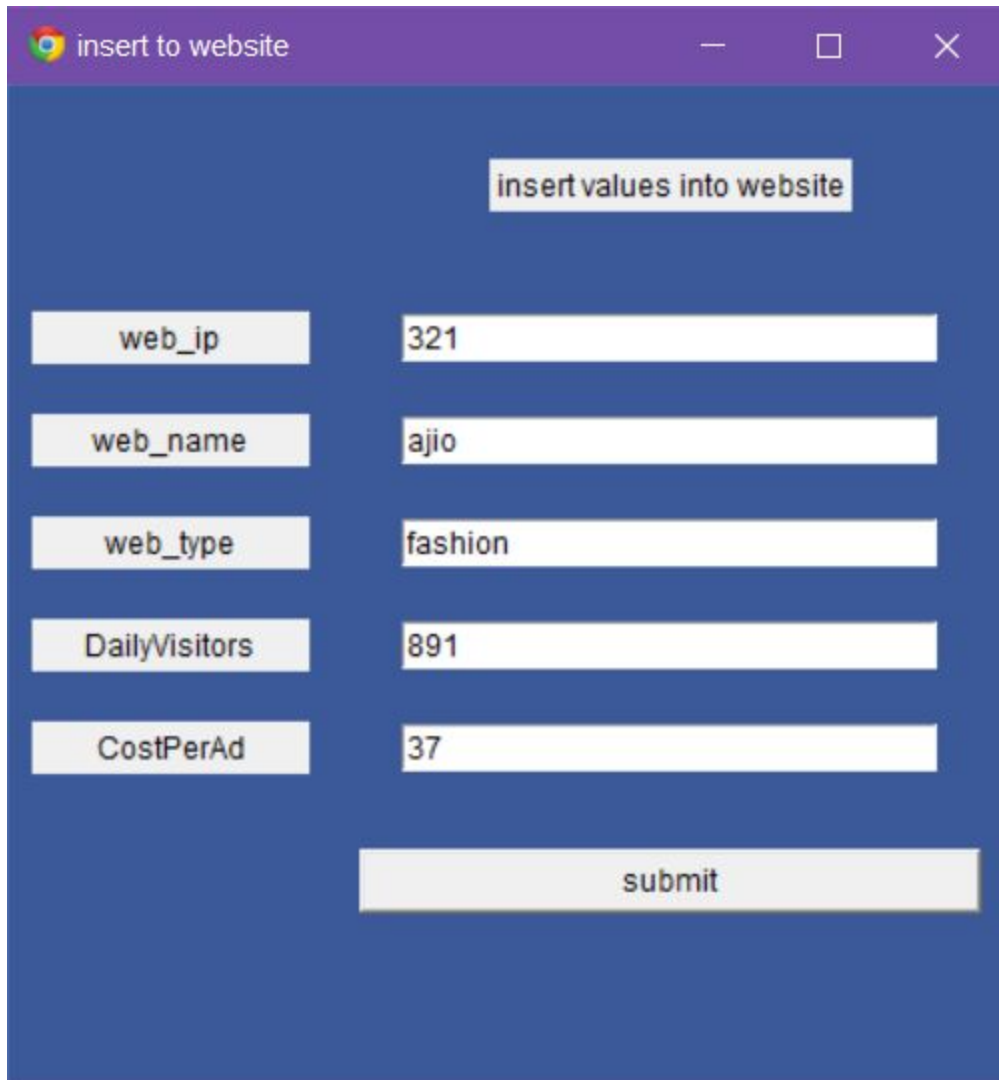
4.2 RESULT DESCRIPTION



FIG(1): Main Menu Page



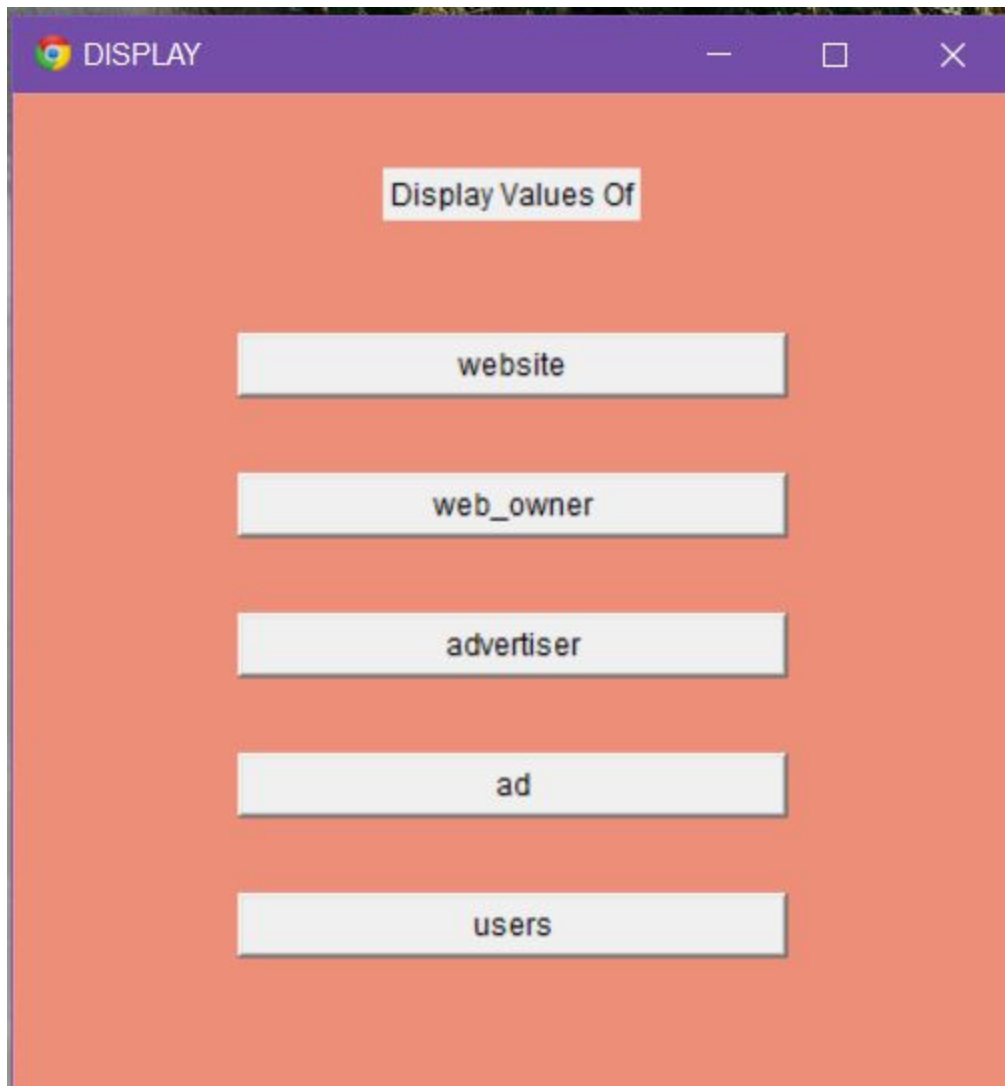
FIG(2): INSERT MENU PAGE



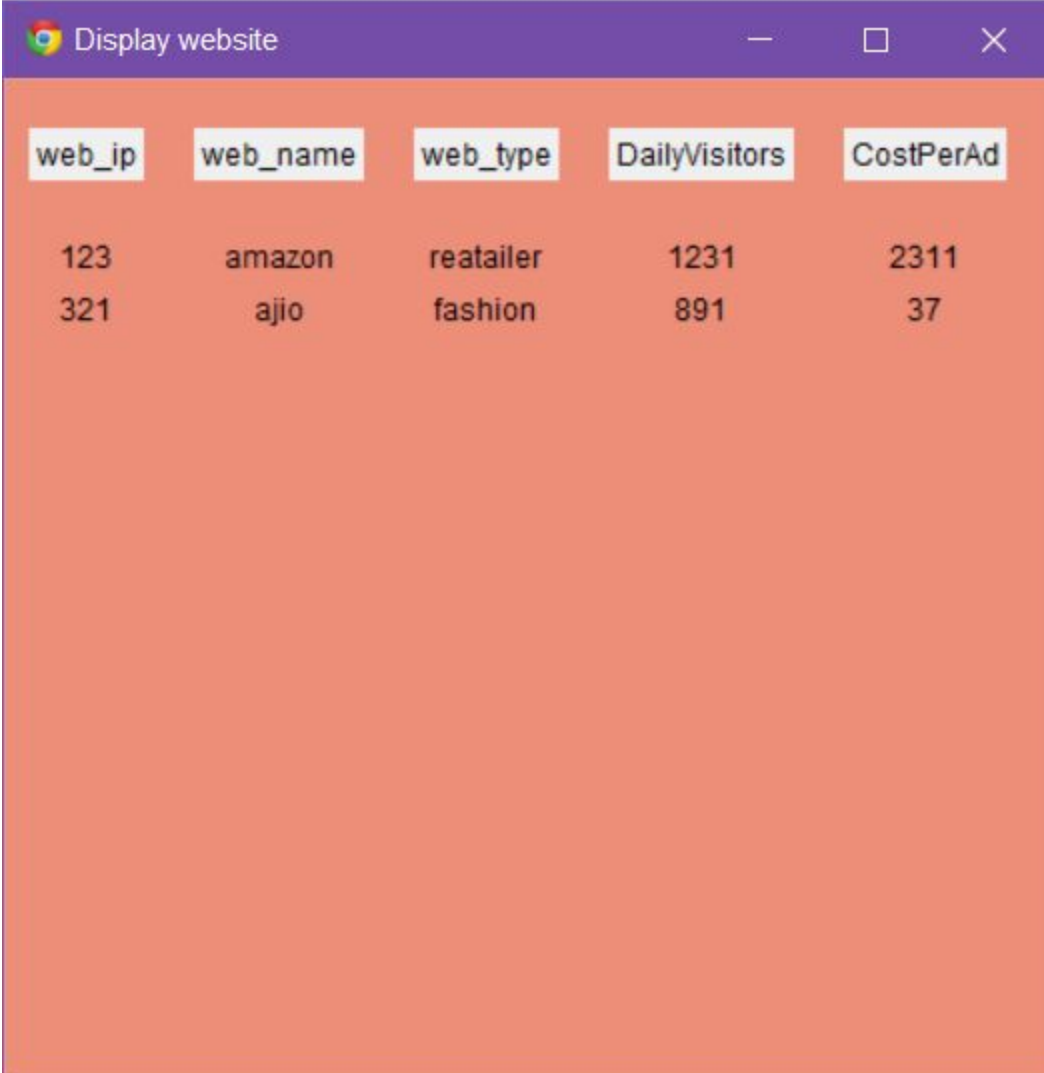
The screenshot shows a web browser window with the title 'insert to website'. The page has a dark blue background. At the top, there is a button labeled 'insert values into website'. Below this, there are five input fields, each with a label on the left and a text box on the right. The labels are 'web_ip', 'web_name', 'web_type', 'DailyVisitors', and 'CostPerAd'. The corresponding values entered in the text boxes are '321', 'ajio', 'fashion', '891', and '37'. At the bottom right of the form, there is a 'submit' button.

Field Name	Value
web_ip	321
web_name	ajio
web_type	fashion
DailyVisitors	891
CostPerAd	37

