**BASAVARAJESWARI GROUP OF INSTITUTIONS**

**BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)**

**"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## A Mini Project Report on

“**FILE MANAGEMENT APPLICATION”**

**Submitted in partial fulfillment of the requirement for**

“Mobile Application Development Laboratory with mini project”

#### For the award of Degree

Bachelor of Engineering in Computer Science & Engineering

#### Submitted by

**POLEPALLI VISHNU SAI USN: 3BR20CS126**

**PRIYANKA MADINUR USN: 3BR20CS128 R KAVYA USN: 3BR20CS129**

#### Under the guidance of

**Ms. MADHURI A Mrs. S STEFFI NIVEDITA**

Asst. Professor. Asst. Professor.

**Dept of CS&E, BITM, Ballari.**

[**Visvesvaraya Technological University**](http://www.vtu.ac.in/)

## Belagavi, Karnataka 2022-2023

**BASAVARAJESWARI GROUP OF INSTITUTIONS**

**BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT**

NACC Accredited Institution\*

**(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belgavi)**

**"Jnana Gangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)**

**Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

CERTIFICATE

This is to certify that the MAD Lab Mini Project entitled “SMS APPLICATION SYSTEM” has been successfully carried out by **POLEPALLI VISHNU SAI** bearing **(3BR20CS126)**, **PRIYANKA MADINUR** bearing **(3BR20CS128)** and **R KAVYA**

bearing **(3BR20CS129),** students of VI semester B.E. for the partial fulfillment of the requirements for the award of **Bachelor Degree in Computer Science & Engineering** of the VISVESVARAYA TECHNOLOGICAL UNIVERSITY during the academic year 2022-2023

Signature of Guide(s) Signature of HOD

#### Ms. Madhuri A

**Mrs. S Steffi Nivedita Dr. R.N. Kulkarni**

**EXTERNAL EXAMINATION**

## Name of the Examiners Signature with Date

**1. ……………………... …………………….**

## 2. ……………………… ..…………………….

**Acknowledgement**

In the accomplishment of completion of the project on **“FILE MANAGEMENT APPLICATION”**, I would like to convey my special gratitude to **Ms. Madhuri A** and **Mrs. S Steffi Nivedita** Project Guides and as well as **Dr. R.N. Kulkarni**, HOD of CSE Dept., BITM, Bellary.

I would also like to express my gratitude towards our principal, **Dr. Yadavalli Basavaraj**, for giving me this great opportunity to do a project. Without their support and suggestions, this project would not have been completed.

Your valuable guidance and suggestions helped me in various phases of the completion of this project. I will always be thankful to you in this regard.

I would like to extend my deep appreciation to all my group members, without their support and coordination we would not have been able to complete this project.

Name USN

POLEPALLI VISHNU SAI 3BR20CS126 PRIYANKAMADINUR 3BR20CS128

R KAVYA 3BR20CS129

**Abstract**

The File Management application is a simple yet effective tool for Android users who want to manage files in their system without much hassle. The application is easy to use, and its interface is designed to be simple and straightforward, with a three button that enables users to create, open save accordingly with ease. The file management application is a versatile software tool that empowers users to create, open, and save files efficiently. It provides a user-friendly interface for creating new documents based on their needs. The application supports seamless file opening, enabling users to access and edit existing files effortlessly. It also offers a robust saving mechanism, ensuring that modifications are preserved accurately. With features such as save and version control, the file application guarantees data integrity and prevents data loss. This application simplifies the file management process, enhancing productivity and enabling users to focus on their tasks without worrying about file organization or data security.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Contents** | **Page No.** |
|  | Acknowledgement | I |
|  | Abstract | II |
|  | Table of Contents | III |
| **1** | **INSTALLATION AND SETUP PROCEDURE** | 1-2 |
| **2** | **INTRODUCTION** | 3 |
| 2.1 | OVERVIEW OF THE PROJECT | 4 |
| 2.2 | AIM OF THE PROJECT | 5 |
| **3** | **REQUIREMENTS SPECIFICATION** | 6 |
| 3.1 | FUNCTIONAL REQUIREMENTS | 6 |
| 3.2 | NON-FUNCTIONAL REQUIREMENTS | 7 |
| 3.3 | HARDWARE & SOFTWARE REQUIREMENTS | 7-8 |
| **4** | **DETAILED DESIGN** | 9-10 |
| **5** | **IMPLEMENTATION** | 11-20 |
| 5.1 | MODULE AND THEIR ROLES | 11 |
| **6** | **TESTING** | 21 |
| **7** | **SNAP SHOTS** | 23 |
| **8** | **CONCLUSION** | 24 |
|  | **REFERENCES** | 25 |

