**AboutActivity.java**

package com.example.mabs;  
  
import android.os.Bundle;  
import android.widget.ImageView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class AboutActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_about*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(v -> finish());  
 }  
}

**BookingActivity.java**

package com.example.mabs;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.CalendarView;  
import android.widget.CheckBox;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.android.gms.common.api.Status;  
import com.google.android.gms.location.FusedLocationProviderClient;  
import com.google.android.gms.location.LocationServices;  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
import com.google.android.libraries.places.api.Places;  
import com.google.android.libraries.places.api.model.Place;  
import com.google.android.libraries.places.api.net.PlacesClient;  
import com.google.android.libraries.places.widget.AutocompleteSupportFragment;  
import com.google.android.libraries.places.widget.listener.PlaceSelectionListener;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
import java.util.Locale;  
  
  
  
public class BookingActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 private CheckBox checkBox1, checkBox2, checkBox3;  
 private CalendarView calView;  
 private RecyclerView tslotView;  
 private Button confirmBtn;  
 private GoogleMap pMap;  
 private TextView selectedLocationText;  
 private SupportMapFragment mapFragment;  
 private FusedLocationProviderClient locationClient;  
  
 private static final int *LOCATION\_PERMISSION\_REQUEST\_CODE* = 101;  
 private static final String *TAG* = "BookingActivity";  
  
 private String selectedDate;  
  
 private String selectedTimeSlot;  
  
 private String selectedLocationName;  
 private String selectedPlaceId;  
  
 private List<String> selectedServices = new ArrayList<>();  
 private LatLng selectedLocationLatLng;  
  
 private TextView totalPriceTextView;  
 private int totalPrice = 0;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_booking*);  
  
 // Initialize total price  
 totalPriceTextView = findViewById(R.id.*labelPrice*);  
  
 // Initialize CheckBoxes  
 checkBox1 = findViewById(R.id.*checkBox1*);  
 checkBox2 = findViewById(R.id.*checkBox2*);  
 checkBox3 = findViewById(R.id.*checkBox3*);  
 // Retrieve from intent  
 Intent intent = getIntent();  
 boolean isGCSelected = intent.getBooleanExtra("General Consultation", false);  
 boolean isSCSelected = intent.getBooleanExtra("Specialist Consultation", false);  
 boolean isDLSelected = intent.getBooleanExtra("Diagnostic and Lab Test Scheduling", false);  
  
 // Update checkboxes  
 checkBox1.setChecked(isGCSelected);  
 checkBox2.setChecked(isSCSelected);  
 checkBox3.setChecked(isDLSelected);  
  
 // Update selectedServices list and total price  
 updateService("General Consultation", isGCSelected, 25);  
 updateService("Specialist Consultation", isSCSelected, 50);  
 updateService("Diagnostic and Lab Test Scheduling", isDLSelected, 30);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 ImageView homeButton = findViewById(R.id.*imgHome*);  
 homeButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent homeIntent = new Intent(BookingActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 }  
 });  
  
 // Places API initialized  
 if (!Places.*isInitialized*()) {  
 Places.*initialize*(getApplicationContext(), "AIzaSyCNRn9Cm3zz1yhBNBgI9kLNDDLVWuHU7qw");  
 }  
  
 setupPlacesAutocomplete();  
  
 // Map fragment initialized  
 mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*map*);  
 mapFragment.getMapAsync(this);  
  
 // Location client Initialized  
 locationClient = LocationServices.*getFusedLocationProviderClient*(this);  
  
 // UI Initialized  
 setupUIComponents();  
 }  
  
 private void setupPlacesAutocomplete() {  
 PlacesClient placesClient = Places.*createClient*(this);  
 AutocompleteSupportFragment autocompleteFragment = (AutocompleteSupportFragment)  
 getSupportFragmentManager().findFragmentById(R.id.*autocomplete\_fragment*);  
  
 autocompleteFragment.setPlaceFields(Arrays.*asList*(Place.Field.*ID*, Place.Field.*NAME*, Place.Field.*LAT\_LNG*));  
 autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {  
 @Override  
 public void onPlaceSelected(@NonNull Place place) {  
 updateMapLocation(place.getLatLng(), place.getName());  
 selectedLocationName = place.getName();  
 selectedPlaceId = place.getId();  
 selectedLocationText.setText("Selected: " + selectedLocationName + " - " + selectedPlaceId);  
 }  
  
 @Override  
 public void onError(@NonNull Status status) {  
 Log.*i*("PlacesAPI Error", "An error occurred: " + status);  
 }  
 });  
 }  
  
 private void updateMapLocation(LatLng latLng, String placeName) {  
 if (latLng != null) {  
 pMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(latLng, 12));  
 pMap.addMarker(new MarkerOptions().position(latLng).title(placeName));  
 }  
 }  
  
 private void setupUIComponents() {  
 checkBox1 = findViewById(R.id.*checkBox1*);  
 checkBox2 = findViewById(R.id.*checkBox2*);  
 checkBox3 = findViewById(R.id.*checkBox3*);  
 calView = findViewById(R.id.*calView*);  
 tslotView = findViewById(R.id.*timeslotRecycler*);  
 confirmBtn = findViewById(R.id.*confirmButton*);  
 selectedLocationText = findViewById(R.id.*selectedLocationText*);  
 totalPriceTextView = findViewById(R.id.*labelPrice*);  
  
 checkBox1.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 updateService("General Consultation", isChecked, 25);  
 });  
  
 checkBox2.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 updateService("Specialist Consultation", isChecked, 50);  
 });  
  
 checkBox3.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 updateService("Diagnostic and Lab Test Scheduling", isChecked, 30);  
 });  
 setupCalendarView();  
 setupTimeSlots();  
  
 confirmBtn.setOnClickListener(v -> {  
 if (selectedDate == null || selectedTimeSlot == null || selectedLocationName == null || selectedServices.isEmpty()) {  
 Toast.*makeText*(BookingActivity.this, "Please select all options before confirming.", Toast.*LENGTH\_LONG*).show();  
 } else {  
 proceedToConfirmation();  
 }  
 });  
 }  
  
 private void updateService(String service, boolean add, int price) {  
 if (add && !selectedServices.contains(service)) {  
 selectedServices.add(service);  
 totalPrice += price;  
 } else if (!add && selectedServices.contains(service)) {  
 selectedServices.remove(service);  
 totalPrice -= price;  
 }  
 totalPriceTextView.setText("$" + totalPrice); // Update the text view with the new total price  
 }  
  
 private void proceedToConfirmation() {  
  
 Intent confirmIntent = new Intent(BookingActivity.this, ConfirmationActivity.class);  
 confirmIntent.putStringArrayListExtra("services", new ArrayList<>(selectedServices));  
 confirmIntent.putExtra("location", selectedLocationName);  
 confirmIntent.putExtra("date", selectedDate);  
 confirmIntent.putExtra("time", selectedTimeSlot);  
 confirmIntent.putExtra("locationLatLng", selectedLocationLatLng);  
 startActivity(confirmIntent);  
 }  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 pMap = googleMap;  
 setupMapSettings();  
 enableMyLocation();  
 }  
  
 private void setupMapSettings() {  
 pMap.getUiSettings().setZoomControlsEnabled(true);  
 }  
  
 private void enableMyLocation() {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*}, *LOCATION\_PERMISSION\_REQUEST\_CODE*);  
 } else {  
 pMap.setMyLocationEnabled(true);  
 getLastLocation();  
 }  
 }  
 private void getLastLocation() {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) == PackageManager.*PERMISSION\_GRANTED*) {  
 locationClient.getLastLocation().addOnSuccessListener(this, location -> {  
 if (location != null) {  
 LatLng userLocation = new LatLng(location.getLatitude(), location.getLongitude());  
 pMap.animateCamera(CameraUpdateFactory.*newLatLngZoom*(userLocation, 12));  
 }  
 });  
 }  
 }  
  
   
 private void setupCalendarView() {  
 calView.setOnDateChangeListener((view, year, month, dayOfMonth) -> {  
 selectedDate = String.*format*(Locale.*getDefault*(), "%d-%02d-%02d", year, month + 1, dayOfMonth);  
 Log.*d*("Calendar", "Selected date: " + selectedDate);  
 });  
 }  
  
 private void setupTimeSlots() {  
 ArrayList<String> timeSlots = new ArrayList<>();  
 timeSlots.add("09:00 AM");  
 timeSlots.add("10:00 AM");  
 timeSlots.add("11:00 AM");  
 timeSlots.add("12:00 PM");  
 timeSlots.add("03:00 PM");  
 timeSlots.add("04:00 PM");  
 timeSlots.add("05:00 PM");  
 timeSlots.add("06:00 PM");  
 timeSlots.add("07:00 PM");  
 timeSlots.add("08:00 PM");  
  
 TimeSlotAdapter adapter = new TimeSlotAdapter(timeSlots, new TimeSlotAdapter.OnTimeSlotClickListener() {  
 @Override  
 public void onTimeSlotClick(String timeSlot, int position) {  
 selectedTimeSlot = timeSlot;  
 }  
 });  
 tslotView.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.*HORIZONTAL*, false));  
 tslotView.setAdapter(adapter);  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if (requestCode == *LOCATION\_PERMISSION\_REQUEST\_CODE*) {  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 enableMyLocation();  
 } else {  
 Toast.*makeText*(this, "Permission denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}

**ConfirmationActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.provider.CalendarContract;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.Locale;  
  
  
public class ConfirmationActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 private ArrayList<String> servicesList;  
 private String datePicked;  
 private String timeSlotPicked;  
 private String locationName;  
 private LatLng locationLatLng;  
  
 private GoogleMap pMap;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_confirmation*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 ImageView homeButton = findViewById(R.id.*imgHome*);  
 homeButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent homeIntent = new Intent(ConfirmationActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 }  
 });  
  
 TextView tvConfirmationMessage = findViewById(R.id.*tvConfirmationMessage*);  
 TextView tvAppointmentDetails = findViewById(R.id.*tvAppointmentDetails*);  
 Button btnAddToCalendar = findViewById(R.id.*btnAddToCalendar*);  
 Button btnCancel = findViewById(R.id.*btnCancel*);  
 TextView tvService = findViewById(R.id.*tvService*);  
 TextView tvCalendar = findViewById(R.id.*tvCalendar*);  
 TextView tvLocation = findViewById(R.id.*tvLocation*);  
  
 // Retrieval of data from the intent  
 Intent intent = getIntent();  
 servicesList = intent.getStringArrayListExtra("services");  
 datePicked = intent.getStringExtra("date");  
 timeSlotPicked = intent.getStringExtra("time");  
 locationName = intent.getStringExtra("location");  
 locationLatLng = intent.getParcelableExtra("locationLatLng");  
 Log.*d*("ConfirmationActivity", "Received LatLng: " + locationLatLng);  
  
 updateMapWithMarker();  
  
 // null or empty list handling  
 String serviceList = servicesList == null || servicesList.isEmpty()  
 ? "No services selected"  
 : TextUtils.*join*(", ", servicesList);  
  
  
 String confirmationMessage = "Your appointment for the following services is confirmed and approved: " + serviceList;  
 tvConfirmationMessage.setText(confirmationMessage);  
  
 String appointmentDetails = serviceList + " at " + timeSlotPicked + " on " + datePicked + " at " + locationName;  
 tvAppointmentDetails.setText(appointmentDetails);  
  
 tvService.setText(serviceList);  
 tvCalendar.setText(datePicked + " " + timeSlotPicked);  
 tvLocation.setText(locationName);  
  
