

[2CEIT5PE18: MOBILE APPLICATION DEVELOPMENT]

Assignment

AIM- Smart Tourist Safety Monitoring & Incident Response System

Submitted By: RAJBHAR PRATHAMKUMAR JAGDISHBHAI
Enrollment number: 24172012068



Department of Computer
Engineering/Information Technology

Smart Tourist Safety Monitoring & Incident Response

MainActivity.kt

```
package com.bytemap.a24172012068_assignment

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.padding
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.Contacts
import androidx.compose.material.icons.filled.Description
import androidx.compose.material.icons.filled.LocationOn
import androidx.compose.material.icons.filled.MonitorHeart
import androidx.compose.material.icons.filled.Notifications
import androidx.compose.material.icons.filled.Settings
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.runtime.rememberCoroutineScope
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.currentBackStackEntryAsState
import androidx.navigation.compose.rememberNavController
import com.bytemap.a24172012068_assignment.components.AppBar
import com.bytemap.a24172012068_assignment.components.BottomBar
import com.bytemap.a24172012068_assignment.components.NavDrawer
import com.bytemap.a24172012068_assignment.components.NavMenuItem
import com.bytemap.a24172012068_assignment.screen.AlertsScreen
import com.bytemap.a24172012068_assignment.screen.Dashboard
import com.bytemap.a24172012068_assignment.screen.DangerZoneScreen
import com.bytemap.a24172012068_assignment.screen.EFirScreen
import com.bytemap.a24172012068_assignment.screen.EmergencyContactsScreen
import com.bytemap.a24172012068_assignment.screen.EmergencySOSScreen
import com.bytemap.a24172012068_assignment.screen.LocationHistoryScreen
import com.bytemap.a24172012068_assignment.screen.SOSSuccessScreen
import com.bytemap.a24172012068_assignment.screen.MapScreen
import com.bytemap.a24172012068_assignment.screen.NotificationsScreen
import com.bytemap.a24172012068_assignment.screen.ProfileScreen
import com.bytemap.a24172012068_assignment.screen.SafetyTipsScreen
import com.bytemap.a24172012068_assignment.screen.ServicesScreen
import com.bytemap.a24172012068_assignment.screen.SettingsScreen
import com.bytemap.a24172012068_assignment.screen.TripMonitorScreen
import kotlinx.coroutines.launch

class MainActivity : ComponentActivity() {
```

Smart Tourist Safety Monitoring & Incident Response

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContent {
        MaterialTheme {
            MainApp()
        }
    }
}

@Composable
fun MainApp() {
    val navController = rememberNavController()
    val drawerState = rememberDrawerState(DrawerValue.Closed)
    val scope = rememberCoroutineScope()

    val userName = "PRATHAM RAJBHAR"
    val userId = "46a08ef58a11b7f86bcb9a8d97c4ac95"

    val menuItems = listOf(
        NavMenuItem("Notifications", "notifications",
Icons.Default.Notifications, false),
        NavMenuItem("File E-FIR", "efir", Icons.Default.Description,
false),
        NavMenuItem("Trip Monitor", "trip_monitor",
Icons.Default.MonitorHeart, false),
        NavMenuItem("Location History", "location_history",
Icons.Default.LocationOn, false),
        NavMenuItem("Emergency Contacts", "emergency_contacts",
Icons.Default.Contacts, false),
        NavMenuItem("Settings", "settings", Icons.Default.Settings, false)
    )

    ModalNavigationDrawer(
        drawerState = drawerState,
        drawerContent = {
            NavDrawer(
                userName = userName,
                userId = userId,
                menuItems = menuItems.map {
                    it.copy(selected =
navController.currentBackStackEntryAsState().value?.destination?.route ==
it.route)
                },
                onItemClick = { route ->

```

Smart Tourist Safety Monitoring & Incident Response

```
        navController.navigate(route) {
            popUpTo(navController.graph.startDestinationId)
            saveState = true
        }
        launchSingleTop = true
        restoreState = true
    }
    scope.launch { drawerState.close() }
},
onLogout = {
}
)
)
}
)
{
    val currentBackStack
        navController.currentBackStackEntryAsState()
        val currentRoute = currentBackStack.value?.destination?.route ?:
    "dashboard"

    val bottomBarTitles = mapOf(
        "dashboard" to "Home",
        "map" to "Map",
        "alerts" to "Alerts",
        "profile" to "Profile"
    )
}

val currentTitle = bottomBarTitles[currentRoute]
?: menuItems.find { it.route == currentRoute }?.label
?: "App"
val canPop = navController.previousBackStackEntry != null

Scaffold(
    topBar = {
        AppBar(
            title = currentTitle,
            canPop = canPop,
            onNavIconClick = {
                if (canPop) {
                    navController.popBackStack()
                } else {
                    scope.launch { drawerState.open() }
                }
            }
        )
    },
    bottomBar = {
        BottomBar(navController = navController)
    }
)
```

Smart Tourist Safety Monitoring & Incident Response

```
        }
    ) { innerPadding ->
    NavHost(
        navController = navController,
        startDestination = "dashboard",
        modifier = Modifier.padding(innerPadding)
    ) {
        composable("dashboard") { Dashboard(navController) }
        composable("map") { MapScreen() }
        composable("alerts") { AlertsScreen() }
        composable("notifications")
    NotificationsScreen(navController) }
        composable("profile") { ProfileScreen() }
        composable("efir") { EFirScreen(navController) }
        composable("trip_monitor")
    TripMonitorScreen(navController) }
        composable("location_history")
    LocationHistoryScreen(navController) }
        composable("emergency_contacts")
    EmergencyContactsScreen(navController) }
        composable("settings") { SettingsScreen(navController) }
        composable("emergency_sos")
    EmergencySOSScreen(navController) }
        composable("sos_sent") { SOSSuccessScreen(navController) }
        composable("safety_tips")
    SafetyTipsScreen(navController) }
        composable("services") { ServicesScreen(navController) }
        composable("danger_zone")
    DangerZoneScreen(navController) }
    }
}
}
```

ActivitiesScreen.kt

```
package com.bytemap.a24172012068_assignment.screen
```

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.Divider
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

data class TripActivity(
    val time: String,
    val step: String,
    val coordinates: String,
)

@Composable
fun ActivitiesScreen(
    activities: List<TripActivity> = sampleActivities
) {
    Box(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFFFF5F7FA))
    ) {
        LazyColumn(
            contentPadding = PaddingValues(16.dp),
            verticalArrangement = Arrangement.spacedBy(0.dp)
        ) {
            items(activities) { activity ->
                ActivityLogLineModern(activity)
                Divider(
                    modifier     = Modifier.padding(vertical     = 6.dp,
horizontal = 6.dp),
                    color = Color(0xFFE0E4EA),
                    thickness = 1.dp
                )
            }
        }
    }
}

@Composable
fun ActivityLogLineModern(activity: TripActivity) {
    Row(
        modifier = Modifier
            .fillWidth()
            .padding(vertical = 6.dp),