**Chapter 1**

# INSTALLATION AND SETUP PROCEDURE

To install and set up the Android Studio development environment, follow these steps:

* **Download Android Studio**: Go to the official Android Studio website (https://developer.android.com/studio) and download the latest version of Android Studio for your operating system (Windows, macOS, or Linux).
* **Install Android Studio**: Run the downloaded installer and follow the on- screen instructions toinstall Android Studio on your computer.
* **Set up Android Virtual Device (AVD)**: Launch Android Studio and open the AVD Manager.Create a new virtual device or use an existing one to emulate an Android device on your computer. This step is optional but recommended for testing and running the application on anemulator.
* **Configure SDK and Dependencies**: Android Studio will guide you through the process of installing the necessary Android SDK components and dependencies. Make sure to install the required SDK versions and any additional libraries or tools as needed.
* **Clone or Import the Project**: Clone the provided project code or import it into Android Studioby selecting "Open an existing Android Studio project" from the welcome screen or "File > Open" from the menu. Select the project directory and let Android Studio load the project.
* **Build the Project**: Once the project is loaded, Android Studio will automatically sync and buildthe project. It may take some time to download any missing dependencies and build the projectfor the first time.
* **Running the Application:** To run the application on an Android device or emulator, follow these steps:
* **Connect a Physical Device (Optional):** If you have a physical Android device, connect it to your computer using a USB cable. Make sure USB debugging is enabled on the device.
* **Select the Target Device**: In Android Studio, click on the target device dropdown menu in thetoolbar. Choose the connected physical device or select the desired AVD emulator from the list.
* **Build and Run the Application**: Click on the "Run" button in the toolbar or select "Run > Run'app'" from the menu. Android Studio will compile the code, build the APK, and deploy the application to the selected device or emulator.
* **Test the Application**: Once the application is installed and launched on the device or emulator, you can interact with it to send and receive SMS messages. Use the provided UI elements to enter a phone number, message, and send SMS. The received SMS messages will be displayedon the screen.

**Chapter 2**

# INTRODUCTION

The file creation and management application is a software tool designed to facilitate the creation, organization, and manipulation of digital files. It provides users with a convenient and efficient way to create new files, store them in a structured manner, and perform various operations on them.

The application typically offers a user-friendly interface where users can create different types of files, such as documents, spreadsheets, presentations, images, and more. It provides essential editing features, allowing users to modify the content of their files, format text, insert media, and apply various styling options.

File management functionalities are an integral part of the application, allowing users to perform actions like renaming files, copying or moving them to different locations, and deleting unnecessary files. The application may also include features like version control, where users can track changes made to files and revert to previous versions if needed.

Overall, the file creation and management application simplifies the process of creating, organizing, and manipulating digital files, offering users a streamlined and efficient workflow for their file-related tasks.

## OVERVIEW OF THE PROJECT

The File Management System project aims to develop a user-friendly and feature-rich text file management app for Android devices. The app provides a convenient platform for users to create, save, andorganize their text-based information.

With File Management System, users can easily create new text files and give them custom names. They can then save the contents of these files, associating them with their respective names. Saved files can be opened for viewing and editing, allowing users to update their content as needed.

One of the key features of File Management System is its file management functionality. The app provides a list of all saved files, allowing users to easily access and manage their files in one place.

The user interface of File Management System is designed to be intuitive and user- friendly. The app offers a visually appealing and responsive layout, making it easy for users to navigate and interact with its various features. The interface provides clear instructions and prompts, guiding users through the process of creating, saving, and managing files.

The File Management System project leverages the power of modern Android development frameworks and tools to ensure optimal performance and stability. It utilizes efficient data storage mechanisms to securely store file contents and associated metadata.

Overall, the File Management System app provides users with a seamless and efficient solution for managing their text-based information. By offering essential features and a user-friendly interface,File Management System aims to enhance productivity, organization, and accessibility for users dealing with textual data on their Android devices.