  
 SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*map*);  
 if (mapFragment != null) {  
 mapFragment.getMapAsync(this);  
 }  
  
 btnAddToCalendar.setOnClickListener(v -> addToCalendar(datePicked, timeSlotPicked, locationName, serviceList));  
 btnCancel.setOnClickListener(v -> finish());  
 }  
  
  
 private void addToCalendar(String date, String time, String location, String service) {  
 Calendar beginTime = Calendar.*getInstance*();  
 SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd hh:mm a", Locale.*getDefault*());  
 try {  
 Date startDate = dateFormat.parse(date + " " + time);  
 beginTime.setTime(startDate);  
 } catch (ParseException e) {  
 Toast.*makeText*(this, "Error parsing date/time", Toast.*LENGTH\_SHORT*).show();  
 e.printStackTrace();  
 return;  
 }  
 Calendar endTime = (Calendar) beginTime.clone();  
 endTime.add(Calendar.*HOUR*, 1);  
  
 Intent intent = new Intent(Intent.*ACTION\_INSERT*)  
 .setData(CalendarContract.Events.*CONTENT\_URI*)  
 .putExtra(CalendarContract.*EXTRA\_EVENT\_BEGIN\_TIME*, beginTime.getTimeInMillis())  
 .putExtra(CalendarContract.*EXTRA\_EVENT\_END\_TIME*, endTime.getTimeInMillis())  
 .putExtra(CalendarContract.Events.*TITLE*, service + " Appointment")  
 .putExtra(CalendarContract.Events.*DESCRIPTION*, service + " appointment at " + location)  
 .putExtra(CalendarContract.Events.*EVENT\_LOCATION*, location)  
 .putExtra(CalendarContract.Events.*AVAILABILITY*, CalendarContract.Events.*AVAILABILITY\_BUSY*);  
 startActivity(intent);  
 }  
  
 private void updateMapWithMarker() {  
 Log.*d*("ConfirmationActivity", "Updating Map Marker: " + locationLatLng);  
 if (pMap != null && locationLatLng != null) {  
 pMap.clear(); // Clears any existing markers on the map  
 pMap.addMarker(new MarkerOptions().position(locationLatLng).title(locationName));  
 pMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(locationLatLng, 15));  
 } else {  
 Log.*d*("ConfirmationActivity", "Map or LatLng is null");  
 }  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 pMap = googleMap;  
 Log.*d*("ConfirmationActivity", "Map is ready. LatLng: " + locationLatLng);  
 updateMapWithMarker();  
 }  
  
  
  
}

**Event.java**

package com.example.mabs;  
  
public class Event {  
 private String title;  
 private String timeRange;  
  
  
 public Event(String title, String dateString, String timeString) {  
 this.title = title;  
 this.timeRange = timeRange;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
  
 public String getTimeRange() {  
 return timeRange;  
 }  
}

**EventAdapter.java**

package com.example.mabs;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ArrayAdapter;  
import android.widget.TextView;  
  
import java.util.List;  
  
public class EventAdapter extends ArrayAdapter<Event> {  
 private List<Event> events;  
  
 public EventAdapter(Context context, List<Event> events) {  
 super(context, R.layout.*item\_event*, events);  
 this.events = events;  
 }  
  
  
 @Override  
 public View getView(int position, View convertView, ViewGroup parent) {  
 ViewHolder holder;  
 if (convertView == null) {  
 // Inflate the layout for each item  
 convertView = LayoutInflater.*from*(getContext()).inflate(R.layout.*item\_event*, parent, false);  
 holder = new ViewHolder();  
 // Set up to contain the views  
 holder.title = convertView.findViewById(R.id.*tvEventTitle*);  
 holder.timeRange = convertView.findViewById(R.id.*tvEventTimeRange*);  
 convertView.setTag(holder);  
 } else {  
 holder = (ViewHolder) convertView.getTag();  
 }  
  
  
 Event event = getItem(position);  
  
  
 if (event != null) {  
 holder.title.setText(event.getTitle());  
 holder.timeRange.setText(event.getTimeRange());  
 }  
  
  
 return convertView;  
 }  
  
  
 static class ViewHolder {  
 TextView title;  
 TextView timeRange;  
 }  
}

**HomeActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.viewpager2.widget.ViewPager2;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class HomeActivity extends AppCompatActivity {  
  
 private ViewPager2 viewPagerImages;  
 private final Handler sliderHandler = new Handler();  
 private final List<Integer> imageList = new ArrayList<>();  
 private final long SLIDE\_DELAY\_MS = 3000; // milliseconds  
  
 // sliding images  
 private final Runnable slideRunnable = new Runnable() {  
 @Override  
 public void run() {  
 int currentItem = viewPagerImages.getCurrentItem();  
 int totalItems = imageList.size();  
 if (viewPagerImages != null && totalItems > 0) {  
 viewPagerImages.setCurrentItem((currentItem + 1) % totalItems, true);  
 }  
 }  
 };  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home*);  
  
 // image list  
 imageList.add(R.drawable.*vp1*);  
 imageList.add(R.drawable.*vp2*);  
 imageList.add(R.drawable.*vp3*);  
  
 // ViewPager2  
 viewPagerImages = findViewById(R.id.*viewPagerImages*);  
 PictureAdapter pictureAdapter = new PictureAdapter(this, imageList);  
 viewPagerImages.setAdapter(pictureAdapter);  
  
  
 startImageSlider();  
  
  
 setupCardClickListeners();  
 }  
  
 private void startImageSlider() {  
  
 sliderHandler.removeCallbacks(slideRunnable);  
  
 sliderHandler.postDelayed(slideRunnable, SLIDE\_DELAY\_MS);  
 }  
  
 private void setupCardClickListeners() {  
  
 findViewById(R.id.*cardWalk*).setOnClickListener(v -> {  
 Intent intent = new Intent(HomeActivity.this, ReminderActivity.class);  
 startActivity(intent);  
 });  
  
  
 findViewById(R.id.*cardGrooming*).setOnClickListener(v -> {  
 Intent intent = new Intent(HomeActivity.this, ServicesActivity.class);  
 startActivity(intent);  
  
 });  
  
 findViewById(R.id.*cardPetCare*).setOnClickListener(v -> {  
 Intent intent = new Intent(HomeActivity.this, QuickCareActivity.class);  
 startActivity(intent);  
 });  
  
 findViewById(R.id.*cardDayCare*).setOnClickListener(v -> {  
 Intent intent = new Intent(HomeActivity.this, InpatientActivity.class);  
 startActivity(intent);  
 });  
  
  
 findViewById(R.id.*cardAbout*).setOnClickListener(v -> {  
 Intent intent = new Intent(HomeActivity.this, AboutActivity.class);  
 startActivity(intent);  
 });  
  
  
 findViewById(R.id.*cardProfile*).setOnClickListener(v -> {  
 Intent intent = new Intent(HomeActivity.this, ProfileActivity.class);  
 startActivity(intent);  
 });  
 }  
  
 @Override  
 protected void onPause() {  
 super.onPause();  
  
 sliderHandler.removeCallbacks(slideRunnable);  
 }  
  
 @Override  
 protected void onResume() {  
 super.onResume();  
  
 startImageSlider();  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
  
 sliderHandler.removeCallbacksAndMessages(null);  
 }  
}

**InPatientActivity.java**

package com.example.mabs;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.CalendarView;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
  
import com.google.android.gms.common.api.Status;  
import com.google.android.gms.location.FusedLocationProviderClient;  
import com.google.android.gms.location.LocationServices;  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
import com.google.android.libraries.places.api.Places;  
import com.google.android.libraries.places.api.model.Place;  
import com.google.android.libraries.places.api.net.PlacesClient;  
import com.google.android.libraries.places.widget.AutocompleteSupportFragment;  
import com.google.android.libraries.places.widget.listener.PlaceSelectionListener;  
  
import java.text.SimpleDateFormat;  
import java.util.Arrays;  
import java.util.Calendar;  
import java.util.Locale;  
  
public class InpatientActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 private Button fourHBtn, eightHBtn, twelveHBtn, twentyFourHBtn, bookBtn;  
 private TextView priceTV, selectedLocationText;  
 private CalendarView calView;  
 private GoogleMap pMap;  
 private FusedLocationProviderClient locationClient;  
  
 private static final int *LOCATION\_PERMISSION\_REQUEST\_CODE* = 101;  
 private static final String *TAG* = "InpatientActivity";  
  
 private double Price;  
 private int hoursPicked;  
 private String datePicked;  
 private String locationName;  
 private LatLng selectedLocationLatLng;  
  
 private String placeId;  
  
 private SupportMapFragment mapFragment;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_inpatient*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 ImageView homeButton = findViewById(R.id.*imgHome*);  
 homeButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent homeIntent = new Intent(InpatientActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 }  
 });  
  
 // Places API.  
 if (!Places.*isInitialized*()) {  
 Places.*initialize*(getApplicationContext(), getString(R.string.*google\_maps\_key*));  
 }  
  
 // Initialize map fragment.  
 mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*map*);  
 if (mapFragment != null) {  
 mapFragment.getMapAsync(this);  
 }  
  
 // Initialize location client.  
 locationClient = LocationServices.*getFusedLocationProviderClient*(this);  
  
 // Initialize UI components.  
 calView = findViewById(R.id.*calendarView*);  
 priceTV = findViewById(R.id.*priceTextView*);  
 selectedLocationText = findViewById(R.id.*selectedLocationText*);  
 fourHBtn = findViewById(R.id.*fourHoursButton*);  
 eightHBtn = findViewById(R.id.*eightHoursButton*);  
 twelveHBtn = findViewById(R.id.*twelveHoursButton*);  
 twentyFourHBtn = findViewById(R.id.*twentyFourHoursButton*);  
 bookBtn = findViewById(R.id.*confirmButton*);  
  
 setupCalendarView();  
 setupDurationButtons();  
 setupPlacesAutocomplete();  
  
 bookBtn.setOnClickListener(v -> proceedToConfirmation());  
 }  
  
 private void setupPlacesAutocomplete() {  
 PlacesClient placesClient = Places.*createClient*(this);  
 AutocompleteSupportFragment autocompleteFragment = (AutocompleteSupportFragment)  
 getSupportFragmentManager().findFragmentById(R.id.*autocomplete\_fragment*);  
  
 autocompleteFragment.setPlaceFields(Arrays.*asList*(Place.Field.*ID*, Place.Field.*NAME*, Place.Field.*LAT\_LNG*));  
 autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {  
 @Override  
 public void onPlaceSelected(@NonNull Place place) {  
 updateMapLocation(place.getLatLng(), place.getName());  
 locationName = place.getName();  
 placeId = place.getId();  
 selectedLocationText.setText("Selected: " + locationName + " - " + placeId);  
 }  
  
 @Override  
 public void onError(@NonNull Status status) {  
 Log.*i*("PlacesAPI Error", "An error occurred: " + status);  
 }  
 });  
 }  
  
 private void updateMapLocation(LatLng latLng, String placeName) {  
 if (latLng != null) {  
 pMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(latLng, 12));  
 pMap.addMarker(new MarkerOptions().position(latLng).title(placeName));  
 }  
 }  
 private void setupCalendarView() {  
 calView.setOnDateChangeListener((view, year, month, dayOfMonth) -> {  
 Calendar cal = Calendar.*getInstance*();  
 cal.set(year, month, dayOfMonth);  
 SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd", Locale.*getDefault*());  
 datePicked = dateFormat.format(cal.getTime());  
 });  
 }  
  
 private void setupDurationButtons() {  
 fourHBtn.setOnClickListener(v -> updatePrice(4)); // $50 for 4 hours  
 eightHBtn.setOnClickListener(v -> updatePrice(8)); // $100 for 8 hours  
 twelveHBtn.setOnClickListener(v -> updatePrice(12)); // $150 for 12 hours  
 twentyFourHBtn.setOnClickListener(v -> updatePrice(24)); // $200 for 24 hours  
 }  
  
 private void updatePrice(int hours) {  
 double basePrice = 50;  
 double price = basePrice;  
  
 if (hours > 4) {  
 int additionalBlocks = (hours - 4) / 4;  
 for (int i = 0; i < additionalBlocks; i++) {  
 price += basePrice;  
 }  
 }  
  
