**NAME:** SHAHISHNU J R

**REG.NO:** 22BCE1046

**APPLICATION FOR HOME AUTOMATION**

**WORKING:**

In today's fast-paced world, managing both personal expenses and the comfort of our homes can be a demanding task. Imagine having a solution that seamlessly combines two essential aspects of daily life: expense tracking and smart home automation. Welcome to our innovative and user-friendly Android application designed to simplify your life.

Our multi-faceted application is more than just a mobile tool; it's a lifestyle enhancer. By integrating expense tracking, home automation, and document management into a single, intuitive platform, we empower you to take control of your finances, optimize your home environment, and effortlessly manage your important documents.

**Key Features:**

***Expense Tracking:***

Easily record your daily expenses, categorizing them for better financial insight.

Monitor your spending habits and stay within your budget.

Attach receipts and invoices as documents to expense entries for reference.

***Smart Home Automation:***

Seamlessly control your home appliances with just a few taps on your device.

Manage lights, fans, thermostats, and more to create the perfect ambiance at home.

Receive real-time feedback on the status of your appliances for enhanced convenience and energy efficiency.

***Document Management:***

Effortlessly store without organizing documents such as Aadhar card , driving license , personal documents including bills, contracts, and receipts etc

Retrieve and access your documents anytime, anywhere, directly from your smartphone with easy search option

Share documents securely with others when needed.

***How It Works:***

Record your expenses on the go, ensuring that you maintain a clear overview of your financial health.

Control your home environment by toggling appliances on or off, all from the convenience of your mobile device.

Manage your important documents digitally, eliminating clutter and streamlining your paperwork.

1. **Expense Tracking:**

Expense Entry: Users open the app and are initially presented with the expense tracking interface. They can enter their expenses by specifying the category (e.g., "Groceries") and the amount spent (e.g., "25.00 rupees").

Adding Expenses: When the user clicks the "Add Expense" button, the entered expense is added to the list of expenses displayed in the Text View. The expense is formatted as "Category: ₹Amount" and displayed on a new line.

Viewing Expenses: The Text View continuously updates to display the list of expenses. Users can view and manage their expenses within this section.

2. **Home Automation of Appliances:**

Appliance Control: The app includes a separate interface for controlling home appliances. Users can interact with buttons, switches, or sliders representing various appliances such as lights, fans, or thermostats.

Integration with IoT Devices: The app communicates with IoT (Internet of Things) devices that are connected to the home appliances. It sends commands to these devices based on user interactions with the appliance controls in the app.

Real-time Feedback: The app provides real-time feedback to users about the status of the appliances. For example, if the user taps a "Lights On" button, the app sends a command to turn on the lights through the IoT device and updates the UI to reflect that the lights are now on.

3**. Document Management:**

Document Storage: The app includes a section for managing documents. Users can upload, view, organize, and search for documents.

File Upload: Users can select and upload documents from their device. The app may include a file picker to facilitate the upload process.

Document Viewing: The app provides a document viewer that can display various document types (PDFs, images, text files, etc.) within the app. Users can open and view documents they have uploaded.

Creating a versatile Android application that seamlessly combines expense tracking, smart home automation, and document management is a complex endeavour using Android Studio. In Android Studio, you'd begin by designing user interfaces (UIs) through XML layouts, creating visually appealing screens for each feature. Then, you'd employ Java or Kotlin to code the functionality behind the UI elements. For expense tracking, you'd handle data storage, user input validation, and interactions. Smart home automation would involve establishing connections with IoT devices, designing a control interface, and implementing real-time feedback. Meanwhile, document management necessitates file upload and organization capabilities, document viewing, and robust data security. Integrating these components while maintaining a user-friendly experience and compliance with regulations is a multidimensional coding challenge, demanding meticulous planning, debugging, and user testing throughout the development process.