## AIM OF THE PROJECT

To design and develop the File Management application system that allow users to upload

, store and manage their files in a user-friendly way

**Chapter 3**

# REQUIREMENTS SPECIFICATION

## FUNCTIONAL REQUIREMENT

* + 1. **File Creation**: The app should allow users to create new text files by providing a file name.
    2. **File Saving**: Users should be able to save the contents of a text file.
    3. **File Opening:** Users should be able to open a saved text file for viewing and editing.
    4. **File Listing:** The app should provide a list of all saved text files for easy access and management.
    5. **User Input Validation:** The app should validate user input for file creation and management operations, ensuring valid file names and preventing errors.
    6. **User Feedback:** The app should provide real-time feedback to users, such as successmessages, error notifications, and informative prompts.

## NON-FUNCTIONAL REQUIREMENTS

* + 1. **Usability:** The app should have a user-friendly interface, intuitive navigation, and clearinstructions to ensure ease of use.
    2. **Performanc**e: The app should be responsive and perform tasks efficiently, with minimallatency in file operations.
    3. **Security:** File contents should be stored securely, ensuring data privacy and protection.
    4. **Compatibility:** The app should be compatible with a wide range of Android devices andoperating system versions.
    5. **Reliability:** The app should function reliably, without crashes or data loss, ensuring theintegrity of saved files.
    6. **Accessibility:** The app should support accessibility features to cater to users with differentabilities.
    7. **Scalability:** The app should be designed to handle a large number of files withoutcompromising performance.

## HARDWARE & SOFTWARE REQUIREMENTS

**Hardware Requirements:**

Computer: A desktop or laptop computer with a minimum of 4GB RAM (8GBor higher recommended) and a multi-core processor.

Storage: Sufficient free disk space for Android Studio installation, project files, and dependencies.

Internet Connection: A stable internet connection for downloading software updates, libraries, and APIs.

## Software Requirements:

Operating System: Windows 10, macOS, or Linux (Ubuntu 18.04 LTS or higher recommended).

Java Development Kit (JDK): Android Studio requires JDK 8 or JDK 11 to be installed. The recommended version is JDK 11.

Android Studio: The latest stable version of Android Studio, which can be downloaded from the official Android developer website (https://developer.android.com/studio).

Android SDK: Android Studio includes the Android SDK, but it is essential to ensure that the necessary SDK components, including the Android API levels required for the project, are installed.

Gradle: The project build system used by Android Studio. The version of Gradle is typically bundled with Android Studio and should be up to date.

Virtual Device: To test the application during development, an Android Virtual Device (AVD) is required. AVD Manager in Android Studio allows the creation and management of virtual devices with different configurations.

**Chapter 4**

# DETAILED DESIGN ( Methodology)

### User Interface Design:

* Design the main activity layout using Android XML layouts.
* Include text boxes, buttons, and other necessary UI elements for file operations.
* Define an XML file `activity\_main.xml` to specify the layout.

### Activity Setup:

* Create the `MainActivity` class that extends `AppCompatActivity`.
* Set up event listeners for buttons and text boxes.
* Handle button clicks and user input using `OnClickListener` and `getText` methods.

### File Creation:

* Implement the logic to create a new text file.
* Retrieve the file name entered by the user.
* Create a corresponding file on the device's storage using `File` and `FileWriter`.
* Display toast messages to inform the user about the file creation status.

### File Saving:

* Implement the logic to save the contents of the text box to the created file.
* Retrieve the text entered by the user.
* Write the text to the file using `FileWriter` and `flush`.
* Display toast messages to inform the user about the file saving status.

### File Opening:

* Implement the logic to open a saved file.
* Retrieve the selected file name from the file list.
* Read the contents of the file using `getFileContent` and display them in the text box.
* Display toast messages to inform the user about the file opening status.

### File Listing:

* Implement the logic to retrieve and display a list of saved files.
* Access the files in the app's storage directory using `File`.
* Retrieve the file names and populate a list view or recycler view using `ArrayAdapter`.
* Display the list of file names to the user.

### Error Handling:

* Implement appropriate error handling mechanisms.
* Display toast messages or dialog boxes to inform the user about errors or issues during file operations.
* Handle cases such as empty file names, missing file content, and file not found.

### Testing and Refinement:

* Thoroughly test the app to ensure all functionalities work as intended.
* Identify and fix any bugs or issues discovered during testing.
* Validate the app's behavior under different scenarios and user interactions.

### Application Deployment:

* Package the app for distribution.
* Publish it on the Google Play Store or generate an APK file for installation on Android devices.
* Follow the necessary steps for app signing and deployment to ensure a smooth distribution process.

