VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI, KARNATAKA-590 018.

A MINI PROJECT REPORT
ON
STORY APP

PLACEMENT MANAGEMENT SYSTEM(SAMPLE)

Submitted in partial fulfilment of the requirements for the Mini Project (18CSMP68) course of the 6th semester.

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

By

HARSH MISHRA	(1JS20CS068)
KARUNA SAGAR	(1JS20CS077)

Under the guidance of

Mrs. B.N. Rashmi

Assistant professor, Dept. of CSE,

JSS Academy of Technical Education, Bengaluru

JSS MAHAVIDYAPEETHA, MYSURU

JSS Academy of Technical Education

JSS Campus, Uttarahalli Kengeri Main Road, Bengaluru – 560060

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled "**Story App**" is a bona fide work carried out by **Mr. Karuna Sagar(1JS20CS077)** and **Mr. Harsh Mishra(1JS20CS078)** in partial fulfillment of the requirements for the course Mobile Application Development of 6th semester, Bachelor of engineering in Computer Science and engineering of the Visvesvaraya Technological University, Belagavi, during the academic year 2022 – 2023. It is certified that all corrections and suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Mrs. B.N. Rashmi	**Dr. Mallikarjuna P.B.** B.E, MTech, Ph.D.
Assistant professor, Dept. of CSE, JSSATE, Bengaluru	**Asso Professor and Head, Dept. of CSE JSSATE, Bengaluru**

Name of the examiners	**Signature with date**
1.	....
2.	....

ABSTRACT

This project introduces "StoryApp," a user-friendly mobile application developed using Kotlin. StoryApp is designed to provide users with a simple and immersive experience of exploring and reading captivating stories.

The main focus of StoryApp is to offer users a diverse range of story options to choose from. Upon launching the application, users are greeted with a clean and intuitive interface that displays various story genres such as fantasy, mystery, romance, science fiction, and more. Each genre showcases a collection of stories for users to explore and enjoy.

The simplicity of StoryApp's design allows users to effortlessly browse through the available story options. Tapping on a genre of interest reveals a list of stories within that genre. Users can then select a story that catches their attention and begin reading it immediately. StoryApp aims to create a distraction-free reading environment, allowing users to fully immerse themselves in the stories. The application offers features like adjustable font sizes, customizable backgrounds, and a night mode for comfortable reading in various lighting conditions.

While StoryApp primarily focuses on providing an extensive collection of stories, it also includes a bookmarking feature that allows users to save their progress within a story. This feature enables users to easily resume reading from where they left off without any hassle.
Although StoryApp doesn't incorporate interactive elements or social sharing features, its simplicity and focus on providing a seamless reading experience make it an ideal platform for users who appreciate the pleasure of reading captivating stories without distractionsThis project introduces "StoryApp," a user-friendly mobile application developed using Kotlin. StoryApp is designed to provide users with a simple and immersive experience of exploring and reading captivating stories.

The main focus of StoryApp is to offer users a diverse range of story options to choose from. Upon launching the application, users are greeted with a clean and intuitive interface that displays various story genres such as fantasy, mystery, romance, science fiction, and more. Each genre showcases a collection of stories for users to explore and enjoy.

The simplicity of StoryApp's design allows users to effortlessly browse through the available story options. Tapping on a genre of interest reveals a list of stories within that genre. Users can then

select a story that catches their attention and begin reading it immediately. StoryApp aims to create a distraction-free reading environment, allowing users to fully immerse themselves in the stories. The application offers features like adjustable font sizes, customizable backgrounds, and a night mode for comfortable reading in various lighting conditions.

While StoryApp primarily focuses on providing an extensive collection of stories, it also includes a bookmarking feature that allows users to save their progress within a story. This feature enables users to easily resume reading from where they left off without any hassle.

ACKNOWLEDGEMENTS

I express my humble greetings to his holiness **Jagadguru Sri Shivarathri Deshikendra Mahaswamijigalavaru** who has showered their blessings on us for framing our career successfully.

The completion of any project involves the efforts of many people. I've been lucky to have received a lot of help and support from Indian Railway aspirants and all other quarters, during the making of this project. I take this opportunity to acknowledge all those, whose guidance and encouragement helped me emerge successful.

I express our sincere thanks to our beloved principal, **Dr. Bhimasen Soragaon** for having supported us in our academic endeavors.

I am thankful to the resourceful guidance, timely assistance and graceful gesture of my guide **Mrs. B.N. Rashmi,** Assistant professor, Department of Computer Science and engineering, who helped me in every aspect of my project work.

I am also indebted to **Dr. Mallikarjuna P.B.**, Associate Professor and head of department of Computer Science and engineering for the facilities and support extended towards us.

Last but not the least, I am pleased to express our heart full thanks to all the teaching and nonteaching staff of department of Computer and Science engineering and my friends who have rendered their help, motivation and support.

KARUNA SAGAR (1JS20CS077)

HARSH MISHRA (1JS20CS068)

TABLE OF CONTENTS

Chapter no.	Chapter Name	Page no.
1	**Introduction**	**7**
1.1	Overview	7
1.2	Problem Statement	7
1.3	Motivation	7
1.4	MAD Technologies	8
1.5	Application of MAD Tech.	9
2	**Requirement Specification**	**10**
2.1	Software Specification	10
2.2	Hardware Specification	10
2.3	User Characteristics	10
3	**System Design**	**11**
3.1	Proposed System	11
3.2	Features Of System	11
3.3	Design	12
4	**Implementation**	**13**
4.1	Module Description	13
4.2	Activity file:	14
4.3	Activity_Splash	15
4.4	Custom Item View	16

4.5	Details.kt	18
4.6	Item Adapter	19
4.7	Main Activity	20
4.8	Splash.kt	22
4.9	Graddle File	22
4.10	Login Activity	23
4.11	Signup Activity	25
5	**Results**	**27**
	Conclusion	
	Future Enhancement	
	References	

CHAPTER-1

INTRODUCTION

1.1 Overview

The "StoryApp" is a mobile application development project aimed at creating an engaging and interactive platform for users to explore and enjoy various stories.