**FLOWCHART:**

**A diagram of a company

Description automatically generated**

**ALGORITHM:**

Here are some algorithms of the features in c:

1.**Expense tracking:**

This program will allow users to input expenses, categorize them, and view a summary

#include <stdio.h>

#include <string.h>

// Define constants

#define MAX\_EXPENSES 100

// Structure to represent an expense

struct Expense {

char category[50];

float amount;

};

// Function to add an expense

void addExpense(struct Expense expenses[], int \*count) {

if (\*count >= MAX\_EXPENSES) {

printf("Maximum number of expenses reached.\n");

return;

}

struct Expense newExpense;

printf("Enter category: ");

scanf("%s", newExpense.category);

printf("Enter amount: ");

scanf("%f", &newExpense.amount);

expenses[\*count] = newExpense;

(\*count)++;

printf("Expense added successfully.\n");

}

// Function to view expenses

void viewExpenses(struct Expense expenses[], int count) {

if (count == 0) {

printf("No expenses to display.\n");

return;

}

printf("Category\tAmount\n");

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

printf("%s\t%.2f\n", expenses[i].category, expenses[i].amount);

}

}

int main() {

struct Expense expenses[MAX\_EXPENSES];

int count = 0;

while (1) {

printf("\nEnter 'add' to add an expense, 'view' to view expenses, or 'exit' to quit: ");

char userInput[10];

scanf("%s", userInput);

if (strcmp(userInput, "add") == 0) {

addExpense(expenses, &count);

} else if (strcmp(userInput, "view") == 0) {

viewExpenses(expenses, count);

} else if (strcmp(userInput, "exit") == 0) {

break;

} else {

printf("Invalid input. Please enter 'add', 'view', or 'exit'.\n");

}

}

return 0;

}

2. **Automation of appliances:**

C algorithm for controlling switches (e.g., lights) in a smart home automation system.

#include <stdio.h>

#include <stdbool.h>

// Define global variables or constants

bool switchState = false; // Assuming the switch is initially off

// Function to turn on the switch

void turnOnSwitch() {

// Code to activate the switch, e.g., set GPIO pin high

// Replace this with the appropriate code for your hardware

// Example: gpio\_set\_high(SWITCH\_PIN);

switchState = true;

printf("Switch turned ON.\n");

}

// Function to turn off the switch

void turnOffSwitch() {

// Code to deactivate the switch, e.g., set GPIO pin low

// Replace this with the appropriate code for your hardware

// Example: gpio\_set\_low(SWITCH\_PIN);

switchState = false;

printf("Switch turned OFF.\n");

}

int main() {

while (1) {

// Prompt the user for input

printf("Enter 'on' to turn on the switch, 'off' to turn off, or 'exit' to quit: ");

char userInput[10];

scanf("%s", userInput);

// Process user input

if (strcmp(userInput, "on") == 0) {

if (!switchState) {

turnOnSwitch();

} else {

printf("Switch is already ON.\n");

}

} else if (strcmp(userInput, "off") == 0) {

if (switchState) {

turnOffSwitch();

} else {

printf("Switch is already OFF.\n");

}

} else if (strcmp(userInput, "exit") == 0) {

break;

} else {

printf("Invalid input. Please enter 'on', 'off', or 'exit'.\n");

}

}

return 0;

}

3**.Document management system:**

This algorithm assumes you want to save and retrieve documents from a local directory

#include <stdio.h>

#include <stdlib.h>

// Define global variables or constants

#define DOCUMENT\_STORAGE\_PATH "/path/to/document/storage/"

// Function to upload a document

void uploadDocument(char\* documentName, char\* content) {

char fullPath[256];

snprintf(fullPath, sizeof(fullPath), "%s%s", DOCUMENT\_STORAGE\_PATH, documentName);

FILE\* file = fopen(fullPath, "w");

if (file == NULL) {

printf("Error uploading document\n");

return;

}

fprintf(file, "%s", content);

fclose(file);

printf("Document '%s' uploaded successfully.\n", documentName);