 Price = price;  
 hoursPicked = hours;  
 priceTV.setText(String.*format*(Locale.*getDefault*(), "Total Cost: $%.2f for %d hours", Price, hoursPicked));  
 }  
  
  
 private void proceedToConfirmation() {  
 Intent confirmIntent = new Intent(InpatientActivity.this, InpatientConfirmationActivity.class);  
 confirmIntent.putExtra("selectedPrice", Price);  
 confirmIntent.putExtra("selectedHours", hoursPicked);  
 confirmIntent.putExtra("selectedDate", datePicked);  
 confirmIntent.putExtra("selectedLocationName", locationName);  
 confirmIntent.putExtra("selectedLocationLatLng", selectedLocationLatLng);  
 startActivity(confirmIntent);  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 pMap = googleMap;  
 enableMyLocation();  
 }  
  
 private void enableMyLocation() {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*}, *LOCATION\_PERMISSION\_REQUEST\_CODE*);  
 return;  
 }  
 pMap.setMyLocationEnabled(true);  
 getLastLocation();  
 }  
  
 private void getLastLocation() {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) == PackageManager.*PERMISSION\_GRANTED*) {  
 locationClient.getLastLocation().addOnSuccessListener(this, location -> {  
 if (location != null) {  
 LatLng userLocation = new LatLng(location.getLatitude(), location.getLongitude());  
 pMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(userLocation, 15));  
 }  
 });  
 }  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if (requestCode == *LOCATION\_PERMISSION\_REQUEST\_CODE* && grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 enableMyLocation();  
 } else {  
 Toast.*makeText*(this, "Permission denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**InPatientConfirmationActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.provider.CalendarContract;  
import android.view.View;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
  
import java.util.Calendar;  
  
public class InpatientConfirmationActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 private String locationName;  
 private LatLng locationLatLng;  
 private GoogleMap pMap;  
 private double Price;  
 private int hoursPicked;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_dcconfirm*);  
  
 TextView tvConfirmationMessage = findViewById(R.id.*tvConfirmationMessage*);  
 TextView tvPrice = findViewById(R.id.*tvPrice*);  
 TextView tvDuration = findViewById(R.id.*tvHours*);  
 TextView tvLocation = findViewById(R.id.*tvLocation*);  
 Button btnCancel = findViewById(R.id.*btnCancel*);  
 Button btnAddToSchedule = findViewById(R.id.*btnAddToCalendar*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 ImageView homeButton = findViewById(R.id.*imgHome*);  
 homeButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent homeIntent = new Intent(InpatientConfirmationActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 }  
 });  
  
  
 Intent intent = getIntent();  
 locationName = intent.getStringExtra("selectedLocationName");  
 locationLatLng = intent.getParcelableExtra("selectedLocationLatLng");  
 Price = intent.getDoubleExtra("selectedPrice", 0); // Make sure the key matches  
 hoursPicked = intent.getIntExtra("selectedHours", 0); // Make sure the key matches  
  
  
  
 tvConfirmationMessage.setText("InPatient Room booking at " + locationName + " is confirmed!");  
 tvPrice.setText(String.*format*("Total Cost: $%.2f", Price));  
 tvDuration.setText(String.*format*("Duration: %d hours", hoursPicked));  
 tvLocation.setText(locationName);  
  
 SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*map*);  
 if (mapFragment != null) {  
 mapFragment.getMapAsync(this);  
 }  
  
  
 btnAddToSchedule.setOnClickListener(v -> addToCalendar());  
  
  
 btnCancel.setOnClickListener(v -> finish());  
 }  
  
 private void addToCalendar() {  
 Calendar beginTime = Calendar.*getInstance*();  
 beginTime.setTimeInMillis(System.*currentTimeMillis*());  
  
  
 beginTime.add(Calendar.*HOUR\_OF\_DAY*, 2);  
  
 Calendar endTime = (Calendar) beginTime.clone();  
 endTime.add(Calendar.*HOUR\_OF\_DAY*, hoursPicked);  
  
 Intent intent = new Intent(Intent.*ACTION\_INSERT*)  
 .setData(CalendarContract.Events.*CONTENT\_URI*)  
 .putExtra(CalendarContract.*EXTRA\_EVENT\_BEGIN\_TIME*, beginTime.getTimeInMillis())  
 .putExtra(CalendarContract.*EXTRA\_EVENT\_END\_TIME*, endTime.getTimeInMillis())  
 .putExtra(CalendarContract.Events.*TITLE*, "InPatient Room Booking")  
 .putExtra(CalendarContract.Events.*DESCRIPTION*, "InPatient Room booked for " + hoursPicked + " hours.")  
 .putExtra(CalendarContract.Events.*EVENT\_LOCATION*, locationName)  
 .putExtra(CalendarContract.Events.*AVAILABILITY*, CalendarContract.Events.*AVAILABILITY\_BUSY*);  
  
 startActivity(intent);  
 }  
  
  
 private void updateMapWithMarker() {  
 if (pMap != null && locationLatLng != null) {  
 pMap.clear();  
 pMap.addMarker(new MarkerOptions().position(locationLatLng).title(locationName));  
 pMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(locationLatLng, 15));  
 }  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 pMap = googleMap;  
 updateMapWithMarker();  
 }  
}

**PatientProfile.java**

package com.example.mabs;  
  
public class PatientProfile {  
 private String patientName;  
 private String medicalRecordNumber;  
 private String age;  
 private String primaryPhysician;  
 private String insuranceProvider;  
 private String allergies;  
 private String specialNotes;  
 private String profileImageUrl;  
  
 public PatientProfile() { }  
  
 public PatientProfile(  
 String patientName,  
 String medicalRecordNumber,  
 String age,  
 String primaryPhysician,  
 String insuranceProvider,  
 String allergies,  
 String specialNotes,  
 String profileImageUrl  
 ) {  
 this.patientName = patientName;  
 this.medicalRecordNumber = medicalRecordNumber;  
 this.age = age;  
 this.primaryPhysician = primaryPhysician;  
 this.insuranceProvider = insuranceProvider;  
 this.allergies = allergies;  
 this.specialNotes = specialNotes;  
 this.profileImageUrl = profileImageUrl;  
 }  
  
 public String getPatientName() { return patientName; }  
 public void setPatientName(String v) { patientName = v; }  
  
 public String getMedicalRecordNumber() { return medicalRecordNumber; }  
 public void setMedicalRecordNumber(String v) { medicalRecordNumber = v; }  
  
 public String getAge() { return age; }  
 public void setAge(String v) { age = v; }  
  
 public String getPrimaryPhysician() { return primaryPhysician; }  
 public void setPrimaryPhysician(String v) { primaryPhysician = v; }  
  
 public String getInsuranceProvider() { return insuranceProvider; }  
 public void setInsuranceProvider(String v) { insuranceProvider = v; }  
  
 public String getAllergies() { return allergies; }  
 public void setAllergies(String v) { allergies = v; }  
  
 public String getSpecialNotes() { return specialNotes; }  
 public void setSpecialNotes(String v) { specialNotes = v; }  
  
 public String getProfileImageUrl() { return profileImageUrl; }  
 public void setProfileImageUrl(String v) { profileImageUrl = v; }  
}

**PictureAdapter.java**

package com.example.mabs;  
  
import android.content.Context;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
  
import androidx.annotation.NonNull;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.List;  
  
public class PictureAdapter extends RecyclerView.Adapter<PictureAdapter.ViewHolder> {  
  
 private List<Integer> images;  
 private LayoutInflater inflater;  
  
 public PictureAdapter(Context context, List<Integer> images) {  
 this.inflater = LayoutInflater.*from*(context);  
 this.images = images;  
 }  
  
 @NonNull  
 @Override  
 public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view = inflater.inflate(R.layout.*item\_view\_pager*, parent, false);  
 return new ViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull ViewHolder holder, int position) {  
 holder.imageView.setImageResource(images.get(position));  
 }  
  
 @Override  
 public int getItemCount() {  
  
 return images.size();  
 }  
  
 public static class ViewHolder extends RecyclerView.ViewHolder {  
 ImageView imageView;  
  
 ViewHolder(View itemView) {  
 super(itemView);  
 imageView = itemView.findViewById(R.id.*imageView*);  
 }  
 }  
}

**PlaceParser.java**

package com.example.mabs;  
  
import android.util.Log;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
  
public class PlaceParser {  
  
  
 public List<HashMap<String, String>> parse(JSONObject jObject) {  
  
 JSONArray jPlaces = null;  
 try {  
  
 jPlaces = jObject.getJSONArray("results");  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 return getPlaces(jPlaces);  
 }  
  
 private List<HashMap<String, String>> getPlaces(JSONArray jPlaces) {  
 int placesCount = jPlaces.length();  
 List<HashMap<String, String>> placesList = new ArrayList<HashMap<String, String>>();  
 HashMap<String, String> place;  
  
  
 for (int i = 0; i < placesCount; i++) {  
 try {  
  
 place = getPlace((JSONObject) jPlaces.get(i));  
 placesList.add(place);  
 } catch (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 return placesList;  
 }  
  
  
 private HashMap<String, String> getPlace(JSONObject jPlace) {  
 HashMap<String, String> place = new HashMap<>();  
 String placeName = "-NA-";  
 String vicinity = "-NA-";  
 String latitude;  
 String longitude;  
 String placeId = "";  
  
 try {  
 if (!jPlace.isNull("name")) {  
 placeName = jPlace.getString("name");  
 }  
 if (!jPlace.isNull("vicinity")) {  
 vicinity = jPlace.getString("vicinity");  
 }  
 if (!jPlace.isNull("place\_id")) {  
 placeId = jPlace.getString("place\_id");  
 }  
  
 latitude = jPlace.getJSONObject("geometry").getJSONObject("location").getString("lat");  
 longitude = jPlace.getJSONObject("geometry").getJSONObject("location").getString("lng");  
  
 place.put("place\_name", placeName);  
 place.put("vicinity", vicinity);  
 place.put("lat", latitude);  
 place.put("lng", longitude);  
 place.put("place\_id", placeId);  
  
 } catch (JSONException e) {  
 Log.*e*("PlaceParser", "Error parsing a place", e);  
 }  
 return place;  
 }  
}