The app seeks to provide a captivating storytelling experience through a user-friendly interface and a wide range of story genres. With StoryApp, users can dive into fictional and non-fictional narratives, participate in interactive storytelling, and share their favorite stories with others.

1.2 Problem Statement

The problem statement for the StoryApp project revolves around the limitations of existing story apps,
including the lack of interactive and immersive experiences, limited diversity and curation of stories,
inefficient search and discovery mechanisms, inadequate social sharing and community engagement features, and the need for effective and non-intrusive monetization strategies.
The project aims to overcome these challenges by creating a story app that provides engaging storytelling experiences, a wide range of diverse and curated stories, efficient search and discovery capabilities, seamless social sharing options, and appropriate monetization methods.

1.3 Motivation

The motivation behind building this story app is to ignite imagination, promote literacy, provide entertainment and escape, foster emotional connections, and embrace technological advancements.

It aims to offer users an immersive and engaging platform for exploring captivating narratives, encouraging creativity, and cultivating a love for reading. By combining the power of storytelling with interactive features, the app seeks to create a transformative and enjoyable experience for its users

1.4 MAD Technologies

- In the mini project, the following mobile application development technologies are used: o Kotlin: Kotlin is the primary programming language used for Android application development. It is a modern, expressive, and interoperable language that is fully supported by the Android platform. Kotlin offers concise syntax, null safety, and improved code readability, making it a popular choice for developing Android applications.
- Android Studio: Android Studio is the official Integrated Development Environment (IDE) for Android app development. It provides a comprehensive set of tools and features specifically designed for building Android applications. Android Studio offers features like code editing, debugging, testing, and performance profiling, making it the preferred choice for Android developers.
- OCR Libraries: The mini project involves utilizing external OCR libraries to extract text from images. These libraries provide the necessary algorithms and functionality for optical character recognition. Examples of popular OCR libraries for Android development include Tesseract, Google Mobile Vision OCR, and ABBYY FineReader.
- Google Play Services: Google Play Services is a set of APIs and services provided by Google that allow developers to integrate various Google functionalities into their Android applications. In the mini project, Google Play Services may be used for features such as Google Search integration, Google Sign-In, or accessing Google Cloud services for image processing and OCR..
- Internet Connectivity: The application requires internet connectivity to perform Google searches and potentially communicate with external OCR services. Standard Android

technologies, such as the Network API or libraries like Retrofit or Volley, may be utilized to establish internet connections and handle network requests.

- These technologies form the foundation for building the Android application that extracts text from photos and integrates with Google search. They provide the necessary tools, frameworks, and services to develop a robust, user-friendly, and feature-rich mobile application for text extraction and search functionality.

1.5 Application of Mobile Application Development

- **Simplicity**- In UI/UX Design: A well-designed and user-friendly interface is essential for a story app .Implementing simple UI designs with intuitive navigation and easy-to-use features will enhance the user experience, ensuring that readers can effortlessly engage with the app and immerse themselves in the stories.

- **Best Performance Loading Speed**: For a story app, optimal performance and fast loading times are crucial .Users expect quick access to stories, smooth transitions between pages, and minimal lag. Mobile app development ensures efficient coding and optimization to deliver the best possible performance, keeping readers engaged and immersed in the storytelling experience.

- **Customization Features:** Story apps can provide customization options to enhance the user experience. Users may have the ability to personalize the app's appearance, such as selecting preferred themes, fonts, or reading modes. Offering customization features allows readers to tailor the app to their preferences, creating a more personalized and enjoyable reading experience.

- **Push Notifications:** Notifications play a significant role in a story app, keeping readers informed and engaged .Users can receive notifications about new story releases, updates.

CHAPTER-2

REQUIRMENT SPECIFICATION

2.1 SOFTWARE SPECIFICATION

Operating System: Windows Vista 7/8/10

Software: Android Studio

Language: Kotlin

2.2 HARDWARE SPECIFICATION

Processor: X86 Compatible processor with 1.7GHz clock

Speed Ram: 4GB or greater Hard

Disk: 400 GB min

Monitor: VGA/SVGA

Keyboard: 104 keys standard

Mouse: 2/3 button optical / mechanical

Smart Phone

2.3 USER CHARACTERISTICS

Every user

- Should be comfortable with the basic working of the computer.
- Must have basic knowledge of English.
- Must have skills of Android Studio and Kotlin.

CHAPTER-3

SYSTEM DESIGN

3.1 Proposed System

In this paper, we introduce a simple and user-friendly story app that aims to provide an enjoyable reading experience without the need for complex sign-up or login processes. The app focuses on delivering captivating stories to users in a straightforward manner, allowing them to dive right into the world of storytelling.

3.2 Features of System

The story app is designed with simplicity in mind, offering the following key features:

- Story Collection: The app offers a curated collection of engaging stories across various genres. Users can easily browse through the available stories and select the ones that pique their interest.

- Intuitive Navigation: The app provides a seamless and intuitive navigation system, allowing users to effortlessly explore different stories and switch between chapters or sections with ease.

- Readability: The app ensures that stories are presented in a clear and readable format. The text is
- optimized for legibility, and users can adjust font size or choose from different reading backgrounds to suit their preferences.

- Story Progress: The app remembers users' progress within each story, enabling them to seamlessly continuereading from where they left off without any hassle.

- Offline Reading: The story app allows users to download stories and access them offline, ensuring that they can enjoy their favorite tales even without an internet connection.