**Chapter 5**

# IMPLEMENTATION

package com.example.memorie; import android.content.Context;

import android.content.SharedPreferences; import android.os.Bundle;

import android.text.TextUtils; import android.view.View;

import android.widget.ArrayAdapter; import android.widget.Button; import android.widget.EditText; import android.widget.ListView; import android.widget.TextView; import android.widget.Toast;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import java.io.File;

import java.io.FileOutputStream; import java.io.FileWriter;

import java.io.IOException; import java.util.HashSet; import java.util.List;

import java.util.Set;

public class MainActivity extends AppCompatActivity {

private static final String FILE\_NAMES\_PREF = "FileNamesPref"; private static final String FILE\_CONTENT\_PREF = "FileContentPref"; private EditText editTextFileName;

private EditText editTextFileContent; private Button buttonCreate;

private Button buttonSave; private Button buttonOpen; private TextView textView;

private SharedPreferences fileNamesPref; private SharedPreferences fileContentPref; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

editTextFileName = findViewById(R.id.editTextFileName); editTextFileContent = findViewById(R.id.editTextFileContent); buttonCreate = findViewById(R.id.buttonCreate);

buttonSave = findViewById(R.id.buttonSave); buttonOpen = findViewById(R.id.buttonOpen); textView = findViewById(R.id.textView);

fileNamesPref = getSharedPreferences(FILE\_NAMES\_PREF, Context.MODE\_PRIVATE);

fileContentPref = getSharedPreferences(FILE\_CONTENT\_PREF, Context.MODE\_PRIVATE);

Button buttonShowFiles = findViewById(R.id.buttonShowFiles); buttonShowFiles.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) { showFileList();

}

});

buttonCreate.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

String fileName = editTextFileName.getText().toString().trim(); if (fileName.isEmpty()) {

Toast.makeText(MainActivity.this, "Please enter a file name", Toast.LENGTH\_SHORT).show();

} else if (!fileExists(fileName)) { createFile(fileName); saveFileNames(fileName); saveFileContent(fileName, "");

Toast.makeText(MainActivity.this, "File created successfully", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(MainActivity.this, "File already exists", Toast.LENGTH\_SHORT).show();

}

}

});

buttonSave.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View v) {

String fileName = editTextFileName.getText().toString().trim(); String fileContent = editTextFileContent.getText().toString();

if (fileName.isEmpty()) {

Toast.makeText(MainActivity.this, "Please enter a file name", Toast.LENGTH\_SHORT).show();

} else if (fileExists(fileName)) { saveFileContent(fileName, fileContent);

Toast.makeText(MainActivity.this, "File saved successfully", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(MainActivity.this, "First create a file", Toast.LENGTH\_SHORT).show();

}

}

});

buttonOpen.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String fileName = editTextFileName.getText().toString().trim(); if (fileName.isEmpty()) {

Toast.makeText(MainActivity.this, "Please enter a file name", Toast.LENGTH\_SHORT).show();

} else if (fileExists(fileName)) {

String fileContent = getFileContent(fileName); editTextFileContent.setText(fileContent);

Toast.makeText(MainActivity.this, "File opened successfully", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(MainActivity.this, "File not found", Toast.LENGTH\_SHORT).show();

}

}

});

}

private boolean fileExists(String fileName) { Set<String> fileNames = getFileNames(); return fileNames.contains(fileName);

}

private void createFile(String fileName) { if (TextUtils.isEmpty(fileName)) {

Toast.makeText(this, "Please enter a file name", Toast.LENGTH\_SHORT).show();

return;

}

EditText editText = findViewById(R.id.editTextFileName); String fileContent = editText.getText().toString();

if (TextUtils.isEmpty(fileContent)) { Toast.makeText(this, "Please enter file content",

Toast.LENGTH\_SHORT).show(); return;

}

File file = new File(getExternalFilesDir(null), fileName);

try {

FileWriter writer = new FileWriter(file); writer.append(fileContent);

writer.flush(); writer.close();

Toast.makeText(this, "File created: " + fileName, Toast.LENGTH\_SHORT).show();

} catch (IOException e) { e.printStackTrace();

Toast.makeText(this, "Error creating file", Toast.LENGTH\_SHORT).show();

}

}

private void saveFileNames(String fileName) { Set<String> fileNames = getFileNames(); fileNames.add(fileName);

SharedPreferences.Editor editor = fileNamesPref.edit(); editor.putStringSet("fileNames", fileNames); editor.apply();

}

private Set<String> getFileNames() {

return fileNamesPref.getStringSet("fileNames", new HashSet<String>());