```

Smart Tourist Safety Monitoring & Incident Response

```
        verticalAlignment = Alignment.CenterVertically
    ) {
    Box(
        modifier = Modifier
            .clip(RoundedCornerShape(16.dp))
            .background(Color(0xFF149EE7).copy(alpha = 0.08f))
            .padding(horizontal = 12.dp, vertical = 4.dp)
    ) {
        Text(
            text = activity.time,
            color = Color(0xFF149EE7),
            fontSize = 14.sp,
            fontWeight = FontWeight.Medium
        )
    }
    Spacer(Modifier.width(14.dp))
    Column {
        Text(
            text = activity.step,
            style = MaterialTheme.typography.bodyLarge.copy(fontWeight
= FontWeight.Bold)
        )
        Text(
            text = activity.coordinates,
            style = MaterialTheme.typography.bodySmall.copy(color =
Color(0xFF9197A6))
        )
    }
}
}

val sampleActivities = listOf(
    TripActivity("08:30 AM", "Trip Started", "23.474803, 72.391667"),
    TripActivity("08:45 AM", "Stopped at Gas Station", "23.480000,
72.393000"),
    TripActivity("09:00 AM", "Reached Shanku's Water Park", "23.474803,
72.391667"),
    TripActivity("09:30 AM", "Rest Stop", "23.501234, 72.398888"),
)

```

```
@Preview(showBackground = true)
@Composable
fun ActivitiesScreenPreview() {
    ActivitiesScreen()
}
```

AlertsScreen.kt

Smart Tourist Safety Monitoring & Incident Response

```
package com.bytemap.a24172012068_assignment.screen

import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.Campaign
import androidx.compose.material.icons.filled.HealthAndSafety
import androidx.compose.material.icons.filled.LocalFireDepartment
import androidx.compose.material.icons.filled.Traffic
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.Divider
import androidx.compose.material3.Icon
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

@Composable
fun AlertCardUI(
    icon: String,
    title: String,
    description: String,
    date: String,
    alertType: String
) {
    val imageVector = when (icon) {
        "local_fire_department" -> Icons.Default.LocalFireDepartment
        "health_and_safety" -> Icons.Default.HealthAndSafety
        "traffic" -> Icons.Default.Traffic
        "weather_mix" -> Icons.Default.Campaign
        else -> Icons.Default.Campaign
    }

    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(vertical = 4.dp, horizontal = 0.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 1.dp)
    )
}
```

Smart Tourist Safety Monitoring & Incident Response

```
) {
    Column(
        modifier = Modifier
            .padding(16.dp)
    ) {
        Row(verticalAlignment =
            androidx.compose.ui.Alignment.CenterVertically) {
            Icon(
                imageVector = imageVector,
                contentDescription = "Alert Icon",
                tint = Color(0xFF1976D2),
                modifier = Modifier.size(24.dp)
            )
            Spacer(modifier = Modifier.width(12.dp))
            Text(
                text = title,
                fontSize = 18.sp,
                fontWeight = FontWeight.Medium,
                color = Color(0xFF22223B),
                maxLines = 1
            )
        }
        Divider(modifier = Modifier.padding(vertical = 10.dp), color =
            Color(0xFFFFFFFF), thickness = 1.dp)
        Text(
            text = description,
            fontSize = 15.sp,
            color = Color(0xFF505050),
            modifier = Modifier.padding(bottom = 8.dp)
        )
    }
    Row {
        Text(
            text = date,
            fontSize = 12.sp,
            color = Color(0xFF909090),
            fontWeight = FontWeight.Normal
        )
        Spacer(modifier = Modifier.width(12.dp))
        Text(
            text = alertType,
            fontSize = 12.sp,
            color = Color(0xFF1976D2),
            fontWeight = FontWeight.SemiBold
        )
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
    }

}

data class Alert(
    val icon: String,
    val title: String,
    val description: String,
    val date: String,
    val alertType: String
)

@Composable
fun AlertsScreen() {
    val alerts = listOf(
        Alert("weather_mix", "Heavy Snowfall Warning", "Heavy snowfall expected in your area. Stay indoors and avoid travel if possible.", "2024-12-01", "Weather"),
        Alert("local_fire_department", "Wildfire Alert", "A wildfire has been reported near your location. Follow evacuation orders and stay updated.", "2024-11-28", "Fire"),
        Alert("health_and_safety", "Health Advisory", "An outbreak of flu has been reported in your community. Take necessary precautions and get vaccinated.", "2024-11-25", "Health"),
        Alert("traffic", "Traffic Accident Alert", "A major traffic accident has occurred on the highway near you. Expect delays and consider alternate routes.", "2024-11-20", "Traffic")
    )

    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFF6F7F9))
            .padding(vertical = 20.dp, horizontal = 16.dp)
    ) {
        LazyColumn(
            verticalArrangement = Arrangement.spacedBy(8.dp)
        ) {
            items(alerts) { alert ->
                AlertCardUI(
                    alert.icon,
                    alert.title,
                    alert.description,
                    alert.date,
                    alert.alertType
                )
            }
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        }
    }
}

@Preview(showBackground = true)
@Composable
fun AlertsScreenPreview() {
    AlertsScreen()
}
```

Dashboard.kt

```
package com.bytemap.a24172012068_assignment.screen
```

```
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.LocationOn
import androidx.compose.material.icons.filled.Person
import androidx.compose.material.icons.filled.Place
import androidx.compose.material.icons.filled.Warning
import androidx.compose.material.icons.outlined.MyLocation
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.Icon
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Brush
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.vector.ImageVector
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.text.style.TextOverflow
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController

data class QuickAction(
    val title: String,
    val subtitle: String,
    val bgColor: Color,
    val icon: ImageVector
)

@Composable
fun Dashboard(navController: NavController) {
    val quickActions = listOf(
        QuickAction("Safety Tips", "Stay protected", Color(0xFF10B981),
Icons.Default.Person),
        QuickAction("E-FIR", "File report", Color(0xFF3B82F6),
Icons.Default.Warning),
        QuickAction("Services", "Emergency help", Color(0xFFFF59E0B),
Icons.Default.Place),
        QuickAction("Danger Zone", "High-risk areas", Color(0xFFEF4444),
Icons.Default.LocationOn)
    )

    Box(
        modifier = Modifier
            .fillMaxSize()
            .background(
                Brush.verticalGradient(
                    colors = listOf(
                        Color(0xFFF8FAFC),
                        Color(0xFFFF6FF)
                    )
                )
            )
    ) {
        Column(
            modifier = Modifier
                .fillMaxSize()