3.3 Design

The design of the story app emphasizes simplicity and ease of use, providing an uncomplicated reading experience for users:

Minimalistic Interface: The app features a clean and minimalist interface, avoiding unnecessary clutter and distractions. The focus is on the stories themselves, allowing users to fully immerse themselves in the narratives.

One-Tap Access: Users can quickly access their favorite stories or the last read story with a single tap, eliminating the need for complex navigation menus.

Visual Enhancements: While keeping the design simple, the app may incorporate subtle visual enhancements such as elegant typography, appealing story cover images, and gentle animations to create an aesthetically pleasing reading environment

CHAPTER-4

Implementation

4.1 Module Description

- Requirements Gathering: Define the objectives of your simple story app.
- Design and Wireframing: Create a clean and intuitive design for your app's user interface (UI) and user experience (UX).
- Frontend Development: Use a mobile app development framework like Kotlin to build the frontend of your story app.
- Story Management: Create a collection of stories with basic details such as title, author, and content.
- User Interactions: Implement features that allow users to browse and read stories.
- Bookmarking: Add a bookmarking feature that allows users to save their progress in a story.
- Testing: Perform thorough testing of your app to ensure functionality and a smooth user experience.

4.2 Activity file:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Details"
android:orientation="vertical"> <ScrollView
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<LinearLayout android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical" >

<ImageView
android:id="@+id/storyFeatureImage"
android:layout_width="match_parent"
android:layout_height="250dp"
android:scaleType="fitXY"
app:srcCompat="@drawable/nature" />

<TextView android:id="@+id/storyDetails"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:fontFamily="cursive"
android:text="TextView"
android:textColor="#252525"
android:textSize="24sp"
android:textStyle="bold" />
</LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

4.3 ACTIVITY_SPLASH

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" android:background="@drawable/naturebanner"
tools:context=".Splash">

<ProgressBar android:id="@+id/progressBar"
style="?android:attr/progressBarStyle"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.97" />

<TextView android:id="@+id/textView2"
android:layout_width="287dp"
android:layout_height="126dp"
android:fontFamily="serif"
android:text="StoryApp   By : HARSH MISHRA     KARUNA SAGAR"
android:textAlignment="center" android:textColor="#131313"
android:textSize="34sp" android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.362"
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.011" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

4.4 CUSTOM ITEM VIEW

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.andr
oid.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="wrap_content">

<androidx.cardview.widget.CardView android:layout_width="0dp"



android:layout_height="wrap_content"
android:layout_marginBottom="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<androidx.constraintlayout.widget.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="match_parent">

<androidx.cardview.widget.CardView
android:id="@+id/cardView"
```

```
android:layout_width="150dp"
android:layout_height="150dp"
android:layout_marginStart="8dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="8dp"
android:layout_marginBottom="8dp"
app:cardCornerRadius="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<ImageView android:id="@+id/cardImage"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:scaleType="fitXY"
app:srcCompat="@drawable/nature" />
</androidx.cardview.widget.CardView>

<TextView android:id="@+id/cardTitle"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginStart="16dp"
android:layout_marginEnd="16dp"
android:text="Sample Story Title"
android:textSize="20sp"
android:textStyle="bold"
```

```
app:layout_constraintBottom_toBottomOf="@+id/cardView"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/cardView"
app:layout_constraintTop_toTopOf="@+id/cardView"
app:layout_constraintVertical_bias="0.04000002" />

<TextView android:id="@+id/cardContent"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginStart="16dp"
android:layout_marginTop="8dp"
android:layout_marginEnd="16dp" android:maxLines="3"
android:text="Sample Story content"
android:textSize="18sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@+id/cardView"
app:layout_constraintTop_toBottomOf="@+id/cardTitle" />
</androidx.constraintlayout.widget.ConstraintLayout>
</androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

4.5 DETAILS.KT

```
package co.smallacademy.storyappv2
importandroidx.appcompat.app.AppCompatActivity
importandroid.os.Bundle
importandroid.view.MenuItem
importcom.squareup.picasso.Picasso
```

```
import kotlinx.android.synthetic.main.activity_details.*

class Details : AppCompatActivity() { override fun
onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_details)

valsTitle = intent.getStringExtra("storyTitle")
valsContent = intent.getStringExtra("storyContent")
valsImage = intent.getStringExtra("storyImage")
supportActionBar?.title = sTitle
supportActionBar?.setDisplayHomeAsUpEnabled(true)

Picasso.get().load(sImage).into(storyFeatureImage) storyDetails.text
= sContent

   }

override fun onOptionsItemSelected(item: MenuItem): Boolean {
if(item.itemId == android.R.id.home){ onBackPressed()
       }
returnsuper.onOptionsItemSelected(item)
   }
}
```

4.6 ITEM ADAPTER

```
package co.smallacademy.storyappv2
importandroid.content.Intent
importandroid.view.LayoutInflater
importandroid.view.View
importandroid.view.ViewGroup
importandroid.widget.ImageView
importandroid.widget.TextView
```

```
importandroidx.recyclerview.widget.RecyclerView importcom.squareup.picasso.Picasso

classItemAdapter(valstoryTitles: Array<String>, valstoryContents: Array<String>,
valstoryImages: Array<String>) : RecyclerView.Adapter<ItemAdapter.ViewHolder>() {

classViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
valcardTitle: TextView = itemView.findViewById(R.id.cardTitle)
valcardContent: TextView = itemView.findViewById(R.id.cardContent)
valcardImage : ImageView = itemView.findViewById(R.id.cardImage) val
view = itemView
}
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
val view =
LayoutInflater.from(parent.context).inflate(R.layout.custom_item_view,parent,false)
returnViewHolder(view) }

override fun onBindViewHolder(holder: ViewHolder, position: Int) {
holder.cardTitle.text = storyTitles[position] holder.cardContent.text
= storyContents[position]
Picasso.get().load(storyImages[position]).into(holder.cardImage)

holder.view.setOnClickListener{
// Toast.makeText(holder.view.context,"Item Number -> " +
position,Toast.LENGTH_SHORT).show()
val intent = Intent(it.context,Details::class.java)
intent.putExtra("storyTitle",storyTitles[position])
intent.putExtra("storyContent",storyContents[position
])
intent.putExtra("storyImage",storyImages[position])
holder.view.context.startActivity(intent)
                    }
                }

            override fun getItemCount(): Int {
returnstoryTitles.size
   }
}
```