**ProfileActivity.java**

package com.example.mabs;  
  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.net.Uri;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Toast;  
import androidx.activity.result.ActivityResultLauncher;  
import androidx.activity.result.contract.ActivityResultContracts;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.bumptech.glide.Glide;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.firestore.SetOptions;  
import com.google.firebase.storage.FirebaseStorage;  
import com.google.firebase.storage.StorageReference;  
  
import de.hdodenhof.circleimageview.CircleImageView;  
  
public class ProfileActivity extends AppCompatActivity {  
  
 private static final String *USERS\_COLLECTION* = "users";  
  
 private ImageView imgHome, imgLogout;  
 private CircleImageView imgProfilePicture;  
 private EditText editPatientName, editMedicalRecordNumber, editAge,  
 editPrimaryPhysician, editInsuranceProvider,  
 editAllergies, editSpecialNotes;  
 private Button btnEdit, btnConfirm;  
  
 private Uri selectedImageUri;  
 private PatientProfile profile;  
  
 private FirebaseAuth auth;  
 private FirebaseFirestore db;  
 private StorageReference storageRef;  
 private ActivityResultLauncher<Intent> pickImageLauncher;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_profile*);  
  
 auth = FirebaseAuth.*getInstance*();  
 db = FirebaseFirestore.*getInstance*();  
 storageRef = FirebaseStorage.*getInstance*().getReference("profile\_pics");  
  
 imgHome = findViewById(R.id.*imgHome*);  
 imgLogout = findViewById(R.id.*imgLogout*);  
 imgProfilePicture = findViewById(R.id.*imgProfilePicture*);  
 editPatientName = findViewById(R.id.*editPatientName*);  
 editMedicalRecordNumber = findViewById(R.id.*editMedicalRecordNumber*);  
 editAge = findViewById(R.id.*editAge*);  
 editPrimaryPhysician= findViewById(R.id.*editPrimaryPhysician*);  
 editInsuranceProvider = findViewById(R.id.*editInsuranceProvider*);  
 editAllergies = findViewById(R.id.*editAllergies*);  
 editSpecialNotes = findViewById(R.id.*editSpecialNotes*);  
 btnEdit = findViewById(R.id.*btnEdit*);  
 btnConfirm = findViewById(R.id.*btnConfirm*);  
  
 pickImageLauncher = registerForActivityResult(  
 new ActivityResultContracts.StartActivityForResult(),  
 result -> {  
 if (result.getResultCode() == *RESULT\_OK* && result.getData() != null) {  
 selectedImageUri = result.getData().getData();  
 Glide.*with*(this).load(selectedImageUri).into(imgProfilePicture);  
 }  
 }  
 );  
 imgProfilePicture.setOnClickListener(v -> {  
 Intent pick = new Intent(Intent.*ACTION\_GET\_CONTENT*);  
 pick.setType("image/\*");  
 pickImageLauncher.launch(pick);  
 });  
  
 imgHome.setOnClickListener(v -> finish());  
  
 imgLogout.setOnClickListener(v -> {  
 new AlertDialog.Builder(this)  
 .setTitle("Confirm Logout")  
 .setMessage("Are you sure you want to log out?")  
 .setPositiveButton("Yes", (d, w) -> {  
 auth.signOut();  
 startActivity(new Intent(this, SignInActivity.class)  
 .addFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*));  
 finish();  
 })  
 .setNegativeButton("No", null)  
 .show();  
 });  
  
 btnEdit.setOnClickListener(v -> setEditingEnabled(true));  
 btnConfirm.setOnClickListener(v -> {  
 setEditingEnabled(false);  
 saveProfileToFirebase(() ->  
 Toast.*makeText*(this, "Profile saved", Toast.*LENGTH\_SHORT*).show()  
 );  
 });  
  
 setEditingEnabled(false);  
 loadProfileFromFirebase();  
 }  
  
 private void setEditingEnabled(boolean enabled) {  
 editPatientName.setEnabled(enabled);  
 editMedicalRecordNumber.setEnabled(enabled);  
 editAge.setEnabled(enabled);  
 editPrimaryPhysician.setEnabled(enabled);  
 editInsuranceProvider.setEnabled(enabled);  
 editAllergies.setEnabled(enabled);  
 editSpecialNotes.setEnabled(enabled);  
 btnConfirm.setVisibility(enabled ? View.*VISIBLE* : View.*GONE*);  
 }  
  
 private void loadProfileFromFirebase() {  
 FirebaseUser user = auth.getCurrentUser();  
 if (user == null) return;  
  
 String uid = user.getUid();  
 Log.*d*("PROFILE", "Current UID: " + uid);  
  
 db.collection(*USERS\_COLLECTION*)  
 .document(uid)  
 .get()  
 .addOnSuccessListener(doc -> {  
 if (doc.exists()) {  
 profile = doc.toObject(PatientProfile.class);  
 populateUI();  
 }  
 })  
 .addOnFailureListener(e ->  
 Log.*e*("PROFILE", "Error loading profile", e)  
 );  
 }  
  
 private void populateUI() {  
 if (profile == null) return;  
 editPatientName.setText(profile.getPatientName());  
 editMedicalRecordNumber.setText(profile.getMedicalRecordNumber());  
 editAge.setText(profile.getAge());  
 editPrimaryPhysician.setText(profile.getPrimaryPhysician());  
 editInsuranceProvider.setText(profile.getInsuranceProvider());  
 editAllergies.setText(profile.getAllergies());  
 editSpecialNotes.setText(profile.getSpecialNotes());  
 String url = profile.getProfileImageUrl();  
 if (url != null && !url.isEmpty()) {  
 Glide.*with*(this).load(url).into(imgProfilePicture);  
 }  
 }  
  
 private void saveProfileToFirebase(Runnable onSuccess) {  
 FirebaseUser user = auth.getCurrentUser();  
 if (user == null) return;  
  
 String uid = user.getUid();  
 profile = new PatientProfile(  
 editPatientName.getText().toString().trim(),  
 editMedicalRecordNumber.getText().toString().trim(),  
 editAge.getText().toString().trim(),  
 editPrimaryPhysician.getText().toString().trim(),  
 editInsuranceProvider.getText().toString().trim(),  
 editAllergies.getText().toString().trim(),  
 editSpecialNotes.getText().toString().trim(),  
 profile != null ? profile.getProfileImageUrl() : ""  
 );  
  
 ProgressDialog pd = ProgressDialog.*show*(this, "Saving", "Please wait...", true);  
  
 Runnable writeProfile = () ->  
 db.collection(*USERS\_COLLECTION*)  
 .document(uid)  
 .set(profile, SetOptions.*merge*())  
 .addOnSuccessListener(a -> {  
 pd.dismiss();  
 onSuccess.run();  
 })  
 .addOnFailureListener(e -> {  
 pd.dismiss();  
 Toast.*makeText*(this, "Save failed: " + e.getMessage(), Toast.*LENGTH\_LONG*).show();  
 });  
  
 if (selectedImageUri != null) {  
 StorageReference picRef = storageRef.child(uid + ".jpg");  
 picRef.putFile(selectedImageUri)  
 .addOnSuccessListener(task -> picRef.getDownloadUrl()  
 .addOnSuccessListener(uri -> {  
 profile.setProfileImageUrl(uri.toString());  
 writeProfile.run();  
 })  
 .addOnFailureListener(e -> {  
 pd.dismiss();  
 Toast.*makeText*(this, "URL fetch failed", Toast.*LENGTH\_SHORT*).show();  
 })  
 )  
 .addOnFailureListener(e -> {  
 pd.dismiss();  
 Toast.*makeText*(this, "Image upload failed", Toast.*LENGTH\_SHORT*).show();  
 });  
 } else {  
 writeProfile.run();  
 }  
 }  
}

**QuickCareActivity.java**

package com.example.mabs;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.CalendarView;  
import android.widget.CheckBox;  
import android.widget.ImageView;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.recyclerview.widget.LinearLayoutManager;  
import androidx.recyclerview.widget.RecyclerView;  
  
import com.google.android.gms.common.api.Status;  
import com.google.android.gms.location.FusedLocationProviderClient;  
import com.google.android.gms.location.LocationServices;  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.SupportMapFragment;  
import com.google.android.gms.maps.model.BitmapDescriptorFactory;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.MarkerOptions;  
import com.google.android.libraries.places.api.Places;  
import com.google.android.libraries.places.api.model.Place;  
import com.google.android.libraries.places.api.net.PlacesClient;  
import com.google.android.libraries.places.widget.AutocompleteSupportFragment;  
import com.google.android.libraries.places.widget.listener.PlaceSelectionListener;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
import java.util.Locale;  
  
  
  
public class QuickCareActivity extends AppCompatActivity implements OnMapReadyCallback {  
  
 CheckBox checkBoxPR;  
 CheckBox checkBoxMA;  
  
 private List<String> selectedServices = new ArrayList<>();  
 private CalendarView calView;  
 private RecyclerView timeslotRV;  
 private Button confirmBtn;  
 private GoogleMap pMap;  
 private TextView selectedLocationText;  
 private SupportMapFragment mapFragment;  
 private FusedLocationProviderClient locationClient;  
  
 private static final int *LOCATION\_PERMISSION\_REQUEST\_CODE* = 101;  
 private static final String *TAG* = "QuickCareActivity";  
  
 private String datePicked;  
  
 private String timeslotPicked;  
  
 private String selectedLocationName;  
 private String PlaceId;  
  
 private String selectedType;  
  
 private LatLng selectedLocationLatLng;  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_quickcare*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 ImageView homeButton = findViewById(R.id.*imgHome*);  
 homeButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent homeIntent = new Intent(QuickCareActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 }  
 });  
  
 // Places API.  
 if (!Places.*isInitialized*()) {  
 Places.*initialize*(getApplicationContext(), "AIzaSyCNRn9Cm3zz1yhBNBgI9kLNDDLVWuHU7qw");  
 }  
  
 // Autocomplete Fragment  
 setupPlacesAutocomplete();  
  
 // map fragment  
 mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*map*);  
 mapFragment.getMapAsync(this);  
  
 // location client  
 locationClient = LocationServices.*getFusedLocationProviderClient*(this);  
  
 // Initialize UI components  
 setupUIComponents();  
 }  
  
 private void setupPlacesAutocomplete() {  
 PlacesClient placesClient = Places.*createClient*(this);  
 AutocompleteSupportFragment autocompleteFragment = (AutocompleteSupportFragment)  
 getSupportFragmentManager().findFragmentById(R.id.*autocomplete\_fragment*);  
  
 autocompleteFragment.setPlaceFields(Arrays.*asList*(Place.Field.*ID*, Place.Field.*NAME*, Place.Field.*LAT\_LNG*));  
 autocompleteFragment.setOnPlaceSelectedListener(new PlaceSelectionListener() {  
 @Override  
 public void onPlaceSelected(@NonNull Place place) {  
 updateMapLocation(place.getLatLng(), place.getName());  
 selectedLocationName = place.getName();  
 PlaceId = place.getId();  
 selectedLocationText.setText("Selected: " + selectedLocationName + " - " + PlaceId);  
 }  
  
 @Override  
 public void onError(@NonNull Status status) {  
 Log.*i*("PlacesAPI Error", "An error occurred: " + status);  
 }  
 });  
 }  
  
 private void updateMapLocation(LatLng latLng, String placeName) {  
 if (latLng != null) {  
 pMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(latLng, 12));  
 pMap.addMarker(new MarkerOptions().position(latLng).title(placeName));  
 }  
 }  
  
 private void setupUIComponents() {  
 checkBoxPR = findViewById(R.id.*checkBoxOP*);  
 checkBoxMA= findViewById(R.id.*checkBoxSurgery*);  
  
 calView = findViewById(R.id.*calendarView*);  
 timeslotRV = findViewById(R.id.*timeslotRecyclerView*);  
 confirmBtn = findViewById(R.id.*confirmButton*);  
 selectedLocationText = findViewById(R.id.*selectedLocationText*);  
  
 setupCheckBoxes();;  
 setupCalendarView();  
 setupTimeSlots();  
  
 confirmBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 // When confirm button is clicked  
 if (datePicked == null || timeslotPicked == null || selectedLocationName == null) {  
 // If any of the selections are null, notify the user to make a selection  
 Toast.*makeText*(QuickCareActivity.this, "Please select a date, time slot, and location.", Toast.*LENGTH\_LONG*).show();  
 } else {  
 proceedToConfirmation();  
 }  
 }  
 });  
 }  
  
 private void proceedToConfirmation() {  
 ArrayList<String> servicesList = new ArrayList<>(selectedServices); // Make sure it's ArrayList  
 Intent confirmIntent = new Intent(QuickCareActivity.this, ConfirmationActivity.class);  
 confirmIntent.putStringArrayListExtra("services", servicesList); // Correct method to use  
 confirmIntent.putExtra("location", selectedLocationName);  
 confirmIntent.putExtra("date", datePicked);  
 confirmIntent.putExtra("time", timeslotPicked);  
 confirmIntent.putExtra("locationLatLng", selectedLocationLatLng);  
 startActivity(confirmIntent);  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googleMap) {  
 pMap = googleMap;  
 setupMapSettings();  
 enableMyLocation();  
 }  
  