```

Smart Tourist Safety Monitoring & Incident Response

```
.verticalScroll(rememberScrollState())
    .padding(20.dp)
) {

    SafetyScoreCard()

    Spacer(modifier = Modifier.height(16.dp))

    LocationCard()

    Spacer(modifier = Modifier.height(16.dp))

    EmergencyButton(
        onClick = { navController.navigate("emergency_sos") }
    )

    Spacer(modifier = Modifier.height(24.dp))

    Text(
        text = "Quick Actions",
        fontSize = 18.sp,
        fontWeight = FontWeight.SemiBold,
        color = Color(0xFF475569)
    )

    Spacer(modifier = Modifier.height(12.dp))

    Column(
        verticalArrangement = Arrangement.spacedBy(12.dp)
    ) {
        Row(
            modifier = Modifier.fillMaxWidth(),
            horizontalArrangement = Arrangement.spacedBy(12.dp)
        ) {
            QuickActionCard(
                action = quickActions[0],
                modifier = Modifier.weight(1f),
                onClick = { navController.navigate("safety_tips") }
            )
            QuickActionCard(
                action = quickActions[1],
                modifier = Modifier.weight(1f),
                onClick = { navController.navigate("efir") }
            )
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
Row(
    modifier = Modifier.fillMaxWidth(),
    horizontalArrangement = Arrangement.spacedBy(12.dp)
) {
    QuickActionCard(
        action = quickActions[2],
        modifier = Modifier.weight(1f),
        onClick = { navController.navigate("services") }
    )
    QuickActionCard(
        action = quickActions[3],
        modifier = Modifier.weight(1f),
        onClick = { navController.navigate("danger_zone") }
}
)
}
}

Spacer(modifier = Modifier.height(20.dp))
}
}
}

@Composable
fun SafetyScoreCard() {
Card(
    shape = RoundedCornerShape(20.dp),
    modifier = Modifier.fillMaxWidth(),
    colors = CardDefaults.cardColors(containerColor = Color.White),
    elevation = CardDefaults.cardElevation(defaultElevation = 0.dp)
) {
    Row(
        modifier = Modifier
            .padding(24.dp)
            .fillMaxWidth(),
        verticalAlignment = Alignment.CenterVertically
    ) {
        Box(
            modifier = Modifier
                .size(80.dp)
                .clip(CircleShape)
                .background(
                    Brush.linearGradient(
                        colors = listOf(Color(0xFF10B981),
Color(0xFF059669)))
    
```

Smart Tourist Safety Monitoring & Incident Response

```
        )
    ),
    contentAlignment = Alignment.Center
) {
Column(
    horizontalAlignment = Alignment.CenterHorizontally
) {
    Text(
        text = "90",
        fontSize = 28.sp,
        fontWeight = FontWeight.Bold,
        color = Color.White
    )
    Text(
        text = "/100",
        fontSize = 11.sp,
        fontWeight = FontWeight.Medium,
        color = Color.White.copy(alpha = 0.9f)
    )
}
}

Spacer(modifier = Modifier.width(20.dp))

Column {
    Text(
        text = "Safety Score",
        fontWeight = FontWeight.Bold,
        fontSize = 20.sp,
        color = Color(0xFF1E293B)
    )
    Spacer(modifier = Modifier.height(8.dp))
    Surface(
        shape = RoundedCornerShape(12.dp),
        color = Color(0xFFD1FAE5)
    ) {
        Text(
            modifier = Modifier.padding(horizontal = 12.dp,
vertical = 6.dp),
            text = "Safe",
            color = Color(0xFF059669),
            fontWeight = FontWeight.SemiBold,
            fontSize = 13.sp
        )
    }
}
Spacer(modifier = Modifier.height(6.dp))
```

Smart Tourist Safety Monitoring & Incident Response

```
        Text(
            text = "Low risk detected in your area",
            fontSize = 13.sp,
            color = Color(0xFF64748B)
        )
    }
}
}

@Composable
fun LocationCard() {
    val placeName = "Shanku's Water Park, SH41, Mevad"
    val coordinates = "23.474803, 72.391667"
    val accuracy = "±5m"

    Card(
        modifier = Modifier.fillMaxWidth(),
        shape = RoundedCornerShape(20.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 0.dp)
    ) {
        Column(
            modifier = Modifier.padding(24.dp)
        ) {
            Row(
                verticalAlignment = Alignment.CenterVertically
            ) {
                Box(
                    modifier = Modifier
                        .size(44.dp)
                        .clip(CircleShape)
                        .background(Color(0xFFDCEEFB)),
                    contentAlignment = Alignment.Center
                )
                Icon(
                    imageVector = Icons.Outlined.MyLocation,
                    contentDescription = "Location",
                    tint = Color(0xFF3B82F6),
                    modifier = Modifier.size(24.dp)
                )
            }
            Spacer(modifier = Modifier.width(16.dp))

            Column(modifier = Modifier.weight(1f)) {
```

Smart Tourist Safety Monitoring & Incident Response

```
        Text(
            text = "Current Location",
            fontWeight = FontWeight.Bold,
            fontSize = 16.sp,
            color = Color(0xFF1E293B)
        )
        Spacer(modifier = Modifier.height(4.dp))
        Row(verticalAlignment = Alignment.CenterVertically) {
            Box(
                modifier = Modifier
                    .size(8.dp)
                    .clip(CircleShape)
                    .background(Color(0xFF10B981)))
            )
            Spacer(modifier = Modifier.width(6.dp))
            Text(
                text = "Sharing active",
                fontSize = 13.sp,
                color = Color(0xFF10B981),
                fontWeight = FontWeight.Medium
            )
        }
    }
}

Spacer(modifier = Modifier.height(20.dp))

Text(
    text = placeName,
    fontWeight = FontWeight.SemiBold,
    fontSize = 15.sp,
    color = Color(0xFF334155),
    maxLines = 2,
    overflow = TextOverflow.Ellipsis,
    lineHeight = 20.sp
)

Spacer(modifier = Modifier.height(12.dp))

Row(
    modifier = Modifier.fillMaxWidth(),
    horizontalArrangement = Arrangement.SpaceBetween,
    verticalAlignment = Alignment.CenterVertically
) {
    Text(
        text = coordinates,
```

Smart Tourist Safety Monitoring & Incident Response

```
        fontSize = 13.sp,
        color = Color(0xFF64748B),
        fontWeight = FontWeight.Medium
    )

    Surface(
        shape = RoundedCornerShape(8.dp),
        color = Color(0xFFD1FAE5)
    ) {
        Text(
            modifier = Modifier.padding(horizontal = 10.dp,
vertical = 4.dp),
            text = accuracy,
            fontSize = 12.sp,
            color = Color(0xFF059669),
            fontWeight = FontWeight.SemiBold
        )
    }
}
}

@Composable
fun EmergencyButton(onClick: () -> Unit = {}) {
Card(
    modifier = Modifier
        .fillMaxWidth()
        .clickable(onClick = onClick),
    shape = RoundedCornerShape(20.dp),
    colors = CardDefaults.cardColors(
        containerColor = Color.Transparent
    ),
    elevation = CardDefaults.cardElevation(defaultElevation = 0.dp)
) {
    Box(
        modifier = Modifier
            .fillMaxWidth()
            .background(
                Brush.horizontalGradient(
                    colors = listOf(Color(0xFFDC2626),
Color(0xFFFF4444))
            )
        )
        .padding(vertical = 28.dp),
        contentAlignment = Alignment.Center
    )
}
```

Smart Tourist Safety Monitoring & Incident Response

```
) {
    Row(
        verticalAlignment = Alignment.CenterVertically,
        horizontalArrangement = Arrangement.Center
    ) {
        Box(
            modifier = Modifier
                .size(48.dp)
                .clip(CircleShape)
                .background(Color.White.copy(alpha = 0.2f)),
            contentAlignment = Alignment.Center
        ) {
            Icon(
                imageVector = Icons.Default.Warning,
                contentDescription = "SOS",
                tint = Color.White,
                modifier = Modifier.size(28.dp)
            )
        }
    }

    Spacer(modifier = Modifier.width(16.dp))

    Column {
        Text(
            text = "EMERGENCY SOS",
            color = Color.White,
            fontSize = 20.sp,
            fontWeight = FontWeight.Bold,
            letterSpacing = 0.5.sp
        )
        Text(
            text = "Tap to trigger alert",
            color = Color.White.copy(alpha = 0.9f),
            fontSize = 13.sp,
            fontWeight = FontWeight.Medium
        )
    }
}
}

@Composable
fun QuickActionCard(
    action: QuickAction,
    modifier: Modifier = Modifier,
```