4.7 MAIN ACTIVITY

```
package co.smallacademy.storyappv2 importandroid.content.Intent
importandroidx.appcompat.app.AppCompatActivity
importandroid.os.Bundle
importandroid.view.MenuItem
importandroidx.appcompat.app.ActionBarDrawerToggle
importandroidx.core.view.GravityCompat
importandroidx.recyclerview.widget.LinearLayoutManager
importcom.google.android.material.navigation.NavigationView
import kotlinx.android.synthetic.main.activity_main.* import
kotlinx.android.synthetic.main.content_main.* import
java.util.*
importkotlin.random.Random

classMainActivity                    :                    AppCompatActivity(),
NavigationView.OnNavigationItemSelectedListener {
varstoryTitles = arrayOf<String>() varstoryContents
= arrayOf<String>()
```

```
varstoryImages = arrayOf<String>()

override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)
setSupportActionBar(toolbar)
= ActionBarDrawerToggle(this,drawerLayout,toolbar,R.string.open,R.string.close)
toggle.isDrawerIndicatorEnabled = true drawerLayout.addDrawerListener(toggle)
toggle.syncState()

navigationView.setNavigationItemSelectedListener(this)

storyTitles = resources.getStringArray(R.array.storyTitles) storyContents
= resources.getStringArray(R.array.storyContents) storyImages =
resources.getStringArray(R.array.storyImages)


val          adapter          =          ItemAdapter(storyTitles,storyContents,storyImages)
storyList.layoutManager = LinearLayoutManager(this)
storyList.adapter = adapter
  }

override fun onNavigationItemSelected(item: MenuItem): Boolean {
drawerLayout.closeDrawer(GravityCompat.START)
if(item.itemId == R.id.random){ valrandPosition =
Random.nextInt(0,storyTitles.size) val intent =
Intent(applicationContext,Details::class.java)
intent.putExtra("storyTitle",storyTitles[randPosition])
intent.putExtra("storyContent",storyContents[randPosition])
intent.putExtra("storyImage",storyImages[randPosition])
startActivity(intent)


    }
return true
  }
}
```

4.8 SPLASH.KT

```
package co.smallacademy.storyappv2 import
android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle import
android.os.Handler
import android.os.Looper

class Splash : AppCompatActivity() {
  override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_splash)

    Handler(Looper.getMainLooper()).postDelayed({
      startActivity(Intent(applicationContext,MainActivity::class.java))
},2000)
  }
}
```

4.9 GRADDLE FILE

```
plugins {
   id     'com.android.application'
id 'kotlin-android'
   id 'kotlin-android-extensions'
}

android {
compileSdkVersion 30
buildToolsVersion "30.0.2"

   defaultConfig {
      applicationId "co.smallacademy.storyappv2" minSdkVersion
         25
         targetSdkVersion                30
         versionCode
versionName "1.0"

          testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
   }
minSdkVersion 25
      targetSdkVersion 30
      versionCode 1 }
```

4.10 LOGIN ACTIVITY

package co.smallacademy.storyappv2

```
import android.content.Context import
android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle import
android.widget.Button import
android.widget.Toast import
androidx.room.Room import
androidx.room.RoomDatabase
import com.google.android.material.textfield.TextInputEditText import
java.lang.Exception

class LoginActivity : AppCompatActivity() {

  override        fun        onCreate(savedInstanceState:        Bundle?)        {
super.onCreate(savedInstanceState)
     setContentView(R.layout.activity_login)

     val emailField = findViewById<TextInputEditText>(R.id.email_field)
val passwordField = findViewById<TextInputEditText>(R.id.password_field)
val gotosignupButton = findViewById<Button>(R.id.gotosignup_button)     val
loginButton = findViewById<Button>(R.id.login_button)


     val                              db                              =
Room.databaseBuilder(this,AppDb::class.java,"appdb").allowMainThreadQueries().build()
val userDao = db.userDao
```

```
    gotosignupButton.setOnClickListener {
      val intent = Intent(this,SignupActivity::class.java)
startActivity(intent)
    }

    loginButton.setOnClickListener {

      if (emailField.text.isNullOrBlank()){
        Toast.makeText(this, "Please enter email", Toast.LENGTH_SHORT).show()
return@setOnClickListener
      }

      if (passwordField.text.isNullOrBlank()){
        Toast.makeText(this, "Please enter password", Toast.LENGTH_SHORT).show()
return@setOnClickListener
      }

      if ((passwordField.text?.toString()?.length ?:0) < 6){
        Toast.makeText(this, "Password must be 6 characters",
Toast.LENGTH_SHORT).show()
        return@setOnClickListener
      }

      val email = emailField.text?.trim().toString()
val password = passwordField.text?.trim().toString()

      try {
            val  user  =  userDao.getUser(email,  password)
if (user == null){
               Toast.makeText(this, "User does not exist", Toast.LENGTH_SHORT).show()
            }else{
               val prefs = getSharedPreferences("main", Context.MODE_PRIVATE)
               prefs.edit().putString("email",email).commit()
val intent = Intent(this,MainActivity::class.java)
startActivity(intent)
               finish()
            }
         }catch                         (e:Exception){
e.printStackTrace()
            Toast.makeText(this, "Something went wrong!", Toast.LENGTH_SHORT).show()
         }
      }


   }
}
```