FIG(3): INSERTING TO WEBSITE TABLE



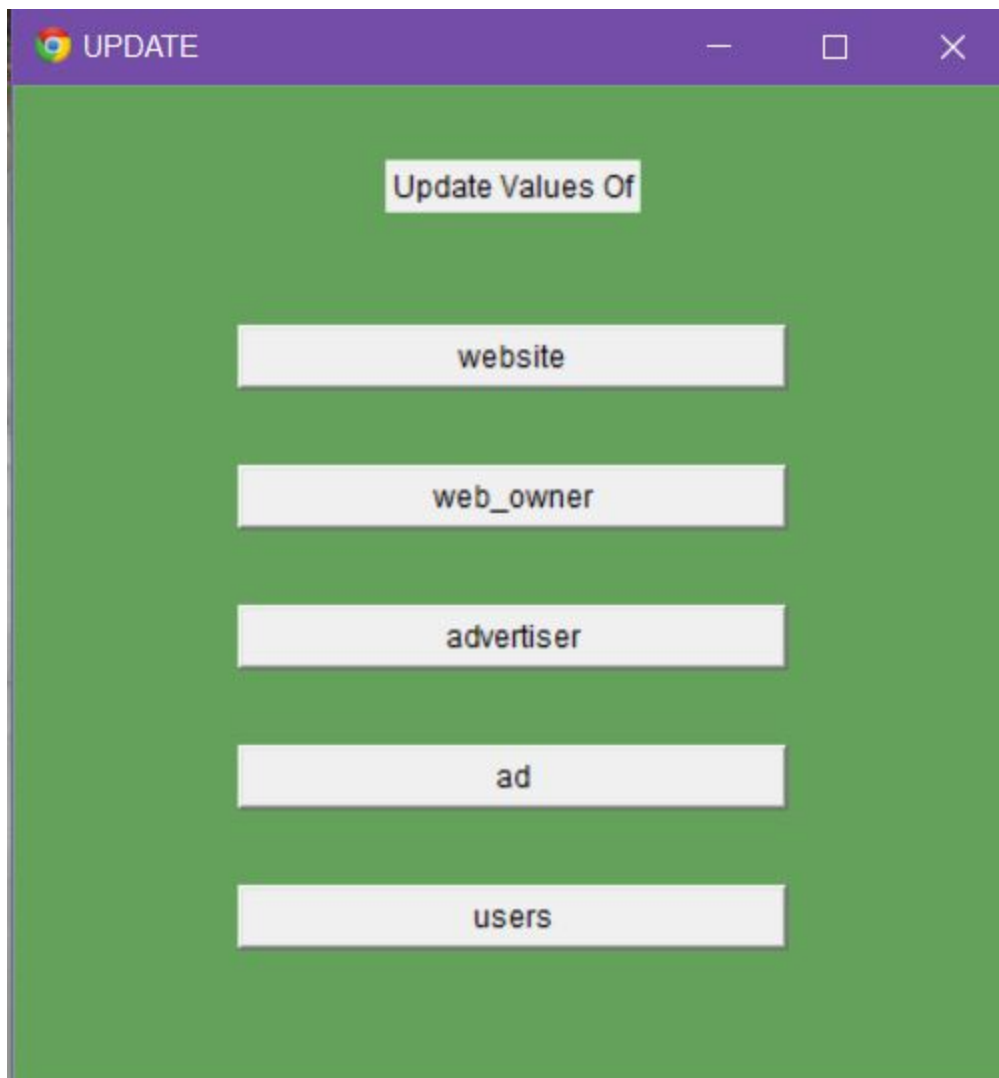
FIG(4): DISPLAY MENU PAGE



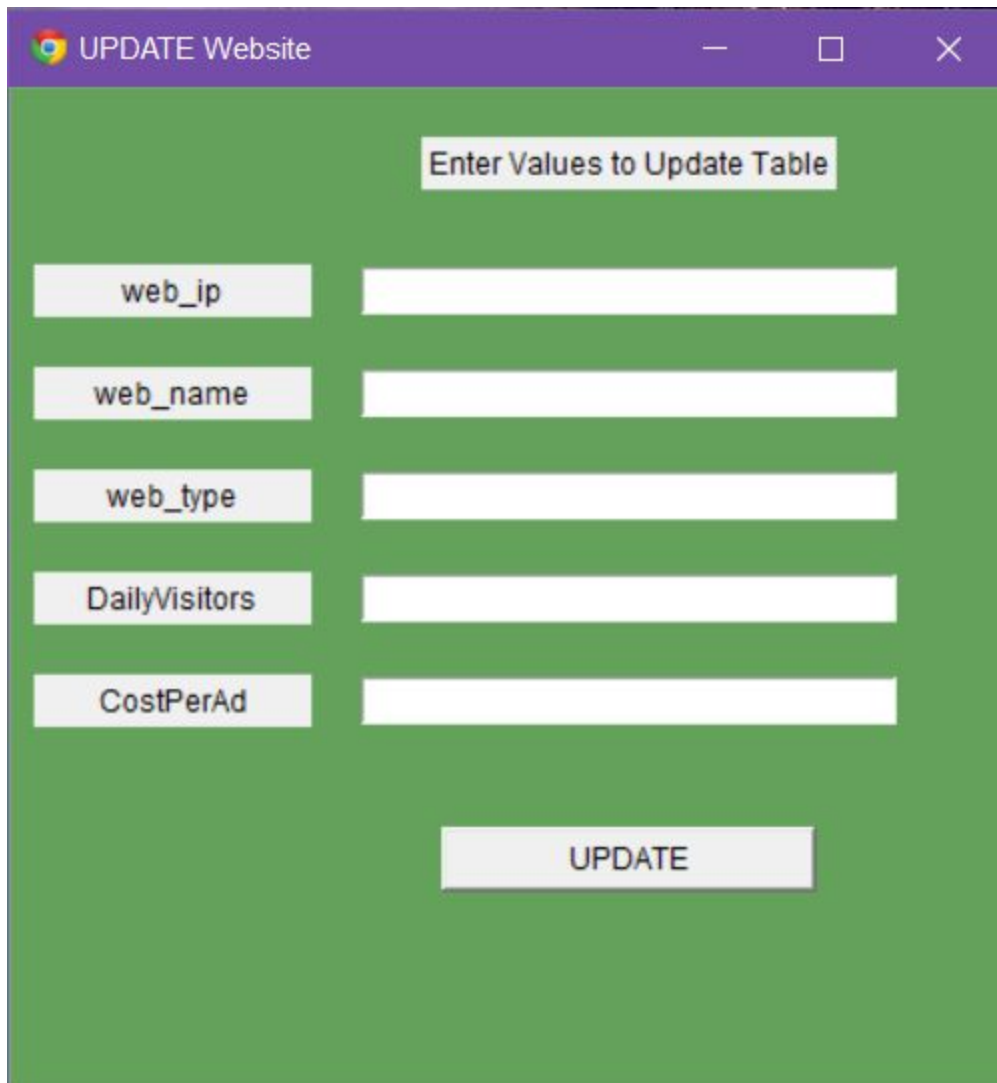
A screenshot of a web browser window titled "Display website". The window displays a table with five columns: "web_ip", "web_name", "web_type", "DailyVisitors", and "CostPerAd". The table contains two rows of data. The first row shows "123" for web_ip, "amazon" for web_name, "reatailer" for web_type, "1231" for DailyVisitors, and "2311" for CostPerAd. The second row shows "321" for web_ip, "ajio" for web_name, "fashion" for web_type, "891" for DailyVisitors, and "37" for CostPerAd.

web_ip	web_name	web_type	DailyVisitors	CostPerAd
123	amazon	reatailer	1231	2311
321	ajio	fashion	891	37