Smart Tourist Safety Monitoring & Incident Response

```
    onClick: () -> Unit = {}
) {
Card(
    modifier = modifier
        .height(130.dp)
        .clickable(onClick = onClick),
    shape = RoundedCornerShape(20.dp),
    colors = CardDefaults.cardColors(containerColor = Color.White),
    elevation = CardDefaults.cardElevation(defaultElevation = 0.dp)
) {
Box(
    modifier = Modifier.fillMaxSize()
) {
Column(
    modifier = Modifier
        .fillMaxSize()
        .padding(16.dp),
    horizontalAlignment = Alignment.Start,
    verticalArrangement = Arrangement.SpaceBetween
) {
Box(
    modifier = Modifier
        .size(48.dp)
        .clip(RoundedCornerShape(12.dp))
        .background(action.bgColor.copy(alpha = 0.1f)),
    contentAlignment = Alignment.Center
) {
Icon(
    imageVector = action.icon,
    contentDescription = action.title,
    tint = action.bgColor,
    modifier = Modifier.size(24.dp)
)
}
}

Column {
    Text(
        text = action.title,
        fontWeight = FontWeight.Bold,
        fontSize = 16.sp,
        color = Color(0xFF1E293B)
    )
    Spacer(modifier = Modifier.height(4.dp))
    Text(
        text = action.subtitle,
        fontSize = 12.sp,
```

Smart Tourist Safety Monitoring & Incident Response

```
        color = Color(0xFF64748B),
        maxLines = 1,
        overflow = TextOverflow.Ellipsis
    )
}
}
}
}
```

```
@Composable
@Preview(showBackground = true)
fun DashboardPreview() {
    Dashboard(navController =
        androidx.navigation.compose.rememberNavController())
}
```

DangerZoneScreen.kt

```
package com.bytemap.a24172012068_assignment.screen

import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.*
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.vector.ImageVector
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
```

```
data class DangerZone(
    val name: String,
    val riskLevel: String,
    val description: String,
    val lastReported: String,
    val distance: String,
    val icon: ImageVector,
```

Smart Tourist Safety Monitoring & Incident Response

```
    val color: Color
)

@Composable
fun DangerZoneScreen(navController: NavController) {
    val dangerZones = listOf(
        DangerZone(
            "Downtown Market Area",
            "High Risk",
            "Multiple incidents reported in the last 24 hours",
            "2 hours ago",
            "1.2 km",
            Icons.Default.Warning,
            Color(0xFFE53E3E)
        ),
        DangerZone(
            "Industrial Zone",
            "Medium Risk",
            "Isolated area with limited lighting",
            "6 hours ago",
            "3.5 km",
            Icons.Default.Construction,
            Color(0xFFFF9800)
        ),
        DangerZone(
            "Parking Lot - Mall",
            "High Risk",
            "Recent theft incidents reported",
            "4 hours ago",
            "2.1 km",
            Icons.Default.LocalParking,
            Color(0xFFE53E3E)
        ),
        DangerZone(
            "Subway Station",
            "Low Risk",
            "Generally safe but stay alert",
            "1 day ago",
            "0.8 km",
            Icons.Default.Train,
            Color(0xFF4CAF50)
        ),
        DangerZone(
            "Night Club District",
            "Medium Risk",
            "Late night incidents common",
            "3 hours ago",
            "0.5 km",
            Icons.Default.NightStay,
            Color(0xFF980080)
        )
    )
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        "8 hours ago",
        "4.2 km",
        Icons.Default.Nightlife,
        Color(0xFFFF9800)
    )
)

Column(
    modifier = Modifier
        .fillMaxSize()
        .background(Color(0xFFF8F9FA))
        .verticalScroll(rememberScrollState())
        .padding(16.dp)
) {
    Card(
        modifier = Modifier.fillMaxWidth(),
        shape = RoundedCornerShape(16.dp),
        colors      =      CardDefaults.cardColors(containerColor      =
Color.White),
        elevation   =   CardDefaults.cardElevation(defaultElevation   =
2.dp)
    ) {
        Column(
            modifier = Modifier.padding(20.dp),
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Icon(
                imageVector = Icons.Default.LocationOn,
                contentDescription = "Danger Zone Icon",
                tint = Color(0xFFE53E3E),
                modifier = Modifier.size(48.dp)
            )
            Spacer(modifier = Modifier.height(12.dp))
            Text(
                text = "Danger Zones",
                fontSize = 24.sp,
                fontWeight = FontWeight.Bold,
                color = Color(0xFF2C3E50)
            )
            Text(
                text = "Areas with reported safety incidents",
                fontSize = 14.sp,
                color = Color(0xFF7F8C8D),
                textAlign
            androidx.compose.ui.text.style.TextAlign.Center
        )
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        }

    }

    Spacer(modifier = Modifier.height(20.dp))

    Card(
        modifier = Modifier.fillMaxWidth(),
        shape = RoundedCornerShape(12.dp),
        colors      =      CardDefaults.cardColors(containerColor      =
Color.White),
        elevation   =      CardDefaults.cardElevation(defaultElevation   =
2.dp)
    ) {
        Column(
            modifier = Modifier.padding(16.dp)
        ) {
            Text(
                text = "Risk Levels",
                fontSize = 16.sp,
                fontWeight = FontWeight.Bold,
                color = Color(0xFF2C3E50)
            )
            Spacer(modifier = Modifier.height(12.dp))

            Row(
                modifier = Modifier.fillMaxWidth(),
                horizontalArrangement = Arrangement.SpaceEvenly
            ) {
                RiskLevelIndicator("High", Color(0xFFE53E))
                RiskLevelIndicator("Medium", Color(0xFFFF9800))
                RiskLevelIndicator("Low", Color(0xFF4CAF50))
            }
        }
    }

    Spacer(modifier = Modifier.height(20.dp))

    Column(
        verticalArrangement = Arrangement.spacedBy(12.dp)
    ) {
        dangerZones.forEach { zone ->
            DangerZoneCard(zone = zone)
        }
    }

    Spacer(modifier = Modifier.height(20.dp))
```

Smart Tourist Safety Monitoring & Incident Response

```
Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(16.dp),
    colors      =      CardDefaults.cardColors(containerColor      =
Color(0xFFE3F2FD)),
    elevation   =      CardDefaults.cardElevation(defaultElevation   =
2.dp)
) {
    Column(
        modifier = Modifier.padding(20.dp),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Icon(
            imageVector = Icons.Default.Security,
            contentDescription = "Safety Icon",
            tint = Color(0xFF2196F3),
            modifier = Modifier.size(32.dp)
        )
        Spacer(modifier = Modifier.height(8.dp))
        Text(
            text = "Stay Safe",
            color = Color(0xFF1976D2),
            fontSize = 18.sp,
            fontWeight = FontWeight.Bold
        )
        Text(
            text = "Avoid high-risk areas when possible. If you
must go, stay alert and share your location.",
            color = Color(0xFF1976D2),
            fontSize = 14.sp,
            textAlign
        androidx.compose.ui.text.style.TextAlign.Center
    )
}
}
}

@Composable
fun RiskLevelIndicator(level: String, color: Color) {
    Column(
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Box(
            modifier = Modifier
```