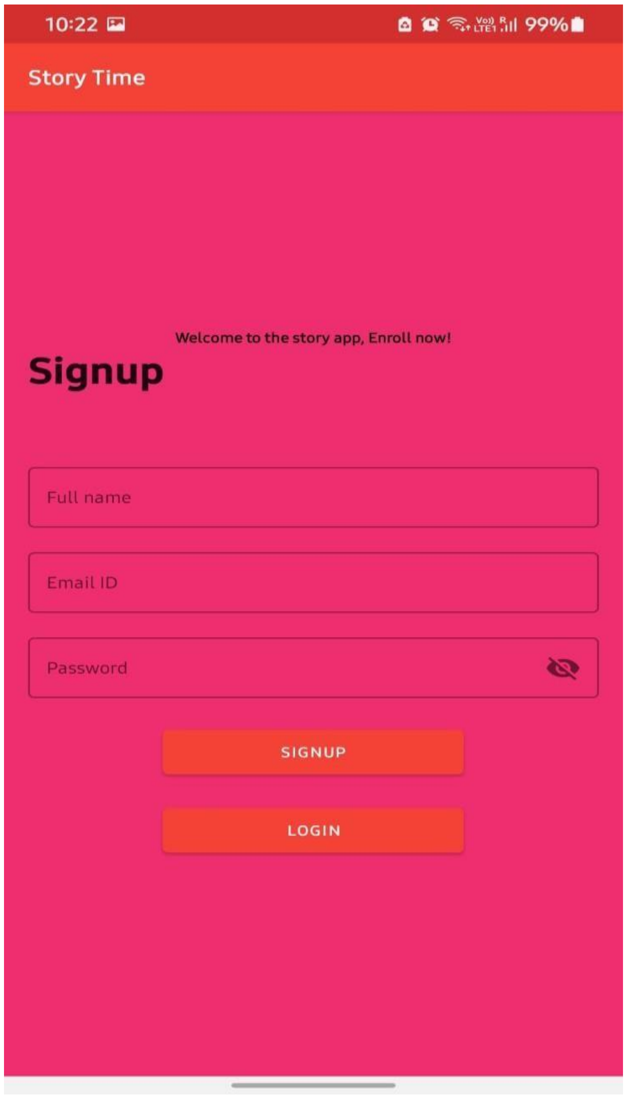
4.11 SIGNUP ACTIVITY

```
package co.smallacademy.storyappv2


import android.content.Context
import android.content.Intent
import android.os.Bundle import
android.widget.Button import
android.widget.Toast
```

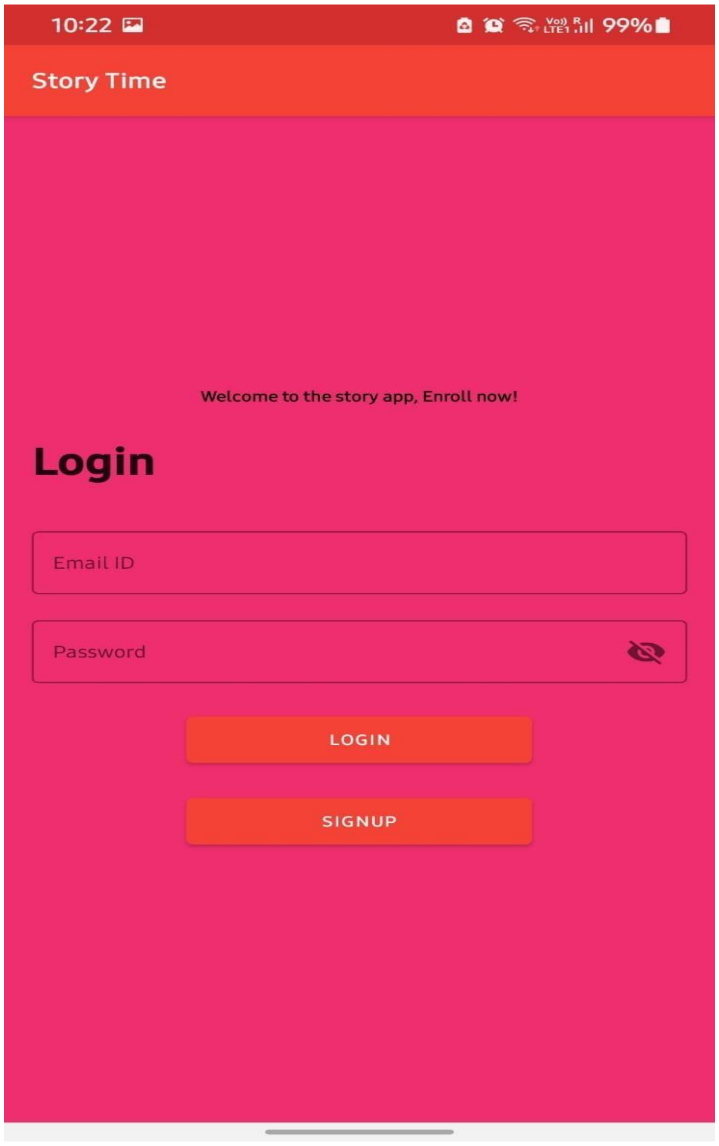
```
import androidx.appcompat.app.AppCompatActivity import
androidx.room.Room
import com.google.android.material.textfield.TextInputEditText

class SignupActivity : AppCompatActivity() {

  override        fun        onCreate(savedInstanceState:        Bundle?)        {
super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_signup)

      val nameField = findViewById<TextInputEditText>(R.id.name_field);
val emailField = findViewById<TextInputEditText>(R.id.email_field);      val
passwordField = findViewById<TextInputEditText>(R.id.password_field);      val
signupButton = findViewById<Button>(R.id.signup_button)      val
gotologinButton = findViewById<Button>(R.id.gotologin_button)



      val db =
        Room.databaseBuilder(this,                                AppDb::class.java,
"appdb").allowMainThreadQueries().build()      val userDao = db.userDao

      gotologinButton.setOnClickListener {
        val        intent        =        Intent(this,LoginActivity::class.java)
startActivity(intent)
      }

      signupButton.setOnClickListener {
        if (nameField.text.isNullOrBlank()) {
          Toast.makeText(this,   "Please   enter   name",   Toast.LENGTH_SHORT).show()
return@setOnClickListener
        }

        if (emailField.text.isNullOrBlank()) {
          Toast.makeText(this,   "Please   enter   email",   Toast.LENGTH_SHORT).show()
return@setOnClickListener
        }

        if (passwordField.text.isNullOrBlank()) {
          Toast.makeText(this,   "Please   enter   password",   Toast.LENGTH_SHORT).show()
return@setOnClickListener
        }

        if ((passwordField.text?.toString()?.length ?: 0) < 6) {
          Toast.makeText(this,     "Password     must     be     6     characters",
Toast.LENGTH_SHORT).show()
          return@setOnClickListener
        }

        val name = nameField.text.toString()
val email = emailField.text.toString()          val
password = passwordField.text.toString()

        try {
          val  user  =  User(name  =  name,  email  =  email,  password  =  password)
userDao.insertUser(user)
```

```
            val prefs = getSharedPreferences("main", Context.MODE_PRIVATE)
prefs.edit().putString("email",email).commit()

            val intent = Intent(this, MainActivity::class.java)
startActivity(intent)
            finish()

        } catch (e: Exception) {
e.printStackTrace()
            Toast.makeText(this, "User already exists", Toast.LENGTH_SHORT).show()
        }
      }
   }

}
```