}

// Function to retrieve a document

void retrieveDocument(char\* documentName) {

char fullPath[256];

snprintf(fullPath, sizeof(fullPath), "%s%s", DOCUMENT\_STORAGE\_PATH, documentName);

FILE\* file = fopen(fullPath, "r");

if (file == NULL) {

printf("Error retrieving document\n");

return;

}

char content[1024]; // Adjust buffer size as needed

if (fgets(content, sizeof(content), file) != NULL) {

printf("Retrieved document '%s':\n%s\n", documentName, content);

}

fclose(file);

}

int main() {

while (1) {

// Prompt the user for input

printf("Enter 'upload' to upload a document, 'retrieve' to retrieve a document, or 'exit' to quit: ");

char userInput[10];

scanf("%s", userInput);

// Process user input

if (strcmp(userInput, "upload") == 0) {

char documentName[256];

printf("Enter the document name: ");

scanf("%s", documentName);

char documentContent[1024]; // Adjust buffer size as needed

printf("Enter the document content: ");

scanf(" %[^\n]s", documentContent);

uploadDocument(documentName, documentContent);

} else if (strcmp(userInput, "retrieve") == 0) {

char documentName[256];

printf("Enter the document name to retrieve: ");

scanf("%s", documentName);

retrieveDocument(documentName);

} else if (strcmp(userInput, "exit") == 0) {

break;

} else {

printf("Invalid input. Please enter 'upload', 'retrieve', or 'exit'.\n");

}

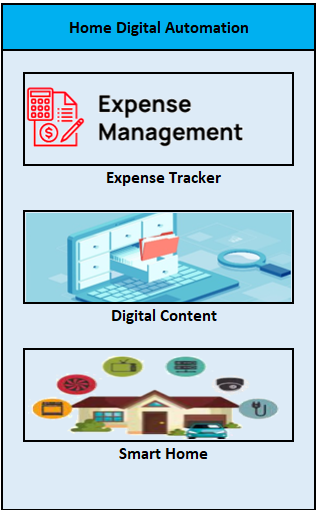
}

return 0;

}

**IMPLEMENTATION:**

**Home Screen:**

****

**Description:**

1. User will login the Mobile Application.
2. If the user is valid it will show the Home Page.

**Home Page:**

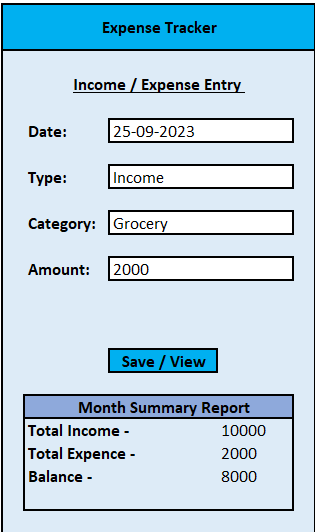
It will have 3 Menus in the Home Page

1. Expense Management.

2. Document management.

3. Smart House.

**Expense Tracker:**

****

**Description:**

User will enter the income and expense in this screen. User can download the monthly report.

1.Enter the Date of Expense or Income.

2. Enter the Income (or) Expense.

3. Enter the Income or Expense Category.

4. Enter the Income or Expense amount.

Click on Save button to save the Data.

Monthly Summary Report will be generated.

**Document Management System:**

**A blue file with black text

Description automatically generated**

**Description:**

User can upload the documents in this screen.

User can retrieve the saved document anytime.

1.Attach the attachment of the document up to 3 Pages

2. Select Document Type (Like: Aadhar, Pan….)

3.Click on "Upload" Button to Store the data

4.Click on "Retrieve" Button to Retrieve the Saved data.

**Appliances Automation:**

**A screenshot of a phone

Description automatically generated**

**Description:**

User can access all the appliances in the home with a single click on the mobile.

User can switch on or switch off the home appliance using this screen.

Click on the Device which you want to switch on and switch off.