Smart Tourist Safety Monitoring & Incident Response

```
.size(16.dp)
    .background(color, RoundedCornerShape(8.dp))
)
Spacer(modifier = Modifier.height(4.dp))
Text(
    text = level,
    fontSize = 12.sp,
    color = Color(0xFF7F8C8D)
)
}
}

@Composable
fun DangerZoneCard(zone: DangerZone) {
Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(12.dp),
    colors = CardDefaults.cardColors(containerColor = Color.White),
    elevation = CardDefaults.cardElevation(defaultElevation = 2.dp)
) {
Column(
    modifier = Modifier.padding(16.dp)
) {
Row(
    modifier = Modifier.fillMaxWidth(),
    horizontalArrangement = Arrangement.SpaceBetween,
    verticalAlignment = Alignment.CenterVertically
) {
Row(
    verticalAlignment = Alignment.CenterVertically
) {
Card(
    shape = RoundedCornerShape(8.dp),
    colors = CardDefaults.cardColors(containerColor =
zone.color.copy(alpha = 0.1f)),
    modifier = Modifier.size(40.dp)
) {
Box(
    modifier = Modifier.fillMaxSize(),
    contentAlignment = Alignment.Center
) {
Icon(
    imageVector = zone.icon,
    contentDescription = "${zone.name} Icon",
    tint = zone.color,
    modifier = Modifier.size(20.dp)
}
}
}
}
}
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        )
    }
}

Spacer(modifier = Modifier.width(12.dp))

Column {
    Text(
        text = zone.name,
        fontSize = 16.sp,
        fontWeight = FontWeight.Bold,
        color = Color(0xFF2C3E50)
    )
    Text(
        text = zone.distance,
        fontSize = 12.sp,
        color = Color(0xFF7F8C8D)
    )
}
}

Card(
    shape = RoundedCornerShape(12.dp),
    colors = CardDefaults.cardColors(containerColor =
zone.color.copy(alpha = 0.1f))
) {
    Text(
        text = zone.riskLevel,
        fontSize = 12.sp,
        fontWeight = FontWeight.Bold,
        color = zone.color,
        modifier = Modifier.padding(horizontal = 8.dp,
vertical = 4.dp)
    )
}
}

Spacer(modifier = Modifier.height(8.dp))

Text(
    text = zone.description,
    fontSize = 14.sp,
    color = Color(0xFF7F8C8D)
)

Spacer(modifier = Modifier.height(8.dp))
```

Smart Tourist Safety Monitoring & Incident Response

```
        Text(
            text = "Last reported: ${zone.lastReported}",
            fontSize = 12.sp,
            color = Color(0xFF95A5A6),
            fontStyle = androidx.compose.ui.text.font.FontStyle.Italic
        )
    }
}
}

@Preview(showBackground = true)
@Composable
fun DangerZoneScreenPreview() {
    DangerZoneScreen(navController =
        androidx.navigation.compose.rememberNavController())
}
```

EFirScreen.kt

```
package com.bytemap.a24172012068_assignment.screen

import android.app.DatePickerDialog
import android.app.TimePickerDialog
import android.icu.util.Calendar
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.*
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.Checkbox
import androidx.compose.material3.DatePicker
import androidx.compose.material3.FilterChip
import androidx.compose.material3.Icon
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.getValue
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.ui.graphics.vector.ImageVector
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.font.FontStyle
import androidx.navigation.NavController

data class Incident(
    val name: String,
    val icon: ImageVector
)

@Composable
fun EFirScreen(navController: NavController){
    Box(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFFFF8F9FA))
    ) {
        EFirUI()
    }
}

@Composable
fun EFirUI() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .verticalScroll(rememberScrollState())
            .padding(16.dp),
        verticalArrangement = Arrangement.spacedBy(16.dp)
    ) {
        IncidentType()
        IncidentDescription()
        DateTime()
        LocationInput()
        WitnessesInput()
        AdditionalDetails()
        SubmitButton()
    }
}

@Composable
```

Smart Tourist Safety Monitoring & Incident Response

```
fun IncidentType() {
    val incidentTypes = listOf(
        Incident("Harassment", Icons.Default.Block),
        Incident("Theft", Icons.Default.Lock),
        Incident("Assault", Icons.Default.Dangerous),
        Incident("Fraud", Icons.Default.CreditCard),
        Incident("Emergency", Icons.Default.Emergency),
        Incident("Other", Icons.Default.Notes)
    )

    var selectedIncident by remember { mutableStateOf<String?>(null) }

    Text(
        text = "Incident Type",
        modifier = Modifier.padding(bottom = 8.dp),
        fontSize = 18.sp,
        color = Color.DarkGray,
        fontWeight = FontWeight.Medium
    )

    FlowRow(
        horizontalArrangement = Arrangement.spacedBy(8.dp),
        verticalArrangement = Arrangement.spacedBy(8.dp),
        modifier = Modifier.fillMaxWidth()
    ) {
        incidentTypes.forEach { incident ->
            FilterChip(
                selected = selectedIncident == incident.name,
                onClick = {
                    selectedIncident = if (selectedIncident == incident.name) null else incident.name
                },
                label = {
                    Row(
                        verticalAlignment = Alignment.CenterVertically
                    ) {
                        Icon(
                            imageVector = incident.icon,
                            contentDescription = "${incident.name} Icon",
                            modifier = Modifier.size(18.dp)
                        )
                        Spacer(modifier = Modifier.width(6.dp))
                        Text(text = incident.name)
                    }
                },
                leadingIcon = if (selectedIncident == incident.name) {
```

Smart Tourist Safety Monitoring & Incident Response

```
        {
            Icon(
                imageVector = Icons.Default.Check,
                contentDescription = "Selected",
                modifier = Modifier.size(18.dp)
            )
        }
    } else null
)
}
}

Text(
    text = "Other incidents",
    fontStyle = FontStyle.Italic,
    color = Color.Gray,
    modifier = Modifier.padding(top = 8.dp)
)
}

@Composable
fun IncidentDescription() {
    var description by remember { mutableStateOf("") }
    Text(
        text = "Incident Description *",
        modifier = Modifier.padding(top = 16.dp, bottom = 8.dp),
        fontSize = 18.sp,
        color = Color.DarkGray,
        fontWeight = FontWeight.Medium
    )
    OutlinedTextField(
        value = description,
        onValueChange = {description = it},
        modifier = Modifier
            .fillMaxWidth()
            .height(150.dp),
        placeholder = { Text(text = "Describe what happened in detail..."),
color = Color.Gray, fontSize = 14.sp) },
        maxLines = 6
    )
    Row(
        modifier = Modifier.fillMaxWidth(),
        horizontalArrangement = Arrangement.SpaceBetween,
        verticalAlignment = Alignment.CenterVertically
    ) {
        Text(
```