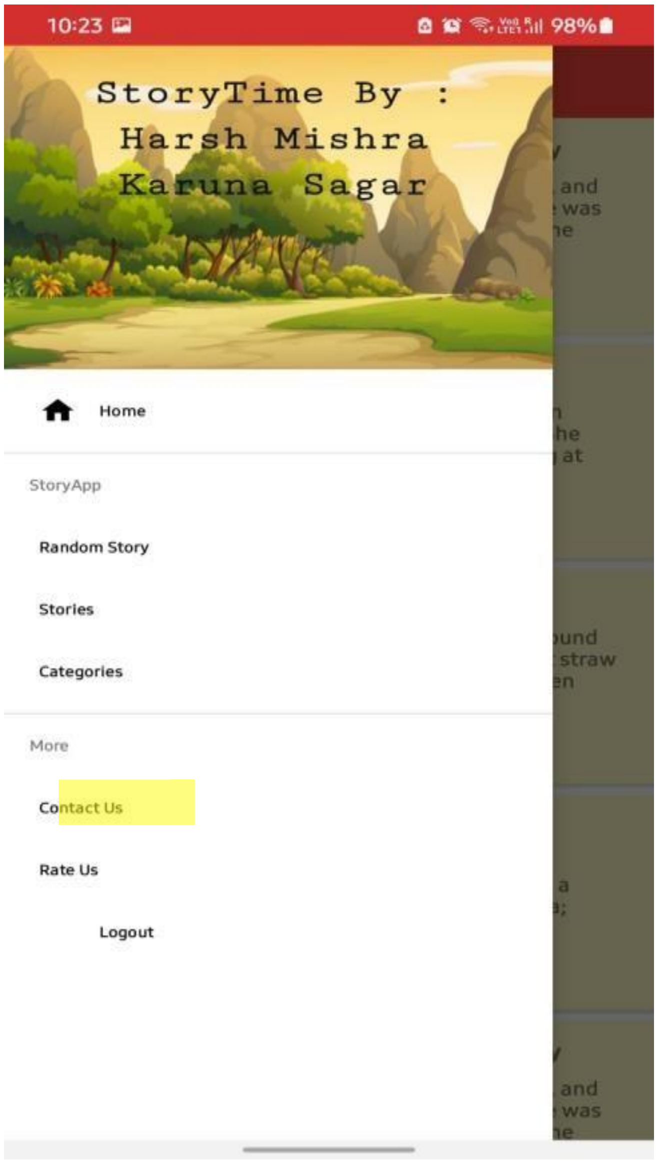
CHAPTER 5: RESULTS



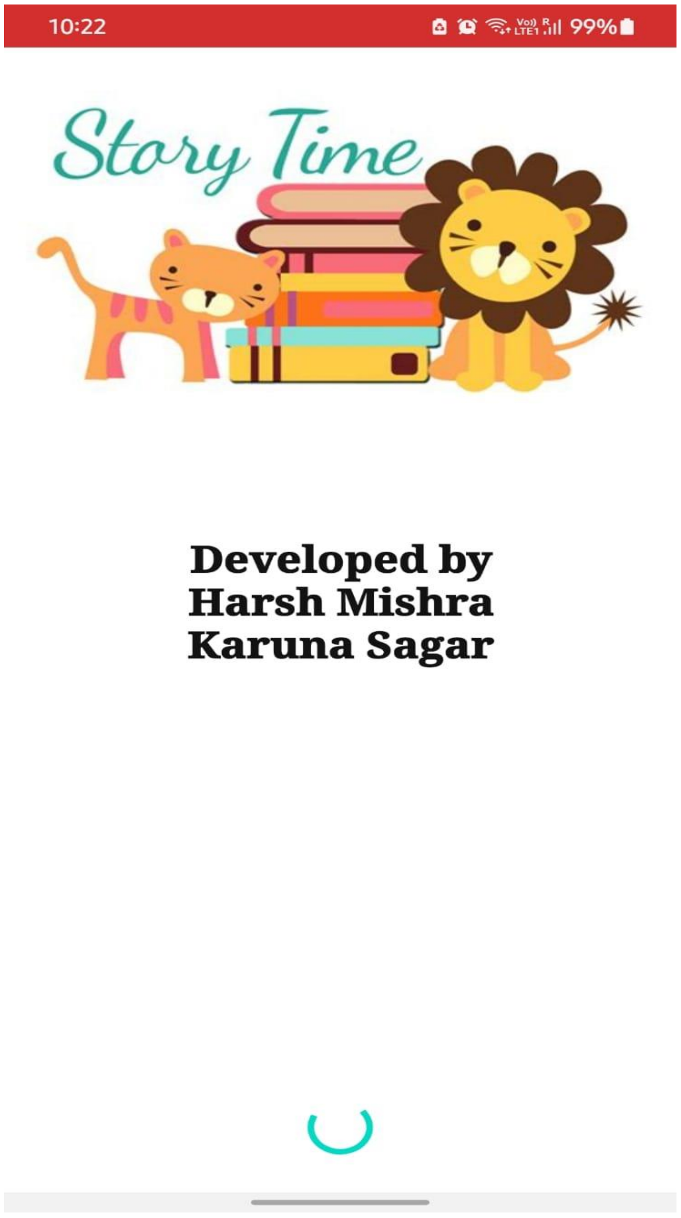
- The sign-up page offers a simple user interface and has 2 plaintext boxes, one for entering username and the other for password.
- The user has to fill in the username and password fields with valid strings and click on signup button, upon clicking the signup button the user's credentials are now saved and the user can proceed to LoginActivity page by clicking on the 'back to login' button.



- The user has to enter his/her credentials as entered in the signup page for a successful login. Upon entering valid details and clicking on the login button the user gets to see a toast message saying "login successful" and is directed to second page which is the MainActivity module or else the user will see a toast message saying "invalid credentials" and the application remains in the same page.