}

private void saveFileContent(String fileName, String fileContent) { SharedPreferences.Editor editor = fileContentPref.edit(); editor.putString(fileName, fileContent);

editor.apply();

}

private String getFileContent(String fileName) { return fileContentPref.getString(fileName, "");

}

private void showFileList() {

Set<String> fileNames = getFileNames(); StringBuilder fileListBuilder = new StringBuilder(); for (String fileName : fileNames) {

fileListBuilder.append(fileName).append("\n");

}

String fileList = fileListBuilder.toString().trim();

// Display the file list using a dialog or any other appropriate UI element

// For example, you can use an AlertDialog to show the list of files AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this); builder.setTitle("Saved Files");

builder.setMessage(fileList); builder.setPositiveButton("OK", null); builder.show();

}

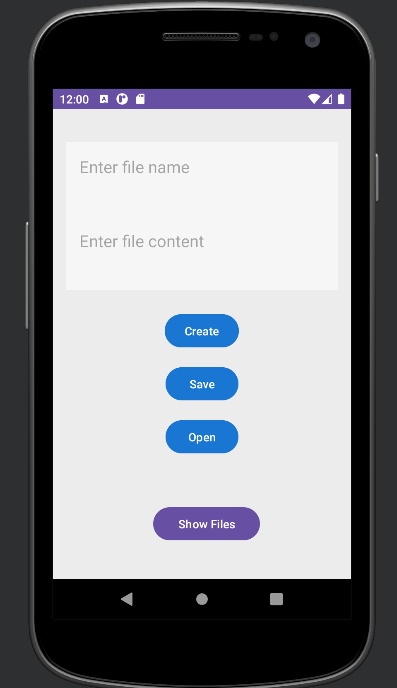
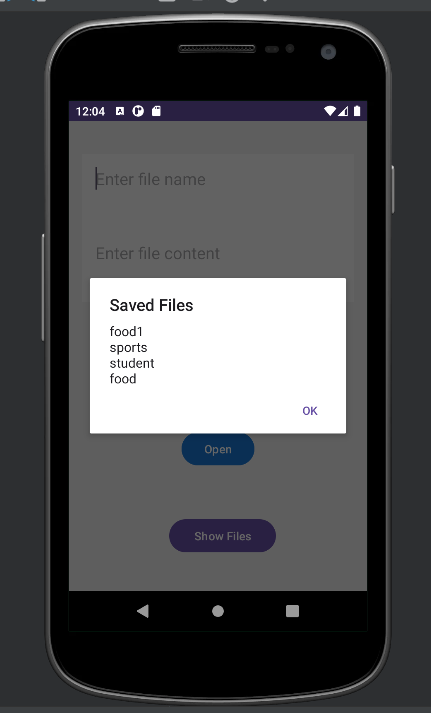
}

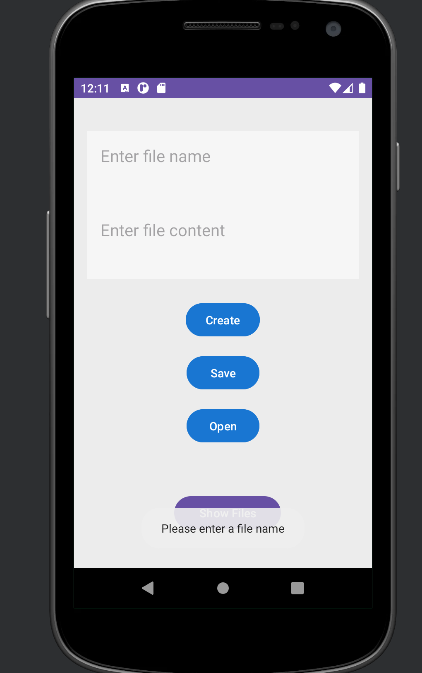
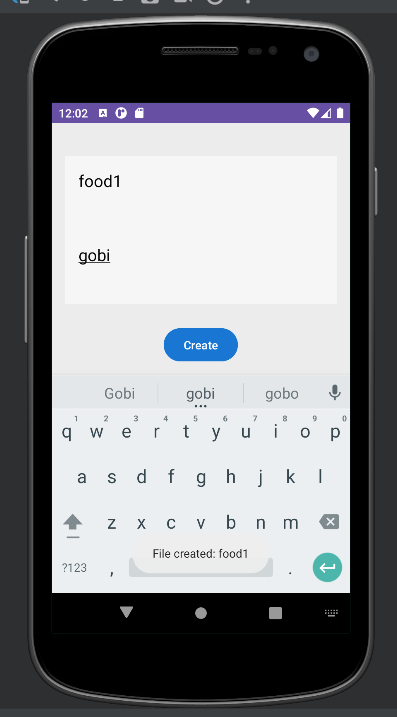
**Chapter 6**