Smart Tourist Safety Monitoring & Incident Response

```
        text = "Be as specific as possible",
        color = Color.Gray,
        fontSize = 12.sp
    )
    Text(
        text = "0/5000",
        color = Color.Gray,
        fontSize = 12.sp
    )
)
}

@Composable
fun DateTime() {
    val context = LocalContext.current
    var selectedDate by remember { mutableStateOf("") }
    val calendar = Calendar.getInstance()

    val datePickerDialog = DatePickerDialog(
        context,
        { _, year, month, dayOfMonth ->
            val timePickerDialog = TimePickerDialog(
                context,
                { _, hourOfDay, minute ->
                    selectedDate = "$dayOfMonth/${month + 1}/$year - "
                    ${"%02d".format(hourOfDay)}:${"%02d".format(minute)}"
                },
                calendar.get(Calendar.HOUR_OF_DAY),
                calendar.get(Calendar.MINUTE),
                false
            )
            timePickerDialog.show()
        },
        calendar.get(Calendar.YEAR),
        calendar.get(Calendar.MONTH),
        calendar.get(Calendar.DAY_OF_MONTH)
    )

    OutlinedTextField(
        value = selectedDate,
        onValueChange = { },
        modifier = Modifier
            .fillMaxWidth()
            .padding(8.dp)
            .clickable { datePickerDialog.show() },
        label = {
```

Smart Tourist Safety Monitoring & Incident Response

```
Row(verticalAlignment = Alignment.CenterVertically) {
    Icon(
        imageVector = Icons.Default.DateRange,
        contentDescription = "Date Icon",
        tint = Color.Gray,
        modifier = Modifier.size(18.dp)
    )
    Spacer(modifier = Modifier.width(8.dp))
    Text(
        text = "Select Date & Time",
        color = Color.Gray,
        fontSize = 14.sp
    )
}
},
placeholder = {
    Text(
        text = "Tap to pick date and time",
        color = Color.Gray,
        fontSize = 13.sp
    )
},
readOnly = true,
enabled = true
)
}

@Composable
fun LocationInput(){
    Text(
        text = "Location",
        modifier = Modifier.padding(bottom = 8.dp),
        fontSize = 18.sp,
        color = Color.DarkGray,
        fontWeight = FontWeight.Medium
    )
    Row(
        verticalAlignment = Alignment.CenterVertically
    ) {
        Checkbox( checked = false, onCheckedChange = { } )
        Text(text = "Use current location", color = Color.Gray)
    }
    OutlinedTextField(
        value = "",
        onValueChange = { },
        leadingIcon = {
```

Smart Tourist Safety Monitoring & Incident Response

```
Icon(
    imageVector = Icons.Default.LocationOn,
    contentDescription = "Location Icon",
    tint = Color.Gray
)
},
modifier = Modifier
    .fillMaxWidth()
    .padding(top = 8.dp),
placeholder = { Text(text = "Enter location details...", color =
Color.Gray, fontSize = 14.sp) }
)
}
}

@Composable
fun WitnessesInput(){
    var witnesses by remember { mutableStateOf("") }
    Text(
        text = "Witnesses (Optional)",
        modifier = Modifier.padding(top = 16.dp, bottom = 8.dp),
        fontSize = 18.sp,
        color = Color.DarkGray,
        fontWeight = FontWeight.Medium
    )
    Row {
        OutlinedTextField(
            value = witnesses,
            onValueChange = { witnesses = it },
            modifier = Modifier
                .weight(1f)
                .heightIn(min = 56.dp)
                .align(Alignment.CenterVertically),
            placeholder = { Text(text = "Enter witness details...", color =
Color.Gray, fontSize = 14.sp) }
        )
        Button(
            onClick = { },
            shape = RoundedCornerShape(8.dp),
            modifier = Modifier
                .padding(start = 8.dp)
                .heightIn(min = 56.dp)
                .align(Alignment.CenterVertically)
        ) {
            Text(text = "+ Add", fontSize = 16.sp)
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        }
    }

@Composable
fun AdditionalDetails() {
    var additional by remember { mutableStateOf("") }
    Text(
        text = "Additional Details (Optional)",
        modifier = Modifier.padding(top = 16.dp, bottom = 8.dp),
        fontSize = 18.sp,
        color = Color.DarkGray,
        fontWeight = FontWeight.Medium
    )
    OutlinedTextField(
        value = additional,
        onValueChange = {additional = it},
        modifier = Modifier
            .fillMaxWidth()
            .height(150.dp),
        placeholder = { Text(text = "Any additional information, evidence
, or notes...", color = Color.Gray, fontSize = 14.sp) },
        maxLines = 6
    )

    Row(
        modifier = Modifier.fillMaxWidth(),
        horizontalArrangement = Arrangement.End
    ) {
        Text(
            text = "0/2000",
            color = Color.Gray,
            fontSize = 12.sp
        )
    }
}

@Composable
fun SubmitButton() {
    Button(
        onClick = {},
        shape = RoundedCornerShape(8.dp),
        colors = ButtonDefaults.buttonColors(
            containerColor = Color(0xFFD32F2F),
            contentColor = Color.White
    )
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        ),
        modifier = Modifier
            .fillMaxWidth()
            .padding(vertical = 20.dp)
            .height(50.dp)
    ) {
    Icon(
        imageVector = Icons.Default.InsertDriveFile,
        contentDescription = "Submit Icon",
        modifier = Modifier.size(20.dp)
    )
    Spacer(modifier = Modifier.width(8.dp))
    Text(text = "Submit Report", fontSize = 18.sp)
}
}

@Preview(showBackground = true)
@Composable
fun EFirPreview() {
    EFirUI()
}
```

EmergencyContactsScreen.kt

```
package com.bytemap.a24172012068_assignment.screen
```

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.Add
import androidx.compose.material.icons.filled.MoreVert
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.tooling.preview.Preview
import androidx.navigation.NavController
```

Smart Tourist Safety Monitoring & Incident Response

```
data class EmergencyContact(val name: String, val number: String, val label: String = "Parent")

@Composable
fun EmergencyContactsScreen(navController: NavController) {
    val contacts = listOf(
        EmergencyContact("PAPA", "8745214752"),
        EmergencyContact("MOTHER", "7547862145")
    )

    Box(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFFFF57FA))
    ) {
        Column(
            modifier = Modifier
                .fillMaxSize()
                .verticalScroll(rememberScrollState())
                .padding(horizontal = 14.dp, vertical = 10.dp),
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            contacts.forEach {
                MiniContactCard(contact = it)
                Spacer(Modifier.height(10.dp))
            }
        }
        Box(
            contentAlignment = Alignment.BottomEnd,
            modifier = Modifier
                .fillMaxSize()
                .padding(18.dp)
        ) {
            FloatingActionButton(
                shape = RoundedCornerShape(12.dp),
                containerColor = Color(0xFF1937B3),
                onClick = { },
                modifier = Modifier.size(50.dp)
            ) {
                Icon(
                    Icons.Default.Add,
                    contentDescription = "Add Contact",
                    tint = Color.White,
                    modifier = Modifier.size(22.dp)
                )
            }
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        }
    }
}