FIG(5): DISPLAYING CONTENTS OF WEBSITE



FIG(6): UPDATE MENU PAGE

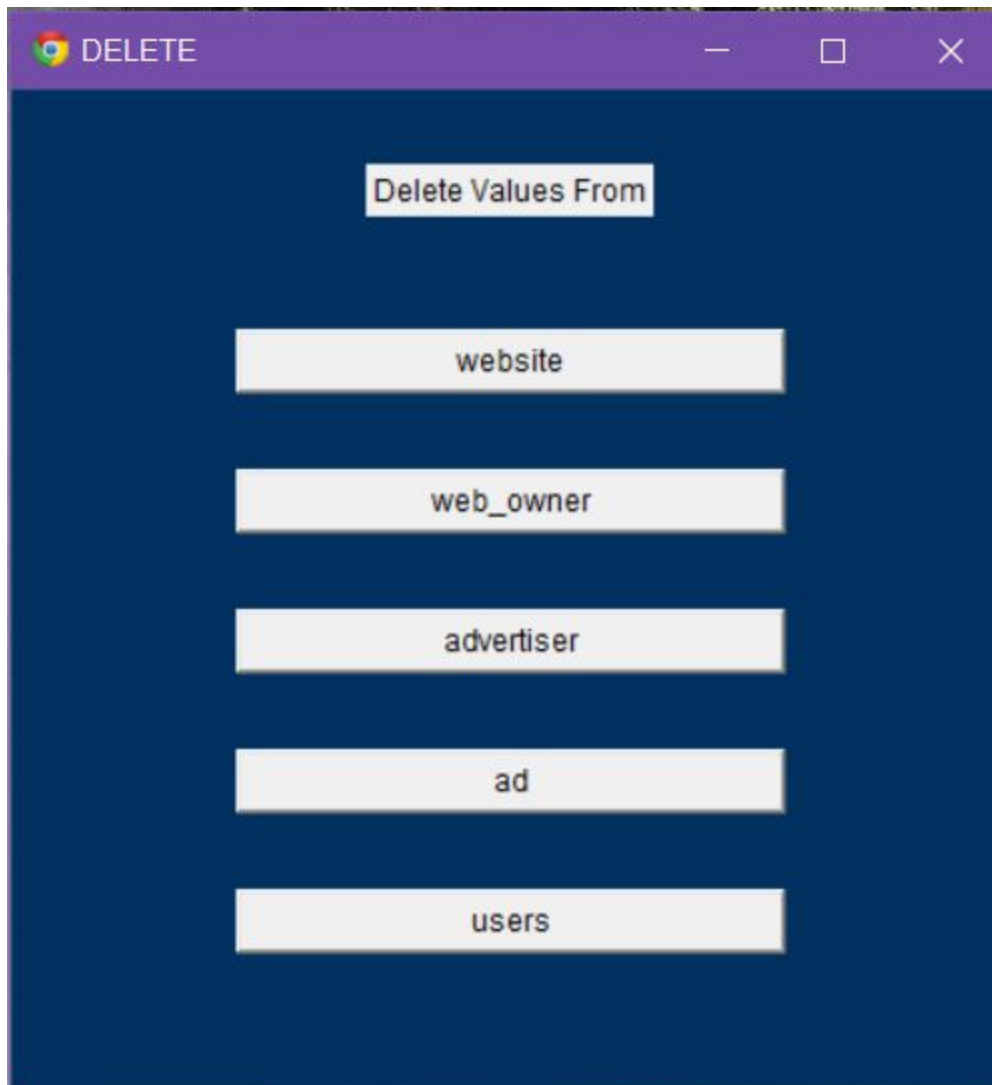


The screenshot shows a web browser window with a purple title bar that reads "UPDATE Website". The main content area has a green background. At the top, there is a white box with the text "Enter Values to Update Table". Below this, there are five rows of input fields. Each row consists of a light gray label box on the left and a white text input field on the right. The labels are "web_ip", "web_name", "web_type", "DailyVisitors", and "CostPerAd". At the bottom center, there is a light gray button with the text "UPDATE".