 private void setupMapSettings() {  
 pMap.getUiSettings().setZoomControlsEnabled(true);  
 }  
  
 private void enableMyLocation() {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*}, *LOCATION\_PERMISSION\_REQUEST\_CODE*);  
 } else {  
 pMap.setMyLocationEnabled(true);  
 getLastLocation();  
 }  
 }  
  
  
 private void getLastLocation() {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) == PackageManager.*PERMISSION\_GRANTED*) {  
 locationClient.getLastLocation().addOnSuccessListener(this, location -> {  
 if (location != null) {  
 LatLng userLocation = new LatLng(location.getLatitude(), location.getLongitude());  
 pMap.animateCamera(CameraUpdateFactory.*newLatLngZoom*(userLocation, 12));  
 }  
 });  
 }  
 }  
  
 private void fetchNearbyPlaces(LatLng location) {  
 String types = "veterinary\_care|pet\_care|pet\_hospital";  
 int radius = 50000; // 50 km radius  
 String apiKey = "AIzaSyCNRn9Cm3zz1yhBNBgI9kLNDDLVWuHU7qw"; // Use your actual API key  
 String url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json" +  
 "?location=" + location.latitude + "," + location.longitude +  
 "&radius=" + radius +  
 "&type=" + types +  
 "&key=" + apiKey;  
  
 new FetchPlacesTask().execute(url);  
 }  
  
 private class FetchPlacesTask extends AsyncTask<String, Void, String> {  
 @Override  
 protected String doInBackground(String... urls) {  
 try {  
 return downloadUrl(urls[0]);  
 } catch (Exception e) {  
 Log.*e*("Background Task", e.toString());  
 return null;  
 }  
 }  
  
 @Override  
 protected void onPostExecute(String result) {  
 try {  
 parseLocations(result);  
 } catch (JSONException e) {  
 Log.*e*("JSON Parser", "Error parsing data " + e.toString());  
 }  
 }  
  
 private void parseLocations(String json) throws JSONException {  
 JSONObject jsonObject = new JSONObject(json);  
 JSONArray resultsArray = jsonObject.getJSONArray("results");  
  
 for (int i = 0; i < resultsArray.length(); i++) {  
 JSONObject place = resultsArray.getJSONObject(i);  
 JSONObject location = place.getJSONObject("geometry").getJSONObject("location");  
 LatLng latLng = new LatLng(location.getDouble("lat"), location.getDouble("lng"));  
 String placeName = place.getString("name");  
  
 pMap.addMarker(new MarkerOptions()  
 .position(latLng)  
 .title(placeName)  
 .icon(BitmapDescriptorFactory.*defaultMarker*(BitmapDescriptorFactory.*HUE\_RED*)));  
 }  
  
 if (resultsArray.length() == 0) {  
 Log.*d*(*TAG*, "No places found.");  
 Toast.*makeText*(QuickCareActivity.this, "No pet-related places found nearby.", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 private String downloadUrl(String strUrl) throws IOException {  
 InputStream iStream = null;  
 HttpURLConnection urlConnection = null;  
 StringBuilder sb = new StringBuilder();  
 try {  
 URL url = new URL(strUrl);  
 urlConnection = (HttpURLConnection) url.openConnection();  
 urlConnection.connect();  
 iStream = urlConnection.getInputStream();  
 BufferedReader br = new BufferedReader(new InputStreamReader(iStream));  
 String line;  
 while ((line = br.readLine()) != null) {  
 sb.append(line);  
 }  
 } catch (Exception e) {  
 Log.*e*("Exception", e.toString());  
 } finally {  
 if (iStream != null) {  
 iStream.close();  
 }  
 urlConnection.disconnect();  
 }  
 return sb.toString();  
 }  
 }  
  
 private void setupCheckBoxes() {  
 // References to CheckBoxes in your layout  
 checkBoxPR = findViewById(R.id.*checkBoxOP*);  
 checkBoxMA = findViewById(R.id.*checkBoxSurgery*);  
  
  
 checkBoxPR.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 if (isChecked) {  
 selectedServices.add("Prescription Refill");  
 } else {  
 selectedServices.remove("Minor Ailments");  
 }  
 });  
  
  
 checkBoxMA.setOnCheckedChangeListener((buttonView, isChecked) -> {  
 if (isChecked) {  
 selectedServices.add("Minor Ailments");  
 } else {  
 selectedServices.remove("Minor Ailments");  
 }  
 });  
 }  
  
  
 private void setupCalendarView() {  
 calView.setOnDateChangeListener((view, year, month, dayOfMonth) -> {  
 datePicked = String.*format*(Locale.*getDefault*(), "%d-%02d-%02d", year, month + 1, dayOfMonth);  
 Log.*d*("Calendar", "Selected date: " + datePicked);  
 });  
 }  
  
 private void setupTimeSlots() {  
 ArrayList<String> timeSlots = new ArrayList<>();  
 timeSlots.add("09:00 AM");  
 timeSlots.add("10:00 AM");  
 timeSlots.add("11:00 AM");  
 timeSlots.add("12:00 PM");  
 timeSlots.add("03:00 PM");  
 timeSlots.add("04:00 PM");  
 timeSlots.add("05:00 PM");  
 timeSlots.add("06:00 PM");  
 timeSlots.add("07:00 PM");  
 timeSlots.add("08:00 PM");  
  
 TimeSlotAdapter adapter = new TimeSlotAdapter(timeSlots, new TimeSlotAdapter.OnTimeSlotClickListener() {  
 @Override  
 public void onTimeSlotClick(String timeSlot, int position) {  
 timeslotPicked = timeSlot;  
 }  
 });  
 timeslotRV.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.*HORIZONTAL*, false));  
 timeslotRV.setAdapter(adapter);  
 }  
  
 @Override  
 public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
 if (requestCode == *LOCATION\_PERMISSION\_REQUEST\_CODE*) {  
 if (grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 enableMyLocation();  
 } else {  
 Toast.*makeText*(this, "Permission denied", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 }  
}

**ReminderActivity.java**

package com.example.mabs;  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Bundle;  
import android.provider.CalendarContract;  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
public class ReminderActivity extends AppCompatActivity {  
 private static final int *PERMISSIONS\_REQUEST\_READ\_CALENDAR* = 100;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_calendar*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 ImageView homeButton = findViewById(R.id.*imgHome*);  
 homeButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent homeIntent = new Intent(ReminderActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 }  
 });  
  
 // ImageView  
 ImageView calendarIcon = findViewById(R.id.*imgCalendarIcon*);  
  
 // OnClickListener  
 calendarIcon.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 openCalendar();  
 }  
 });  
  
 // Check permission  
 checkCalendarPermission();  
 }  
  
 // Method to check permission  
 private void checkCalendarPermission() {  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_CALENDAR*) != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this, new String[]{Manifest.permission.*READ\_CALENDAR*}, *PERMISSIONS\_REQUEST\_READ\_CALENDAR*);  
 }  
 }  
  
 // Method to handle permission request result  
 @Override  
 public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {  
 if (requestCode == *PERMISSIONS\_REQUEST\_READ\_CALENDAR* && grantResults.length > 0 && grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) {  
 openCalendar();  
 } else {  
 Toast.*makeText*(this, "Calendar permission denied. Cannot set reminder.", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 // open the calendar app to add a reminder  
 private void openCalendar() {  
 Intent intent = new Intent(Intent.*ACTION\_INSERT*)  
 .setData(CalendarContract.Events.*CONTENT\_URI*)  
 .putExtra(CalendarContract.Events.*TITLE*, "Appointment Reminder")  
 .putExtra(CalendarContract.*EXTRA\_EVENT\_BEGIN\_TIME*, System.*currentTimeMillis*())  
 .putExtra(CalendarContract.*EXTRA\_EVENT\_END\_TIME*, System.*currentTimeMillis*() + 60 \* 60 \* 1000); // Event duration (1 hour)  
  
 startActivity(intent);  
 }  
}

**ServicesActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.ImageView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class ServicesActivity extends AppCompatActivity {  
  
 private ImageView groomCheckmark, nailCheckmark, bathCheckmark;  
 private boolean isGCSelected = false, isSCSelected = false, isDLSelected = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*services\_screen*);  
  
 // Initialize checkmark views  
 groomCheckmark = findViewById(R.id.*checkmark\_groom*);  
 nailCheckmark = findViewById(R.id.*checkmark\_nail*);  
 bathCheckmark = findViewById(R.id.*checkmark\_bath*);  
  
 // click listeners for the service cards  
 setupServiceSelection();  
  
 // Setup navigation buttons  
 setupNavigationButtons();  
 }  
  
 private void setupServiceSelection() {  
 findViewById(R.id.*groomCard*).setOnClickListener(v -> {  
 isGCSelected = !isGCSelected;  
 toggleCheckmark(groomCheckmark, isGCSelected);  
 });  
  
 findViewById(R.id.*nailClipCard*).setOnClickListener(v -> {  
 isSCSelected = !isSCSelected;  
 toggleCheckmark(nailCheckmark, isSCSelected);  
 });  
  
 findViewById(R.id.*bathCard*).setOnClickListener(v -> {  
 isDLSelected = !isDLSelected;  
 toggleCheckmark(bathCheckmark, isDLSelected);  
 });  
  
 findViewById(R.id.*button\_book\_appointment*).setOnClickListener(v -> {  
 Intent intent = new Intent(ServicesActivity.this, BookingActivity.class);  
 intent.putExtra("General Consultation", isGCSelected);  
 intent.putExtra("Specialist Consultation", isSCSelected);  
 intent.putExtra("Diagnostic and Lab Test Scheduling", isDLSelected);  
 startActivity(intent);  
 });  
 }  
  
 private void toggleCheckmark(ImageView checkmark, boolean isSelected) {  
 if (isSelected) {  
 checkmark.setVisibility(View.*VISIBLE*); // Show the checkmark on selection  
 } else {  
 checkmark.setVisibility(View.*GONE*); // Hide the checkmark on deselection  
 }  
 }  
  
 private void setupNavigationButtons() {  
 findViewById(R.id.*imgBack*).setOnClickListener(v -> finish());  
  
 findViewById(R.id.*imgHome*).setOnClickListener(v -> {  
 Intent homeIntent = new Intent(ServicesActivity.this, HomeActivity.class);  
 homeIntent.addFlags(Intent.*FLAG\_ACTIVITY\_CLEAR\_TOP* | Intent.*FLAG\_ACTIVITY\_NEW\_TASK*);  
 startActivity(homeIntent);  
 finish();  
 });  
 }  
}

**SignInActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.firebase.auth.FirebaseAuth;  
  
public class SignInActivity extends AppCompatActivity {  
  
 private EditText emailID;  
 private EditText passwordP;  
 private Button signInBtn;  
 private TextView signUp;  
 private TextView forgotPassword;  
 private FirebaseAuth pAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*login\_screen*);  
  
 // Initialize Firebase Auth  
 pAuth = FirebaseAuth.*getInstance*();  
  
  
 emailID = findViewById(R.id.*emailEditText*);  
 passwordP = findViewById(R.id.*loginPasswordEditText*);  
 signInBtn = findViewById(R.id.*loginButton*);  
 signUp = findViewById(R.id.*textViewSignUp*);  
 forgotPassword = findViewById(R.id.*forgotPasswordText*);  
  
 // click listeners  
 signInBtn.setOnClickListener(view -> attemptSignIn());  
 signUp.setOnClickListener(view -> gotoSignUp());  
 forgotPassword.setOnClickListener(view -> sendResetEmail());  
  
 if (getIntent().hasExtra("email")) {  
 String email = getIntent().getStringExtra("email");  
 emailID.setText(email);  
 }  
  