@Composable
fun MiniContactCard(contact: EmergencyContact) {
    Card(
        shape = RoundedCornerShape(16.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        modifier = Modifier
            .fillMaxWidth()
            .height(IntrinsicSize.Min),
        elevation = CardDefaults.cardElevation(2.dp)
    ) {
        Row(
            modifier = Modifier.padding(12.dp),
            verticalAlignment = Alignment.CenterVertically
        ) {
            Box(
                modifier = Modifier
                    .size(38.dp)
                    .clip(CircleShape)
                    .background(Color(0xFF1937B3)),
                contentAlignment = Alignment.Center
            ) {
                Text(
                    text = contact.name.first().toString(),
                    color = Color.White,
                    fontWeight = FontWeight.SemiBold,
                    fontSize = 16.sp
                )
            }
            Spacer(Modifier.width(10.dp))
            Column(
                modifier = Modifier.weight(1f),
                verticalArrangement = Arrangement.spacedBy(3.dp)
            ) {
                Text(
                    contact.name,
                    fontWeight = FontWeight.Medium,
                    fontSize = 14.sp,
                    color = Color(0xFF0B1432)
                )
                Text(
                    contact.number,
                    color = Color(0xFF6B7A99),
                )
            }
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        fontSize = 13.sp,
        fontWeight = FontWeight.Normal
    )
    Box(
        modifier = Modifier
            .clip(RoundedCornerShape(7.dp))
            .background(Color(0xFFE8EAF6))
            .padding(horizontal = 8.dp, vertical = 3.dp)
    ) {
        Text(
            contact.label,
            color = Color(0xFF1937B3),
            fontSize = 11.sp,
            fontWeight = FontWeight.SemiBold
        )
    }
}
IconButton(
    onClick = { },
    modifier = Modifier.size(32.dp)
) {
    Icon(
        Icons.Default.MoreVert,
        contentDescription = "More options",
        tint = Color(0xFF9CA3AF),
        modifier = Modifier.size(16.dp)
    )
}
}
}

@Preview(showBackground = true)
@Composable
fun EmergencyContactsScreenPreview() {
    EmergencyContactsScreen(navController =
```

LocationHistoryScreen.kt

package com.bytemap.a24172012068_assignment.screen

```
import androidx.compose.foundation.layout.*  
import androidx.compose.material3.*  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController

@Composable
fun LocationHistoryScreen(navController: NavController) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text(
            text = "Location History",
            fontSize = 24.sp,
            fontWeight = FontWeight.Bold
        )

        Spacer(modifier = Modifier.height(16.dp))

        Text(
            text = "View your location history here",
            fontSize = 16.sp
        )
    }
}

@Preview(showBackground = true)
@Composable
fun LocationHistoryScreenPreview() {
    LocationHistoryScreen(navController =
        androidx.navigation.compose.rememberNavController())
}
```

LoginScreen.kt

```
package com.bytemap.a24172012068_assignment.screen

import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Box
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons)
import androidx.compose.material.icons.filled.Person
import androidx.compose.material.icons.filled.VisibilityOff
import androidx.compose.material3.Button
import androidx.compose.material3.ButtonColors
import androidx.compose.material3.ButtonDefaults
import androidx.compose.material3.Checkbox
import androidx.compose.material3.Text
import com.bytemap.a24172012068_assignment.R
import androidx.compose.runtime.Composable
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.getValue
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.bytemap.a24172012068_assignment.components.FormField

@Composable
fun LoginScreen() {
    var emailState by remember { mutableStateOf("") }
    var passwordState by remember { mutableStateOf("") }
    var isChecked by remember { mutableStateOf(false) }
    Column(
        horizontalAlignment = Alignment.CenterHorizontally,
        modifier = Modifier
            .fillMaxSize()
            .verticalScroll(rememberScrollState())
            .padding(20.dp)
    ) {
        Image(painter = painterResource(R.drawable.logo),
            contentDescription = "Login Image",
            modifier = Modifier
```

Smart Tourist Safety Monitoring & Incident Response

```
.align(
    alignment = Alignment.CenterHorizontally
)
.size(150.dp)
.padding(0.dp, 50.dp, 0.dp, 10.dp))

Text(text = "Welcome Back", modifier = Modifier
    .align(
        alignment = Alignment.CenterHorizontally
    )
    .padding(0.dp, 20.dp, 0.dp, 10.dp),
    fontSize = 25.sp,
    fontWeight = FontWeight.Bold)
Spacer(modifier = Modifier.height(10.dp))
Text(text = "Sign in to access your account",
    modifier = Modifier.align(
        alignment = Alignment.CenterHorizontally),
    fontSize = 16.sp,
    )
Spacer(modifier = Modifier.height(30.dp))
FormField(
    label = "Email",
    textState = emailState,
    onTextChange = { emailState = it },
    icon = Icons.Default.Person
)
Spacer(modifier = Modifier.height(10.dp))
FormField(
    label = "Password",
    textState = passwordState,
    onTextChange = { passwordState = it },
    icon = Icons.Default.VisibilityOff,
    isPasswordField = true
)
Spacer(modifier = Modifier.height(10.dp))
Row(
    modifier = Modifier.fillMaxWidth(),
    verticalAlignment = Alignment.CenterVertically,
    horizontalArrangement =
)
    androidx.compose.foundation.layout.Arrangement.SpaceBetween
)
{
    Row(verticalAlignment = Alignment.CenterVertically) {
        Checkbox(
            checked = isChecked,
            onCheckedChange = { isChecked = it }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        )
    Text(
        text = "Remember Me",
        fontSize = 15.sp,
        modifier = Modifier.padding(start = 8.dp)
    )
}

Text(
    text = "Forgot Password?",
    fontSize = 15.sp
)
}

Button( modifier = Modifier.width(250.dp)
    .height(90.dp)
    .padding(top = 30.dp),
    colors = ButtonDefaults.buttonColors(
        containerColor = Color(0xFF000000),
        contentColor = Color.White
    )
    , onClick = {
})
{
    Text(text = "Login", fontSize = 20.sp)
}
Spacer(modifier = Modifier.height(20.dp))
Row (
    modifier = Modifier.align(Alignment.CenterHorizontally),
    verticalAlignment = Alignment.CenterVertically
){
    Text(
        text = "Don't have an account?",
        fontSize = 18.sp,
    )
    Spacer(modifier = Modifier.width(5.dp))
    Text(
        text = "Sign up",
        fontSize = 18.sp,
        color = Color.Red
    )
}
}
}

@Composable
@Preview(showBackground = true)
```

Smart Tourist Safety Monitoring & Incident Response

```
fun LoginScreenPreview() {
    LoginScreen()
}

MapScreen.kt
package com.bytemap.a24172012068_assignment.screen

import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.MyLocation
import androidx.compose.material.icons.filled.Search
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalInspectionMode
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import com.bytemap.a24172012068_assignment.compose.SimpleMapView
import org.osmdroid.util.GeoPoint

@Composable
fun MapScreen() {
    var searchQuery by remember { mutableStateOf("") }

    Box(modifier = Modifier.fillMaxSize()) {
        SimpleMapView(
            modifier = Modifier.fillMaxSize(),
            initialLocation = GeoPoint(37.7749, -122.4194)
        )