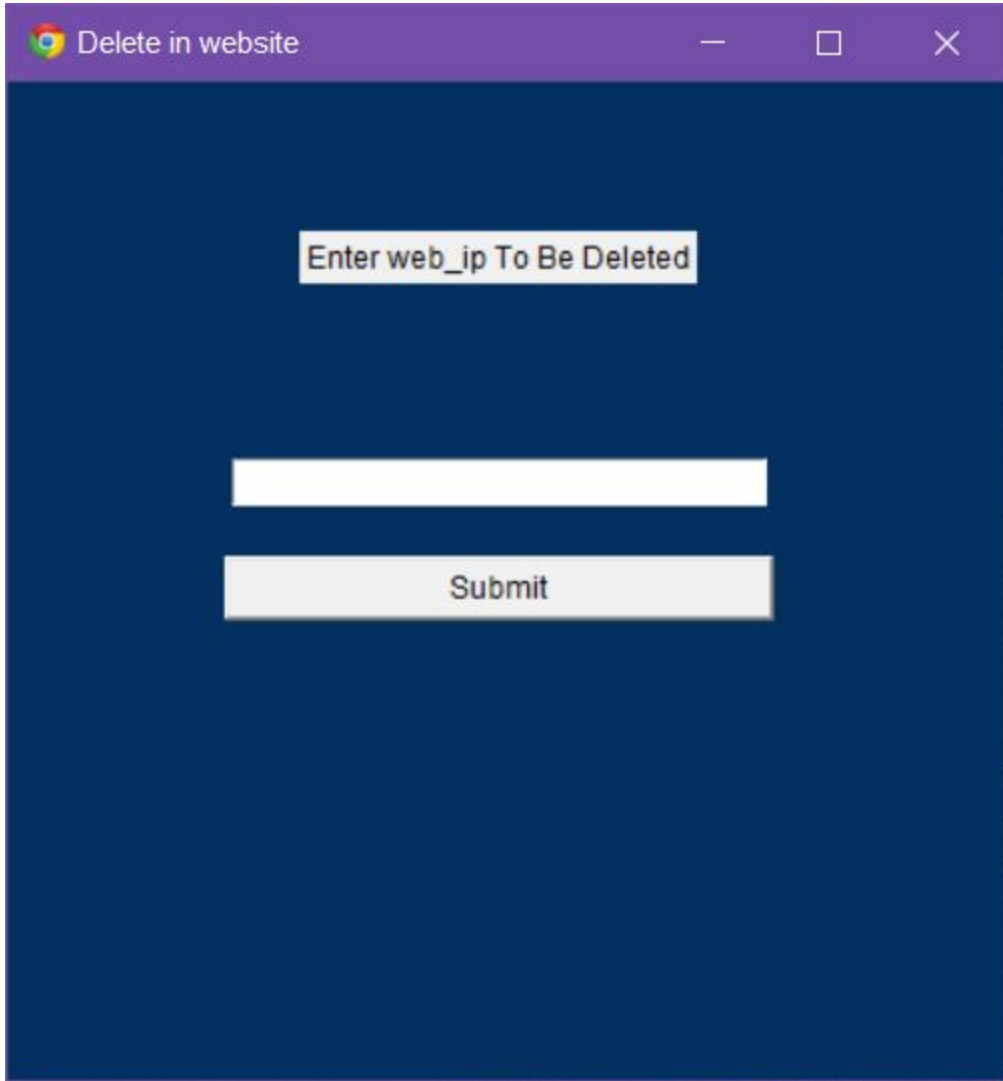
Enter Values to Update Table	
web_ip	<input type="text"/>
web_name	<input type="text"/>
web_type	<input type="text"/>
DailyVisitors	<input type="text"/>
CostPerAd	<input type="text"/>

UPDATE

FIG(7): UPDATE CONTENTS OF WEBSITE



FIG(8): DELETE MENU PAGE



The image shows a web browser window with a purple title bar that reads "Delete in website". The main content area has a dark blue background. In the center, there is a white rectangular box containing the text "Enter web_ip To Be Deleted". Below this box is a white input field. Underneath the input field is a light gray button with the text "Submit".

FIG(9): DELETE CONTENTS OF WEBSITE


```
SQL> select * from website;
```

WEB_IP	WEB_NAME	WEB_TYPE	DAILYVISITORS	COSTPERAD
123	amazon1	retailer	110	143
321	ajio	fashion	891	37

```
SQL> select * from web_owner;
```

OWNER_ID	OWNER_NAME	WEB_IP
781	amazon	123
190	reliance	321

```
SQL> select * from advertiser;
```

ADV_ID	ADV_TYPE	ADV_NAME
878	Retailer	nykaa
980	education	ant coaching

```
SQL> select * from ad;
```

AD_ID	AD_CATEGOR	AD_COUNT	ADV_ID	WEB_IP
888	fashion	1	878	123
980	education	2	980	321

```
SQL> select * from users;
```

USER_IP	USER_NAME	U	USER_R	WEB_IP
5435	guru	m	urban	123
6767	harish	m	rural	321

FIG(10): BACKEND TABLE CONTENTS

CHAPTER 5:

CONCLUSION

This database management system is used to improve overall efficiency and accuracy of information storing and provide effectiveness in maintaining data related to Ads advertised on websites. This can be used by various companies to advertise their different products or services on websites. This database system provides user-friendly environment to insert, update or delete the data and also it generates reports as per user requirements.

CHAPTER 6:

REFERENCES

1. www.w3schools.com (For GUI)
2. “Fundamentals of Database Systems” by Shamkant.b.navathe, 5th Edition, published by Dorling Kindersley.