 }  
  
 // Add this onStart() method  
 @Override  
 protected void onStart() {  
 super.onStart();  
  
 // Check if user is signed in (non-null) and update UI accordingly.  
 if (pAuth.getCurrentUser() != null) {  
 gotoHome();  
 }  
 }  
  
 private void attemptSignIn() {  
 String email = emailID.getText().toString();  
 String password = passwordP.getText().toString();  
  
 if (validateEntries(email, password)) {  
 pAuth.signInWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, task -> {  
 if (task.isSuccessful()) {  
 gotoHome();  
 } else {  
 displayToast("Authentication failed: " + task.getException().getMessage());  
 }  
 });  
 }  
 }  
  
 private boolean validateEntries(String email, String password) {  
 if (email.isEmpty()) {  
 displayToast("Email cannot be empty.");  
 return false;  
 }  
 if (password.isEmpty()) {  
 displayToast("Password cannot be empty.");  
 return false;  
 }  
 return true;  
 }  
  
 private void gotoHome() {  
 Intent intent = new Intent(SignInActivity.this, HomeActivity.class);  
 startActivity(intent);  
 finish(); // Close the SignInActivity  
  
 }  
  
 private void gotoSignUp() {  
 Intent intent = new Intent(SignInActivity.this, SignUpActivity.class);  
 startActivity(intent);  
 }  
  
 private void sendResetEmail() {  
 String emailAddress = emailID.getText().toString();  
  
 if (!emailAddress.isEmpty()) {  
 pAuth.sendPasswordResetEmail(emailAddress)  
 .addOnCompleteListener(task -> {  
 if (task.isSuccessful()) {  
 displayToast("Password reset email sent.");  
 } else {  
 displayToast("Failed to send reset email. " + task.getException().getMessage());  
 }  
 });  
 } else {  
 displayToast("Please enter your email address to reset the password.");  
 }  
 }  
  
 private void displayToast(String message) {  
 Toast.*makeText*(SignInActivity.this, message, Toast.*LENGTH\_SHORT*).show();  
 }  
}

**SignUpActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.firebase.auth.FirebaseAuth;  
  
public class SignUpActivity extends AppCompatActivity {  
  
 private EditText emailID, passwordP, confirmPasswordP;  
 private Button signUpBtn;  
 private FirebaseAuth pAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*signup\_screen*);  
  
 ImageView backButton = findViewById(R.id.*imgBack*);  
 backButton.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 finish();  
 }  
 });  
  
 // Initialize Firebase Auth  
 pAuth = FirebaseAuth.*getInstance*();  
  
 emailID = findViewById(R.id.*emailEditText*);  
 passwordP = findViewById(R.id.*passwordEditText*);  
 confirmPasswordP = findViewById(R.id.*repeatPasswordEditText*);  
 signUpBtn = findViewById(R.id.*signUpButton*);  
  
 signUpBtn.setOnClickListener(view -> registerUser());  
 }  
  
  
 private void registerUser() {  
 String email = emailID.getText().toString().trim();  
 String password = passwordP.getText().toString().trim();  
 String confirmPassword = confirmPasswordP.getText().toString().trim();  
  
 // Input validation  
 if (email.isEmpty() || password.isEmpty()) {  
 Toast.*makeText*(this, "Email and password fields cannot be empty.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 if (!password.equals(confirmPassword)) {  
 Toast.*makeText*(this, "Passwords do not match.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 // Firebase registration  
 pAuth.createUserWithEmailAndPassword(email, password)  
 .addOnCompleteListener(this, task -> {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(SignUpActivity.this, "Registration successful", Toast.*LENGTH\_SHORT*).show();  
  
 // Navigate back to SignInActivity and pass the email  
 Intent intent = new Intent(SignUpActivity.this, SignInActivity.class);  
 intent.putExtra("email", email);  
 startActivity(intent);  
 finish();  
 } else {  
 Toast.*makeText*(SignUpActivity.this, "Registration failed: " + task.getException().getMessage(), Toast.*LENGTH\_LONG*).show();  
 }  
 });  
 }  
}

**SplashActivity.java**

package com.example.mabs;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
public class SplashActivity extends AppCompatActivity {  
  
 private static final int *SPLASH\_DISPLAY\_LENGTH* = 5000; //5 seconds(5000 ms)  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*splash\_screen*);  
  
 new Handler().postDelayed(() -> {  
 Intent intent = new Intent(SplashActivity.this, SignInActivity.class);  
 startActivity(intent);  
 finish();  
 }, *SPLASH\_DISPLAY\_LENGTH*);  
 }  
}

**TimeSlotAdapter.java**

package com.example.mabs;  
  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.TextView;  
  
import androidx.core.content.ContextCompat;  
import androidx.recyclerview.widget.RecyclerView;  
  
import java.util.ArrayList;  
  
  
public class TimeSlotAdapter extends RecyclerView.Adapter<TimeSlotAdapter.TimeSlotViewHolder> {  
 private ArrayList<String> timeSlots;  
 private int selectedItem = -1;  
 private OnTimeSlotClickListener timeSlotClickListener;  
  
 public interface OnTimeSlotClickListener {  
 void onTimeSlotClick(String timeSlot, int position);  
 }  
  
 public TimeSlotAdapter(ArrayList<String> timeSlots, OnTimeSlotClickListener listener) {  
 this.timeSlots = timeSlots;  
 this.timeSlotClickListener = listener;  
 }  
  
 @Override  
 public TimeSlotViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*timeslot\_item*, parent, false);  
 return new TimeSlotViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(TimeSlotViewHolder holder, int position) {  
 String timeSlot = timeSlots.get(position);  
 holder.timeSlotTextView.setText(timeSlot);  
  
 holder.itemView.setBackground(ContextCompat.*getDrawable*(holder.itemView.getContext(),  
 selectedItem == position ? R.drawable.*time\_slot\_background\_selected* : R.drawable.*time\_slot\_background*));  
  
 holder.itemView.setOnClickListener(v -> {  
 int previousItem = selectedItem;  
 selectedItem = position;  
 notifyItemChanged(previousItem);  
 notifyItemChanged(position);  
 if (timeSlotClickListener != null) {  
 timeSlotClickListener.onTimeSlotClick(timeSlot, position);  
 }  
 });  
 }  
  
 @Override  
 public int getItemCount() {  
 return timeSlots.size();  
 }  
  
 static class TimeSlotViewHolder extends RecyclerView.ViewHolder {  
 TextView timeSlotTextView;  
  
 TimeSlotViewHolder(View itemView) {  
 super(itemView);  
 timeSlotTextView = itemView.findViewById(R.id.*tvTimeSlot*);  
 }  
 }  
  
 public String getSelectedTimeSlot() {  
 return selectedItem >= 0 ? timeSlots.get(selectedItem) : null;  
 }  
}

**activity\_about.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/cardAbout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 app:cardElevation="4dp"  
 app:cardCornerRadius="8dp"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:layout\_marginTop="16dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:padding="16dp">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:gravity="center\_horizontal">  
  
  
 <com.google.android.material.imageview.ShapeableImageView  
 android:id="@+id/imagePapa"  
 android:layout\_width="96dp"  
 android:layout\_height="96dp"  
 android:src="@drawable/james\_papademas"  
 android:scaleType="centerCrop"  
 app:shapeAppearanceOverlay="@style/CircularShapeAppearance"  
 android:layout\_marginEnd="8dp" />  
  
  
 <com.google.android.material.imageview.ShapeableImageView  
 android:id="@+id/imageIcon"  
 android:layout\_width="96dp"  
 android:layout\_height="96dp"  
 android:src="@drawable/icon1"  
 android:scaleType="centerCrop"  
 app:shapeAppearanceOverlay="@style/CircularShapeAppearance"  
 android:layout\_marginEnd="8dp" />  
  
 </LinearLayout>  
  
 <!-- Paragraph 1 -->  
 <TextView  
 android:id="@+id/textViewParagraph1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:text="@string/about\_paragraph\_1"  
 android:textAlignment="center"  
 android:textSize="28sp" />  
  
 <!-- Paragraph 2 -->  
 <TextView  
 android:id="@+id/textViewParagraph2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:text="@string/about\_paragraph\_2"  
 android:textSize="14sp" />  
  
 <!-- Paragraph 3 -->  
 <TextView  
 android:id="@+id/textViewParagraph3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:text="@string/about\_paragraph\_3"  
 android:textSize="14sp" />  
  
  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/cardAbout"  
 android:layout\_marginTop="16dp"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**activity\_booking.xml**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.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">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="16dp">  
  
 <!-- Back -->  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/back"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="2dp" />  
  
 <!-- Home -->  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/home"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginTop="2dp" />  
  
  
 <TextView  
 android:id="@+id/labelServiceOptions"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Choose services:"  
 android:textStyle="bold"  
 android:layout\_marginTop="32dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <!-- Checkboxes -->  
 <CheckBox  
 android:id="@+id/checkBox1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="General Consultation"  
 android:layout\_below="@+id/labelServiceOptions"  
 android:layout\_marginTop="16dp"/>  
  
 <CheckBox  
 android:id="@+id/checkBox2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Specialist Consultation"  
 android:layout\_below="@id/checkBox1"  
 android:layout\_marginTop="8dp"/>  
  
 <CheckBox  
 android:id="@+id/checkBox3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Diagnostic and Lab Test Scheduling"  
 android:layout\_below="@id/checkBox2"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/labelCalView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Pick a date:"  
 android:textStyle="bold"  
 android:layout\_below="@id/checkBox3"  
 android:layout\_marginTop="16dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <CalendarView  
 android:id="@+id/calView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"  
 android:layout\_below="@id/labelCalView"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/labelTimeSlots"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Pick your time slot:"  
 android:textStyle="bold"  
 android:layout\_below="@id/calView"  
 android:layout\_marginTop="16dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/timeslotRecycler"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/labelTimeSlots"  
 android:layout\_marginTop="8dp"  
 tools:listitem="@layout/timeslot\_item"/>  
  
 <TextView  
 android:id="@+id/labelMap"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Select a location:"  
 android:textStyle="bold"  
 android:layout\_below="@id/timeslotRecycler"  
 android:layout\_marginTop="16dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <fragment  
 android:id="@+id/autocomplete\_fragment"  
 android:name="com.google.android.libraries.places.widget.AutocompleteSupportFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/labelMap"  
 android:background="@drawable/border"  
 android:padding="8dp"/>  
  
 <fragment  
 android:id="@+id/map"  
 android:name="com.google.android.gms.maps.SupportMapFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"  
 android:layout\_below="@id/autocomplete\_fragment"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/selectedLocationText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="No location selected"  
 android:padding="16dp"  
 android:layout\_below="@id/map"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/labelPrice"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Total Price: $0.00"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textStyle="bold"  
 android:layout\_below="@id/selectedLocationText"  
 android:layout\_marginTop="16dp"/>  
  
 <Button  
 android:id="@+id/confirmButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Confirm Booking"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:textColor="@android:color/white"  
 android:layout\_below="@id/labelPrice"  
 android:layout\_marginTop="16dp"/>  
 </RelativeLayout>  
</ScrollView>

**activity\_calendar.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
  
 <!-- Back -->  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 android:padding="8dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="16dp"/>  
  
  
 <!-- Home -->  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/home"  
 android:padding="8dp"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginTop="16dp"/>  
  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:layout\_below="@id/imgBack"  
 android:layout\_marginTop="48dp">  
  
 <ImageView  
 android:id="@+id/imgCalendarIcon"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:src="@drawable/remind"/>  
  
  
 <TextView  
 android:id="@+id/txtSetDogWalkReminder"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="24dp"  
 android:text="Upcoming Appointments"  
 android:textStyle="bold"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium" />  
  
