### 

**A CAPSTONE PROJECT REPORT**

*Submitted to*

**SAVEETHA SCHOOL OF ENGINEERING**

****

Submitted by P.Suhitha(192311152)

GUIDED BY Dr.Vasavi

Course Code: CSA0409

Course Name: Operating Systems

# **Capstone Project**

# **Marketing Plan to Help a Small Business Reach New Customers**

## 

## **1. Project Overview**

This project aims to develop a marketing plan integrated with a web application to help a small business expand its customer base. The application includes frontend interfaces for customer interactions, a backend system for data management, and a database for storing customer and marketing data.

## **2. Problem Statement**

Many small businesses struggle to reach new customers due to limited resources and expertise in digital marketing. This project addresses the challenge by offering an automated system to:

* Design and track marketing campaigns.
* Collect and analyze customer feedback.
* Improve engagement through targeted promotions.

## **3. Scenario**

A local coffee shop wants to increase its customer base and promote its new loyalty program. The proposed system will allow the shop to:

* Register customers into a loyalty program.
* Send personalized promotional offers.
* Analyze customer preferences based on past purchases.

## **4. Tasks**

1. Design a user-friendly frontend using AngularJS and JavaScript.
2. Set up a MySQL database to manage customer and campaign data.
3. Develop APIs to handle customer registration, promotional campaigns, and feedback collection.
4. Create comprehensive documentation covering system architecture and usage.

## **5. Deliverables**

1. **Frontend Code:** AngularJS-based responsive user interface.
2. **Backend Code:** Node.js APIs with RESTful endpoints.
3. **Database:** MySQL schema for customer and campaign data.
4. **Documentation:** Technical guide covering system implementation and usage.

**SOLUTION:**

#include <stdio.h>

#include <string.h>

#define MAX\_CUSTOMERS 100

#define NAME\_LENGTH 50

#define EMAIL\_LENGTH 100

// Define a structure to hold customer information

typedef struct {

char name[NAME\_LENGTH];

char email[EMAIL\_LENGTH];

int contacted; // 0 = Not contacted, 1 = Contacted

} Customer;

// Function to display customer list

void displayCustomers(Customer customers[], int totalCustomers) {

printf("\nCustomer List:\n");

printf("------------------------------------------------\n");

printf("Name\t\t\tEmail\t\t\t\tContacted\n");

printf("------------------------------------------------\n");

for (int i = 0; i < totalCustomers; i++) {

printf("%s\t\t%s\t\t\t%s\n",

customers[i].name,

customers[i].email,

customers[i].contacted ? "Yes" : "No");

}

printf("\n");

}

// Function to add a new customer

void addCustomer(Customer customers[], int \*totalCustomers) {

if (\*totalCustomers >= MAX\_CUSTOMERS) {

printf("Customer list is full.\n");

return;

}

Customer newCustomer;

printf("Enter customer name: ");

getchar(); // To clear the newline character

fgets(newCustomer.name, NAME\_LENGTH, stdin);

newCustomer.name[strcspn(newCustomer.name, "\n")] = 0; // Remove the trailing newline

printf("Enter customer email: ");

fgets(newCustomer.email, EMAIL\_LENGTH, stdin);

newCustomer.email[strcspn(newCustomer.email, "\n")] = 0; // Remove the trailing newline

newCustomer.contacted = 0; // Mark as not contacted initially

customers[\*totalCustomers] = newCustomer;

(\*totalCustomers)++;

printf("Customer added successfully!\n");

}

// Function to mark a customer as contacted

void contactCustomer(Customer customers[], int totalCustomers) {

char email[EMAIL\_LENGTH];

printf("Enter the email of the customer to mark as contacted: ");

getchar(); // To clear the newline character

fgets(email, EMAIL\_LENGTH, stdin);

email[strcspn(email, "\n")] = 0; // Remove the trailing newline

int found = 0;

for (int i = 0; i < totalCustomers; i++) {

if (strcmp(customers[i].email, email) == 0) {

customers[i].contacted = 1;

printf("Customer %s marked as contacted.\n", customers[i].name);

found = 1;

break;

}

}

if (!found) {

printf("Customer with email %s not found.\n", email);

}

}

// Main function to interact with the user

int main() {

Customer customers[MAX\_CUSTOMERS];

int totalCustomers = 0;

int choice;

while (1) {

printf("\nMarketing Plan Helper Tool\n");

printf("1. Add a new customer\n");

printf("2. Display customer list\n");

printf("3. Mark customer as contacted\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

addCustomer(customers, &totalCustomers);

break;

case 2:

displayCustomers(customers, totalCustomers);

break;

case 3:

contactCustomer(customers, totalCustomers);

break;

case 4:

printf("Exiting the program.\n");

return 0;

default:

printf("Invalid choice! Please try again.\n");

}

}

return 0;

}

## **b)API Documentation**

### **Endpoints**

#### **1. Customer Registration**

**URL:** /api/register  
 **Method:** POST  
 **Payload:**

{

"name": "John Doe",

"email": "john.doe@example.com",

"phone": "1234567890"

}

**Response:**

{

"message": "Customer registered successfully."

}

## **System Documentation**

### **1. Technologies and Tools**

* **Frontend:** AngularJS, JavaScript, HTML, CSS
* **Backend:** Node.js, Express.js
* **Database:** MySQL
* **Others:** Postman for API testing, Git for version control

### **2. System Architecture**

* **Frontend:** AngularJS handles user interactions and communicates with the backend.
* **Backend:** Node.js serves API endpoints.
* **Database:** MySQL stores customer and campaign data.

### **3. Installation Instructions**

1. Clone the repository: git clone <repo-url>
2. Install dependencies: npm install
3. Set up MySQL database using the provided schema.

### **4. Configuration**

* Configure database connection in config.js.

### **5. Implementation Details**

* APIs follow RESTful conventions.
* Error handling implemented for robust operation.

### **6. Running the Application**

1. Start the backend server: node server.js
2. Launch the frontend: Open index.html in a browser.

### **7. Usage Instructions**

* Register a customer via the registration form.
* Monitor campaigns through the admin dashboard.

### **8. Future Enhancements**

* Add AI-based recommendation for personalized promotions.
* Integrate with social media for wider outreach.

### **Conclusion**

The developed system provides a scalable and efficient way for small businesses to reach new customers. With additional features and ongoing updates, it can become a comprehensive solution for digital marketing challenges.