- There is also a text view that says "Don't have an account? signup", upon clicking on this textview the user is directed to the signup activity where the user can create an account by specifying the necessary credentials.

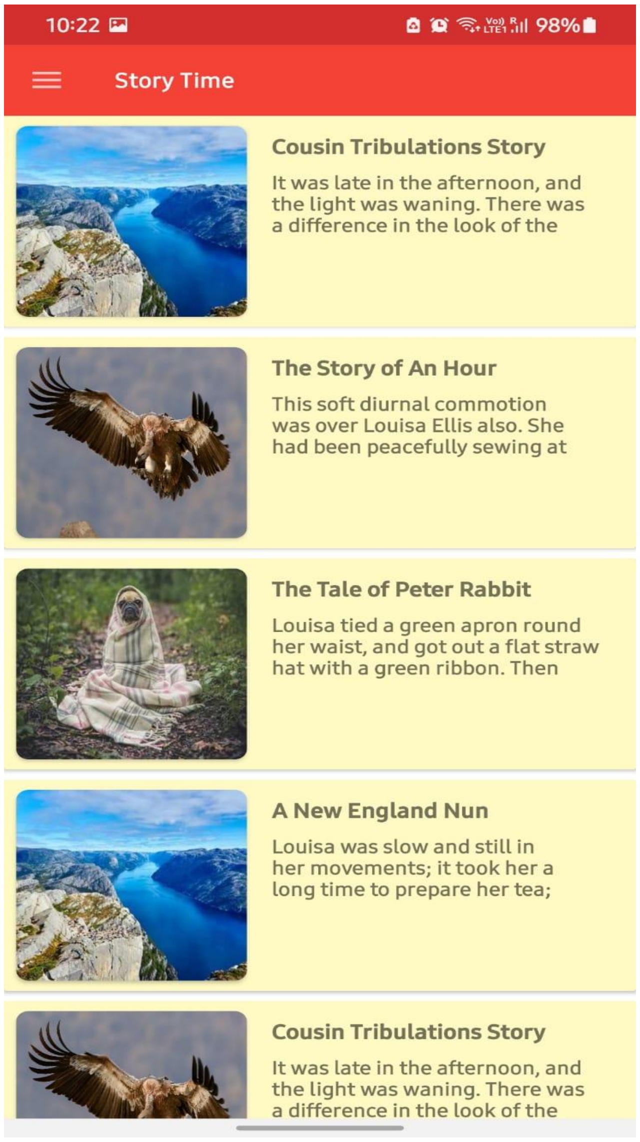


- The home page offers a simple user interface and has options for stories ,categories, and the logout option.
- The user can click on Logout button upon clicking the logout button the user's credentials are now saved and the user can proceed to LoginActivity/SignIN Activity page by clicking on the 'back to login' button.



Developed by
Harsh Mishra
Karuna Sagar

- Welcome Page of the App. .It contains the name of the App and the Developers.