 </LinearLayout>  
  
</RelativeLayout>

**activity\_confirmation.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/white"  
 android:padding="16dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 android:padding="16dp"/>  
  
 <View  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"/>  
  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/home"  
 android:padding="16dp"/>  
 </LinearLayout>  
  
  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 card\_view:cardCornerRadius="8dp"  
 card\_view:cardElevation="4dp"  
 android:layout\_margin="8dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/tvConfirmationMessage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:drawableStart="@drawable/checkcircle"  
 android:drawablePadding="8dp"  
 android:text="Scheduled"  
 android:gravity="start"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textStyle="bold"  
 android:layout\_marginBottom="8dp"/>  
  
 <TextView  
 android:id="@+id/tvAppointmentDetails"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Your appointment is confirmed and approved!"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:gravity="start"  
 android:layout\_marginBottom="16dp"/>  
  
 <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="@android:color/darker\_gray"  
 android:layout\_marginBottom="16dp"/>  
  
 <TextView  
 android:id="@+id/tvService"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:drawableStart="@drawable/event"  
 android:drawablePadding="8dp"  
 android:text="Details"  
 android:gravity="start"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 android:layout\_marginBottom="8dp"/>  
  
 <TextView  
 android:id="@+id/tvCalendar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="10:00am - 11:00am February 15, 2023"  
 android:gravity="start"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:layout\_marginBottom="8dp"/>  
  
 <TextView  
 android:id="@+id/tvLocation"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:drawableStart="@drawable/conic"  
 android:drawablePadding="8dp"  
 android:text="Fido's Grooming with Groomer Name at PawPerfection Salon"  
 android:gravity="start"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:layout\_marginBottom="24dp"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:layout\_marginTop="24dp">  
  
 <Button  
 android:id="@+id/btnAddToCalendar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="ADD TO SCHEDULE"  
 android:backgroundTint="@color/colorPrimary"  
 android:textColor="@android:color/white"  
 android:layout\_marginBottom="16dp"  
 android:minHeight="48dp"  
 android:elevation="2dp"/>  
  
 <Button  
 android:id="@+id/btnCancel"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="CANCEL"  
 android:backgroundTint="@color/colorPrimary"  
 android:textColor="@android:color/white"  
 android:minHeight="48dp"  
 android:elevation="2dp"/>  
 </LinearLayout>  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
</LinearLayout>

**activity\_inpatient.xml**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="16dp">  
  
  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/back"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="2dp"/>  
  
  
 <!-- Home-->  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/home"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginTop="2dp"/>  
  
  
 <TextView  
 android:id="@+id/tvDaycareService"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="InPatient Room Reservation"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textStyle="bold"  
 android:layout\_below="@+id/imgHome"  
 android:layout\_marginTop="24dp"  
 android:layout\_marginBottom="8dp"/>  
  
 <!-- price -->  
 <TextView  
 android:id="@+id/priceTextView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Price will be calculated here"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:layout\_below="@id/tvDaycareService"  
 android:layout\_marginTop="8dp"/>  
  
 <!-- duration buttons -->  
 <LinearLayout  
 android:id="@+id/hoursLinearLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/priceTextView"  
 android:orientation="horizontal"  
 android:layout\_marginTop="8dp">  
  
 <Button  
 android:id="@+id/fourHoursButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:layout\_weight="1"  
 android:text="4 Hours"/>  
  
 <Button  
 android:id="@+id/eightHoursButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:layout\_weight="1"  
 android:text="8 Hours"/>  
  
 <Button  
 android:id="@+id/twelveHoursButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:layout\_weight="1"  
 android:text="12 Hours"/>  
  
 <Button  
 android:id="@+id/twentyFourHoursButton"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:focusable="true"  
 android:layout\_weight="1"  
 android:text="24 Hours"/>  
 </LinearLayout>  
  
 <!-- CalendarView -->  
 <CalendarView  
 android:id="@+id/calendarView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"  
 android:layout\_below="@id/hoursLinearLayout"  
 android:layout\_marginTop="8dp"/>  
  
 <!-- Autocomplete fragment -->  
 <fragment  
 android:id="@+id/autocomplete\_fragment"  
 android:name="com.google.android.libraries.places.widget.AutocompleteSupportFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/calendarView"  
 android:background="@drawable/border"  
 android:padding="8dp"/>  
  
 <!-- Map fragment -->  
 <fragment  
 android:id="@+id/map"  
 android:name="com.google.android.gms.maps.SupportMapFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"  
 android:layout\_below="@id/autocomplete\_fragment"  
 android:layout\_marginTop="8dp"/>  
  
  
 <TextView  
 android:id="@+id/selectedLocationText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/map"  
 android:text="No location selected"  
 android:padding="16dp"  
 android:layout\_marginTop="8dp"/>  
  
 <!-- confirm booking -->  
 <Button  
 android:id="@+id/confirmButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/selectedLocationText"  
 android:text="Confirm Booking"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:textColor="@android:color/white"  
 android:layout\_marginTop="16dp"/>  
  
  
 </RelativeLayout>  
</ScrollView>

**activity\_ipconfirm.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/white"  
 android:padding="16dp">  
  
 <RelativeLayout  
 android:id="@+id/topBarRelativeLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="8dp"  
 android:paddingBottom="8dp">  
  
 <!-- Back -->  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="16dp"/>  
  
  
 <!-- Home -->  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/home"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="16dp"  
 android:contentDescription="@string/home" />  
 </RelativeLayout>  
  
  
  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 card\_view:cardCornerRadius="8dp"  
 card\_view:cardElevation="4dp"  
 android:layout\_margin="8dp"  
 android:layout\_marginTop="48dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/tvConfirmationMessage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:drawableStart="@drawable/checkcircle"  
 android:drawablePadding="8dp"  
 android:text="InPatient Room Booking Confirmed"  
 android:gravity="start"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textStyle="bold"  
 android:layout\_marginBottom="8dp"/>  
  
 <TextView  
 android:id="@+id/tvPrice"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Total cost will be displayed here."  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:gravity="start"  
 android:layout\_marginBottom="16dp"/>  
  
 <TextView  
 android:id="@+id/tvLocation"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Location details will be displayed here."  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:gravity="start"  
 android:layout\_marginBottom="16dp"/>  
  
 <View  
 android:layout\_width="match\_parent"  
 android:layout\_height="1dp"  
 android:background="@android:color/darker\_gray"  
 android:layout\_marginBottom="16dp"/>  
  
 <TextView  
 android:id="@+id/tvHours"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Hours selected will be displayed here."  
 android:gravity="start"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:layout\_marginBottom="24dp"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"  
 android:layout\_marginTop="24dp">  
  
 <Button  
 android:id="@+id/btnAddToCalendar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="ADD TO SCHEDULE"  
 android:backgroundTint="@color/colorPrimary"  
 android:textColor="@android:color/white"  
 android:layout\_marginBottom="16dp"  
 android:minHeight="48dp"  
 android:elevation="2dp"/>  
  
 <Button  
 android:id="@+id/btnCancel"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="CANCEL"  
 android:backgroundTint="@color/colorPrimary"  
 android:textColor="@android:color/white"  
 android:minHeight="48dp"  
 android:elevation="2dp"/>  
 </LinearLayout>  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
</LinearLayout>

**activity\_profile.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:card\_view="http://schemas.android.com/apk/res-auto"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/white"  
 android:padding="16dp">  
  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/home"  
 android:contentDescription="@string/home"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="16dp" />  
  
 <ImageView  
 android:id="@+id/imgLogout"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/logout"  
 android:contentDescription="@string/logout"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginTop="16dp" />  
  
 <TextView  
 android:id="@+id/tvProfileHeader"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/my\_profile"  
 android:textAppearance="@android:style/TextAppearance.Large"  
 android:textStyle="bold"  
 android:layout\_below="@id/imgHome"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="16dp" />  
  
 <TextView  
 android:id="@+id/tvAccountSettings"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/account\_settings"  
 android:textAppearance="@android:style/TextAppearance.Medium"  
 android:textStyle="bold"  
 android:layout\_below="@id/tvProfileHeader"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="8dp" />  
  
 <androidx.cardview.widget.CardView  
 android:id="@+id/cardViewProfile"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tvAccountSettings"  
 card\_view:cardCornerRadius="8dp"  
 card\_view:cardElevation="4dp"  
 android:layout\_marginTop="16dp">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="16dp">  
  
 <de.hdodenhof.circleimageview.CircleImageView  
 android:id="@+id/imgProfilePicture"  
 android:layout\_width="96dp"  
 android:layout\_height="96dp"  
 android:src="@drawable/ic\_person"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="4dp"  
 app:civ\_border\_width="2dp"  
 app:civ\_border\_color="@android:color/black"/>  
  
 <EditText  
 android:id="@+id/editPatientName"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/imgProfilePicture"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="16dp"  
 android:hint="@string/patient\_name"  
 android:inputType="textPersonName" />  
  
 <EditText  
 android:id="@+id/editMedicalRecordNumber"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editPatientName"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/medical\_record\_number"  
 android:inputType="text" />  
  
 <EditText  
 android:id="@+id/editAge"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editMedicalRecordNumber"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/age"  
 android:inputType="number" />  
  
 <EditText  
 android:id="@+id/editPrimaryPhysician"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editAge"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/primary\_physician"  
 android:inputType="textPersonName" />  
  
 <EditText  
 android:id="@+id/editInsuranceProvider"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editPrimaryPhysician"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/insurance\_provider"  
 android:inputType="text" />  
  
 <EditText  
 android:id="@+id/editAllergies"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editInsuranceProvider"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/allergies"  
 android:inputType="text" />  
  
 <EditText  
 android:id="@+id/editSpecialNotes"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/editAllergies"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="8dp"  
 android:hint="@string/special\_notes"  
 android:inputType="text" />  
  
 <Button  
 android:id="@+id/btnEdit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/edit"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:textColor="@android:color/white"  
 android:layout\_below="@id/editSpecialNotes"  
 android:layout\_marginTop="16dp" />  
  
 <Button  
 android:id="@+id/btnConfirm"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/confirm"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:textColor="@android:color/white"  
 android:layout\_below="@id/btnEdit"  
 android:layout\_marginTop="8dp" />  
 </RelativeLayout>  
 </androidx.cardview.widget.CardView>  
</RelativeLayout>

**activity\_quickcare.xml**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <RelativeLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="16dp">  
  
 <!-- Back -->  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/back"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="2dp"/>  
  
  
 <!-- Home -->  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="24dp"  
 android:layout\_height="24dp"  
 android:src="@drawable/home"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginTop="2dp" />  
  
 <TextView  
 android:id="@+id/tvVetCareTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="48dp"  
 android:text="Quick Care"  
 android:textStyle="bold"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <LinearLayout  
 android:id="@+id/checkGroupType"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/tvVetCareTitle"  
 android:orientation="vertical"  
 android:layout\_marginTop="8dp">  
  
 <CheckBox  
 android:id="@+id/checkBoxOP"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Prescription Refill"/>  
  
 <CheckBox  
 android:id="@+id/checkBoxSurgery"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Minor Ailments"/>  
 </LinearLayout>  
  
 <TextView  
 android:id="@+id/labelCalendarView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/checkGroupType"  
 android:text="Pick a date:"  
 android:textStyle="bold"  
 android:layout\_marginTop="16dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <CalendarView  
 android:id="@+id/calendarView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"  
 android:layout\_below="@id/labelCalendarView"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/labelTimeSlots"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/calendarView"  
 android:text="Select a time slot:"  
 android:textStyle="bold"  
 android:layout\_marginTop="16dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <!-- RecyclerView -->  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/timeslotRecyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/labelTimeSlots"  
 android:layout\_marginTop="8dp"  
 android:orientation="horizontal"  
 tools:listitem="@layout/timeslot\_item"/>  
  