# TESTING

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Function With Parameters** | **Under Test** | **Expected Result** | **Actual Result** | **Comments** |
| 1 | Create File | File Name: "hello1.txt" | New file "hello1.txt" is created | New file "hello1.txt" is created | The file is successfully created with the provided name. |
| 2 | Save Content | File Name: "hello1.txt" | Content is saved to the file | Content is saved to the file | The content entered by the user is successfully saved to the specified  file. |
| 3 | Open File | File Name: "hello1.txt" | File "hello1.txt" isopened | File "hello1.txt" is opened | The selected file is successfully opened and its content is displayed  in the text box. |
| 4 | List Files | - | List of saved files is displayed | List of saved files is displayed | The app successfully retrieves the list of saved files and displays them in the file list  view. |
| 5 | Invalid File Name | File Name: " " | Display error message "Invalid file name" | Display error message "Invalid file name" | The app properly handles the scenario when the user enters an empty file name and displays an appropriate  error message. |
| 6 | Non-existent File | File Name: "nonexistent.txt" | Display error message "File not found" | Display error message "File not found" | The app handles the case when the user tries to open a file that does not exist and informs the user with an error  message. |
| 7 | Empty Content | File Name: "hello1.txt" | Display error message "Empty  content" | Display error message "Empty  content" | The app correctly detects when the user tries to save an empty  content to a file. |

**SNAP SHOTS**

**Chapter 8**

# CONCLUSION

File application has been successfully completed, resulting in the creation of a functional and user-friendly mobile application for file management.Analysis of the requirements and designed the mobile application accordingly. Successfully implemented the designed features and functionalities of the mobile application.Special attention was given to the user interface and experience of the mobile application The application's features were accessible, and actions were intuitive, providing a satisfying user experience.The mobile application offers comprehensive file management functionality, allowing users to create, edit, organize, and search for files easily.Performance optimization techniques were employed to ensure the mobile application's smooth operation.Overall, the completion of the mobile application development lab project for the file application has resulted in a robust and user-friendly solution .

# REFERENCES

[1]. Sukhpal Singh Gill, Inderveer Chana, Rajkumar Buyya “*IoT Based Agriculture as a Cloud and Big Data Service: The Beginning of Digital India***”,** Journal of Organizational and End User Computing Volume 29 • Issue 4 • October- December 2017, DOI: 10.4018/JOEUC.2017100101.

[2]. *Agriculture Data, Government of India. (n.d.). Retrieved* from [https://data.gov.in](https://data.gov.in/catalogs/sector/Agriculture-9212)

[/catalogs/sector/Agriculture-9212.](https://data.gov.in/catalogs/sector/Agriculture-9212)

[3]. *India Agriculture And Climate DataSet.(n.d.). Retrieved* from [https://ipl.econ.du](https://ipl.econ.duke.edu/dthomas/dev_data/datafiles/india_agric_climate.htm) [ke.edu/dthomas/dev\_data/datafiles/india\_agric\_climate.htm](https://ipl.econ.duke.edu/dthomas/dev_data/datafiles/india_agric_climate.htm)

[4]. Deepanwita Gita Niyogi “*India’s millets policy: is it headed in the right direction?*” Mongabay: News & Inspiration from nature’s frontline in India. Mongabay Series: Conserving Agro-biodiversity.

[5]. B.Manjula Josephine, K.Ruth Ramya, K.V.S.N Rama Rao, Swarna Kuchibhotla,

P. Venkata Bala Kishore, S. Rahamathulla “*Crop Yield Prediction Using Machine Learning*” International journal of scientific & technology research volume 9, issue 02, february 2020, issn 2277-8616.

[6]. Sahu, S., Chawla, M., & Khare, N. (2017, May). “*An efficient analysis of crop yield prediction using Hadoop framework based on random forest approach*”. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 53-57). IEEE.

[7]. Bhanumathi, S., Vineeth, M., & Rohit, N. (2019, April). “*Crop Yield Prediction and Efficient use of Fertilizers*”. In 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0769-0773). IEEE.

[8]. Manoj Kumar Behera “*Assessment of the state of millets farming in India*”, Review Artical: MedCrave MOJ Ecology & Environmental Science, Volume 2 Issue 1 – 2017.