Cousin Tribulations Story
It was late in the afternoon, and the light was waning. There was a difference in the look of the

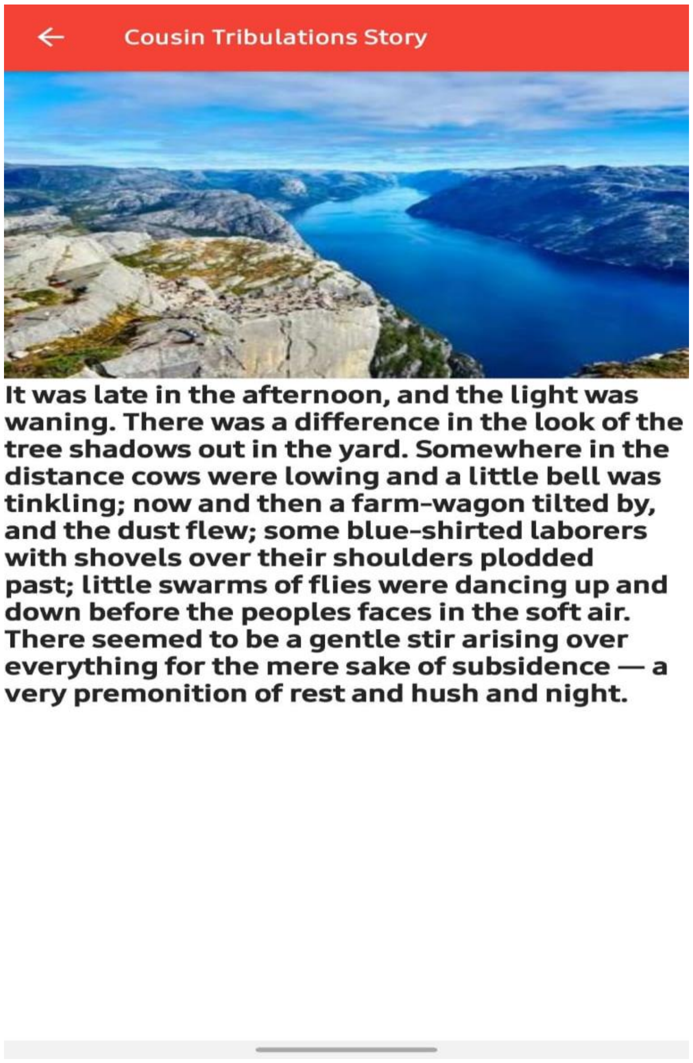
The Story of An Hour
This soft diurnal commotion was over Louisa Ellis also. She had been peacefully sewing at

The Tale of Peter Rabbit
Louisa tied a green apron round her waist, and got out a flat straw hat with a green ribbon. Then

A New England Nun
Louisa was slow and still in her movements; it took her a long time to prepare her tea;

Cousin Tribulations Story
It was late in the afternoon, and the light was waning. There was a difference in the look of the

- This is the Menu for users to choose the story they want to read.



It was late in the afternoon, and the light was waning. There was a difference in the look of the tree shadows out in the yard. Somewhere in the distance cows were lowing and a little bell was tinkling; now and then a farm-wagon tilted by, and the dust flew; some blue-shirted laborers with shovels over their shoulders plodded past; little swarms of flies were dancing up and down before the peoples faces in the soft air. There seemed to be a gentle stir arising over everything for the mere sake of subsidence — a very premonition of rest and hush and night.

- This is the story page ,after selecting the story .

- The user can read the story in this page.

Conclusion

The development and implementation of a simple story app offer numerous opportunities for engaging users in the captivating world of storytelling. By focusing on simplicity, usability, and an intuitive user interface, the app provides a seamless reading experience for users of all ages and interests.

The story app, with its minimalistic design and curated collection of stories, allows users to explore various genres and immerse themselves in captivating narratives. The future enhancements, such as personalized recommendations, social media integration, and interactive elements, will further enhance the app's appeal and keep users engaged.

With the potential for continuous improvement and expansion, the story app has the ability to grow and adapt to the changing needs and preferences of its users. By gathering user feedback, incorporating innovative features, and fostering a sense of community, the app can become a goto platform for reading enthusiasts and storytelling enthusiasts alike.
Overall, the story app serves as a digital gateway to the enchanting world of stories, offering users an accessible and enjoyable way to discover, read, and share their favorite narratives. Whether it's a short story, a novel, or a serialized series, the app brings the joy of reading and storytelling to users' fingertips, making it an invaluable companion for those seeking literary adventures.

FUTURE ENHANCEMENTS

Personalized Recommendations: Implement a recommendation system that suggests stories based on users' reading preferences, browsing history, and ratings. This personalized feature helps users discover new stories aligned with their interests, enhancing their overall reading experience.

Social Media Integration: Enable users to connect their social media accounts to the app, allowing them to share their favorite stories, quotes, or recommendations with their friends and followers. This integration enhances user engagement and promotes the app through social sharing.

Reading Progress Sync: Introduce a syncing feature that allows users to seamlessly switch between devices and continue reading from where they left off. This feature keeps track of users' reading progress and bookmarks,ensuring a smooth reading experience across different devices.

Audio Narration: Add an audio narration feature that enables users to listen to the stories instead of reading them. This feature caters to users who prefer audio content and expands the accessibility of the app to those with visual impairments.

Interactive Elements: Enhance the reading experience by incorporating interactive elements such as illustrations, animations, or mini-games within the stories. These interactive elements make thestories more immersive and engaging for users.

User-generated Content: Introduce a platform for users to submit their own stories and share them with the community. This user-generated content feature encourages creativity, fosters a sense of community, and expands the app's content library.

Advanced Search and Filters: Improve the search functionality by implementing advanced filters such as sorting by popularity, release date, or author. This helps users find stories more efficiently based on their specific preferences.

Offline Mode: Enable users to download stories for offline reading, allowing them to access their favorite stories even without an internet connection. This feature is beneficial for users who travel or have limited access to the internet.

References:

1. Donald Hearn & Pauline Baker: Computer Graphics with OpenGL Version,3rd / 4th Edition, Pearson Education,2011.
2. Edward Angel: Interactive Computer Graphics- A Top Down approach with OpenGL, 5th edition. Pearson Education, 2008.
3. Wikipedia: The free encyclopaedia.
4. Geeks for Geeks and TutorialsPoint.

****