 <TextView  
 android:id="@+id/labelMap"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/timeslotRecyclerView"  
 android:text="Select a location:"  
 android:textStyle="bold"  
 android:layout\_marginTop="16dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"/>  
  
 <!-- Autocomplete Fragment -->  
 <fragment  
 android:id="@+id/autocomplete\_fragment"  
 android:name="com.google.android.libraries.places.widget.AutocompleteSupportFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/labelMap"  
 android:background="@drawable/border"  
 android:padding="8dp"/>  
  
 <!-- Map Fragment -->  
 <fragment  
 android:id="@+id/map"  
 android:name="com.google.android.gms.maps.SupportMapFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="300dp"  
 android:layout\_below="@id/autocomplete\_fragment"  
 android:layout\_marginTop="8dp"/>  
  
 <TextView  
 android:id="@+id/selectedLocationText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/map"  
 android:text="No location selected"  
 android:padding="16dp"  
 android:layout\_marginTop="8dp"/>  
  
 <Button  
 android:id="@+id/confirmButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/selectedLocationText"  
 android:text="Confirm Appointment"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:textColor="@android:color/white"  
 android:layout\_marginTop="16dp"/>  
 </RelativeLayout>  
</ScrollView>

**item\_event.xml**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="8dp">  
  
  
 <TextView  
 android:id="@+id/tvEventTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="16sp"  
 android:textStyle="bold"  
 android:textColor="@android:color/black" />  
  
  
 <TextView  
 android:id="@+id/tvEventTimeRange"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textSize="14sp"  
 android:textColor="@android:color/black" />  
  
</LinearLayout>

**item\_view\_pager.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scaleType="centerCrop" />  
</FrameLayout>

**login\_screen.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
  
 <ImageView  
 android:id="@+id/loginLogo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/circic"  
 android:maxHeight="200dp"  
 android:maxWidth="200dp"  
 android:adjustViewBounds="true"  
 android:scaleType="centerInside"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="48dp"/>  
  
 <EditText  
 android:id="@+id/emailEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/loginLogo"  
 android:layout\_marginTop="32dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:hint="Email"  
 android:drawableStart="@drawable/ic\_email"  
 android:drawablePadding="10dp"  
 android:inputType="textEmailAddress"/>  
  
 <EditText  
 android:id="@+id/loginPasswordEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/emailEditText"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:layout\_marginTop="16dp"  
 android:hint="Password"  
 android:drawableStart="@drawable/ic\_password"  
 android:drawablePadding="10dp"  
 android:inputType="textPassword"/>  
  
 <Button  
 android:id="@+id/loginButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/loginPasswordEditText"  
 android:layout\_marginTop="24dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:text="SIGN IN"  
 android:textColor="@color/black"  
 android:textAllCaps="false"/>  
  
 <TextView  
 android:id="@+id/forgotPasswordText"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/loginButton"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="16dp"  
 android:text="Forgot your password?"  
 android:textColor="@color/black"  
 android:clickable="true"  
 android:focusable="true"  
 android:textSize="14sp"/>  
  
 <TextView  
 android:id="@+id/textViewSignUp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/forgotPasswordText"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="16dp"  
 android:text="New here? Sign up"  
 android:textColor="@color/black"  
 android:clickable="true"  
 android:focusable="true"  
 android:textSize="14sp"/>  
  
  
</RelativeLayout>

**services\_screen.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <!-- Back -->  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 android:padding="16dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentStart="true"/>  
  
 <!-- Home -->  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/home"  
 android:padding="16dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"/>  
  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_below="@+id/imgHome"  
 android:fillViewport="true">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <!-- Grooming Service -->  
 <androidx.cardview.widget.CardView  
 android:id="@+id/groomCard"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 app:cardElevation="4dp"  
 app:cardCornerRadius="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="toggleCheckmark"  
 app:cardBackgroundColor="@android:color/white"  
 android:foreground="?android:attr/selectableItemBackground">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@android:color/transparent">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:src="@drawable/gencon"  
 android:scaleType="centerCrop" />  
  
 <ImageView  
 android:id="@+id/checkmark\_groom"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/selected"  
 android:visibility="gone"  
 android:layout\_centerInParent="true" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="General Consultation"  
 android:padding="8dp"  
 android:background="@android:color/transparent" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="$25"  
 android:padding="8dp"  
 android:paddingBottom="16dp"  
 android:background="@android:color/transparent" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
  
 <!-- Nail Trimming -->  
 <androidx.cardview.widget.CardView  
 android:id="@+id/nailClipCard"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 app:cardElevation="4dp"  
 app:cardCornerRadius="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="toggleCheckmark"  
 app:cardBackgroundColor="@android:color/white"  
 android:foreground="?android:attr/selectableItemBackground">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@android:color/transparent">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:src="@drawable/spcon"  
 android:scaleType="centerCrop" />  
  
 <ImageView  
 android:id="@+id/checkmark\_nail"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/selected"  
 android:visibility="gone"  
 android:layout\_centerInParent="true" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Specialist Consultation"  
 android:padding="8dp"  
 android:background="@android:color/transparent" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="$50"  
 android:padding="8dp"  
 android:paddingBottom="16dp"  
 android:background="@android:color/transparent" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
  
 <!-- Hygiene and Bath -->  
 <androidx.cardview.widget.CardView  
 android:id="@+id/bathCard"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 app:cardElevation="4dp"  
 app:cardCornerRadius="8dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:onClick="toggleCheckmark"  
 app:cardBackgroundColor="@android:color/white"  
 android:foreground="?android:attr/selectableItemBackground">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:background="@android:color/transparent">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:src="@drawable/labcon"  
 android:scaleType="centerCrop" />  
  
 <ImageView  
 android:id="@+id/checkmark\_bath"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/selected"  
 android:visibility="gone"  
 android:layout\_centerInParent="true" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Diagnostic and Lab Test Scheduling"  
 android:padding="8dp"  
 android:background="@android:color/transparent" />  
  
 <TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="$30"  
 android:padding="8dp"  
 android:paddingBottom="16dp"  
 android:background="@android:color/transparent" />  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
  
  
 <!-- Schedule -->  
 <Button  
 android:id="@+id/button\_book\_appointment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:text="Book Appointment"  
 android:textColor="@android:color/white" />  
  
 </LinearLayout>  
 </ScrollView>  
</RelativeLayout>

**signup\_screen.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="16dp">  
  
 <!-- Back -->  
 <ImageView  
 android:id="@+id/imgBack"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/back"  
 android:padding="8dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_marginTop="16dp"/>  
  
  
  
 <ImageView  
 android:id="@+id/signupLogo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/circic"  
 android:maxHeight="200dp"  
 android:maxWidth="200dp"  
 android:adjustViewBounds="true"  
 android:scaleType="centerInside"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="48dp"/>  
  
 <EditText  
 android:id="@+id/emailEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/signupLogo"  
 android:layout\_marginTop="32dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:drawableStart="@drawable/ic\_email"  
 android:drawablePadding="10dp"  
 android:hint="Email"  
 android:inputType="text"/>  
  
 <EditText  
 android:id="@+id/passwordEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/emailEditText"  
 android:layout\_marginTop="16dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:drawableStart="@drawable/ic\_password"  
 android:drawablePadding="10dp"  
 android:hint="Password"  
 android:inputType="textPassword"/>  
  
 <EditText  
 android:id="@+id/repeatPasswordEditText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/passwordEditText"  
 android:layout\_marginTop="16dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:drawableStart="@drawable/ic\_confirm\_password"  
 android:drawablePadding="10dp"  
 android:hint="Repeat password"  
 android:inputType="textPassword"/>  
  
 <Button  
 android:id="@+id/signUpButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/repeatPasswordEditText"  
 android:layout\_marginTop="24dp"  
 android:background="@drawable/edit\_text\_capsule\_shape"  
 android:text="SIGN UP"  
 android:textColor="@color/black"  
 android:textAllCaps="false"/>  
  
  
</RelativeLayout>

**splash\_screen.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/textViewBanner"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="MABS"  
 android:textAlignment="center"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="50dp"  
 android:textSize="48sp"  
 android:textStyle="bold"  
 android:textColor="@android:color/black" />  
  
 <ImageView  
 android:id="@+id/imageViewLogo"  
 android:layout\_width="400dp"  
 android:layout\_height="400dp"  
 android:src="@drawable/icon1"  
 android:layout\_below="@id/textViewBanner"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="20dp" />  
  
  
</RelativeLayout>

**timeslot\_item.xml**

<?xml version="1.0" encoding="utf-8"?>  
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:background="@drawable/time\_slot\_background"  
 android:clickable="true"  
 android:focusable="true">  
  
 <TextView  
 android:id="@+id/tvTimeSlot"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="09:00"  
 android:gravity="center"  
 android:textSize="18sp" />  
  
</FrameLayout>

**strings.xml**

<resources>  
 <string name="app\_name">MABS</string>  
 <string name="google\_maps\_key">AIzaSyCNRn9Cm3zz1yhBNBgI9kLNDDLVWuHU7qw</string>  
 <string name="home">Home</string>  
 <string name="logout">Logout</string>  
 <string name="my\_profile">My Profile</string>  
 <string name="account\_settings">Account Settings</string>  
 <string name="age">Age</string>  
 <string name="allergies">Allergies</string>  
 <string name="special\_notes">Special Notes</string>  
 <string name="confirm">Confirm</string>  
 <string name="edit">Edit</string>  
 <string name="add\_photo">Add Photo</string>  
 <string name="colorPrimary">#6200EE</string> <!-- Replace with actual color value from colors.xml -->  
 <string name="white">#FFFFFF</string> <!-- Replace with actual color value from colors.xml -->  
 <string name="select\_image">Select Image</string>  
 <string name="about\_paragraph\_1">MABS</string>  
 <string name="about\_paragraph\_2">Copyright © 2025 Designed by Padmavathi Vyshnavi Madarampalli and Venkata Sai Gunisetty.</string>  
 <string name="about\_paragraph\_3">We are Padmavathi Vyshnavi Madarampalli and Venkata Sai Gunisetty, and we built MABS as a straightforward way to schedule medical appointments. In the app, you pick a doctor, choose a date and time from preset slots, and enter your details. MABS keeps track of your upcoming visits and can send simple reminders so you don’t forget.We created MABS to make setting up a doctor’s visit as easy as possible.  
</string>  
 <string name="default\_web\_client\_id">992300470029-jgi1ni66bgesnguj2oc5pu3qd18fptoa.apps.googleusercontent.com</string>  
 <string name="total\_price\_format">Total Price: $%d</string>  
 <string name="patient\_name">Patient Name</string>  
 <string name="medical\_record\_number">Medical Record Number</string>  
 <string name="primary\_physician">Primary Physician</string>  
 <string name="insurance\_provider">Insurance Provider</string>  
  
</resources>

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MABS"  
 tools:targetApi="31">  
 <meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="AIzaSyCNRn9Cm3zz1yhBNBgI9kLNDDLVWuHU7qw"/>  
  
  
 <activity  
 android:name=".SplashActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 <activity  
 android:name=".SignInActivity"  
 android:exported="true">  
 </activity>  
 <activity  
 android:name=".SignUpActivity"  
 android:exported="true">  
 </activity>  
 <activity android:name=".ServicesActivity"  
 android:exported="true">  
 </activity>  
 <activity android:name=".BookingActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".ConfirmationActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".HomeActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".ReminderActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".QuickCareActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".InpatientActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".InpatientConfirmationActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".AboutActivity"  
 android:exported="true">  
 </activity>  
  
 <activity android:name=".ProfileActivity"  
 android:exported="true">  
 </activity>  
  
 </application>  
  
</manifest>