        Card(
            modifier = Modifier
                .fillMaxWidth()
                .padding(horizontal = 16.dp, vertical = 12.dp)
                .align(Alignment.TopCenter),
            shape = RoundedCornerShape(12.dp),
            elevation = CardDefaults.cardElevation(defaultElevation =
4.dp),
            colors = CardDefaults.cardColors(containerColor =
MaterialTheme.colorScheme.surface)
        ) {
            OutlinedTextField(
                value = searchQuery,
                onValueChange = { searchQuery = it },
            )
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        modifier = Modifier
            .fillMaxWidth()
            .padding(8.dp),
        placeholder = { Text("Search location...") },
        leadingIcon = {
            Icon(
                imageVector = Icons.Default.Search,
                contentDescription = "Search",
                tint = MaterialTheme.colorScheme.primary
            )
        },
        singleLine = true,
        shape = RoundedCornerShape(8.dp),
        colors = OutlinedTextFieldDefaults.colors(
            focusedBorderColor = Color.Transparent,
            unfocusedBorderColor = Color.Transparent
        )
    )
}

FloatingActionButton(
    onClick = { },
    modifier = Modifier
        .align(Alignment.BottomEnd)
        .padding(16.dp),
    containerColor = MaterialTheme.colorScheme.primary,
    elevation
    FloatingActionButtonDefaults.elevation(defaultElevation = 6.dp)
) {
    Icon(
        imageVector = Icons.Default.MyLocation,
        contentDescription = "My Location",
        tint = Color.White
    )
}
}

@Preview(showBackground = true)
@Composable
fun MapScreenPreview() {
    MaterialTheme {
        MapScreen()
    }
}
```

NotificationsScreen.kt

Smart Tourist Safety Monitoring & Incident Response

```
package com.bytemap.a24172012068_assignment.screen

import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController

@Composable
fun NotificationsScreen(navController: NavController) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Text(
            text = "Notifications",
            fontSize = 24.sp,
            fontWeight = FontWeight.Bold
        )

        Spacer(modifier = Modifier.height(16.dp))

        Text(
            text = "No notifications yet",
            fontSize = 16.sp
        )
    }
}

@Preview(showBackground = true)
@Composable
fun NotificationsScreenPreview() {
    NotificationsScreen(navController =
        androidx.navigation.compose.rememberNavController())
}
```

OurTripScreen.kt

```
package com.bytemap.a24172012068_assignment.screen
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.*
import androidx.compose.material.icons.Icons
import androidx.compose.material.icons.filled.LocationOn
import androidx.compose.material.icons.filled.Refresh
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

@Composable
fun OurTripScreen() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(16.dp)
    ) {
        Card(
            modifier = Modifier.fillWidth(),
            shape = RoundedCornerShape(16.dp),
            elevation = CardDefaults.cardElevation(defaultElevation =
4.dp)
        ) {
            Column(modifier = Modifier.padding(16.dp)) {
                Row(verticalAlignment = Alignment.CenterVertically) {
                    Icon(
                        imageVector = Icons.Default.LocationOn,
                        contentDescription = "Location",
                        tint = Color(0xFF149EE7)
                    )
                    Spacer(Modifier.width(8.dp))
                    Text(
                        text = "Current Location",
                        style
                    )
                    MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold)
                }
            }
            Spacer(Modifier.height(8.dp))
            Text("Address: Shanku's Water Park, SH41")
            Text("Coordinates: 23.474803, 72.391667")
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        Text("Speed: 0.0 km/h")
        Text("Updates: 1 auto-updates completed")
    }
}

Spacer(Modifier.height(16.dp))

Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(16.dp),
    elevation = CardDefaults.cardElevation(defaultElevation =
4.dp)
) {
    Column(modifier = Modifier.padding(16.dp)) {
        Text(
            text = "Set Destination",
            style
MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold)
        )
        Spacer(Modifier.height(8.dp))
        OutlinedTextField(
            value = "",
            onValueChange = {},
            label = { Text("Enter destination name") },
            modifier = Modifier.fillMaxWidth()
        )
        Spacer(Modifier.height(12.dp))
        Button(
            onClick = {},
            modifier = Modifier.fillMaxWidth()
        ) {
            Text("Search Destination")
        }
    }
}

Spacer(Modifier.height(16.dp))

Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(16.dp),
    elevation = CardDefaults.cardElevation(defaultElevation =
4.dp)
) {
    Column(modifier = Modifier.padding(16.dp)) {
        Text(
```

Smart Tourist Safety Monitoring & Incident Response

```
        text = "Trip Status",
        style
MaterialTheme.typography.titleMedium.copy(fontWeight = FontWeight.Bold)
    )
    Spacer(Modifier.height(8.dp))
    Text("Monitoring Inactive")
    Spacer(Modifier.height(8.dp))
    Button(
        onClick = {},
        modifier = Modifier
            .fillMaxWidth(),
        colors = ButtonDefaults.buttonColors(containerColor =
Color.White)
    ) {
        Text("Start Monitoring", color = Color(0xFF16C35B))
    }
}
}

Spacer(Modifier.height(16.dp))

OutlinedButton(
    onClick = {},
    modifier = Modifier.fillMaxWidth(),
    colors = ButtonDefaults.outlinedButtonColors(contentColor =
Color(0xFF149EE7))
) {
    Icon(Icons.Default.Refresh, contentDescription = "Refresh")
    Spacer(modifier = Modifier.width(8.dp))
    Text("Refresh Now")
}
}
}
```

```
@Preview(showBackground = true)
```

```
@Composable
```

```
fun OurTripScreenPreview() {
    OurTripScreen()
}
```

ProfileScreen.kt

```
package com.bytemap.a24172012068_assignment.screen
```

```
import android.widget.Toast
import androidx.compose.foundation.BorderStroke
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Arrangement
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.layout.width
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.CircleShape
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons)
import androidx.compose.material.icons.automirrored.filled.Logout
import androidx.compose.material.icons.filled.CalendarToday
import androidx.compose.material.icons.filled.LocationOn
import androidx.compose.material.icons.filled.Mail
import androidx.compose.material.icons.filled.Person
import androidx.compose.material.icons.filled.Phone
import androidx.compose.material.icons.filled.Security
import androidx.compose.material.icons.filled.TrackChanges
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.Icon
import androidx.compose.material3.OutlinedButton
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
```

```
@Composable
fun ProfileScreen() {
    val context = LocalContext.current
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(Color(0xFFF8F9FA))
            .verticalScroll(rememberScrollState())
            .padding(8.dp)
```

Smart Tourist Safety Monitoring & Incident Response

```
) {
    UserIDCard()
    ContactUI()
    LocationUI()

    Spacer(modifier = Modifier.height(18.dp))

    OutlinedButton(
        modifier = Modifier
            .fillMaxWidth()
            .background(Color.White),
        shape = RoundedCornerShape(12.dp),
        onClick = {
            Toast.makeText(context, "Logged Out",
                Toast.LENGTH_SHORT).show()
        },
        border = BorderStroke(1.dp, Color.Red)
    ) {
        Icon(
            imageVector = Icons.AutoMirrored.Filled.Logout,
            contentDescription = "Logout Icon",
            tint = Color.Red,
            modifier = Modifier
                .size(28.dp)
                .padding(4.dp)
        )
        Text(
            text = "Logout",
            fontSize = 16.sp,
            color = Color.Red,
            modifier = Modifier.padding(8.dp)
        )
    }
}
```

```
@Composable
fun UserIDCard() {
    Card(
        modifier = Modifier
            .padding(8.dp)
            .fillMaxWidth(),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp)
    ) {
```

Smart Tourist Safety Monitoring & Incident Response

```
Column(
    modifier = Modifier
        .padding(32.dp)
        .fillMaxWidth(),
    horizontalAlignment = Alignment.CenterHorizontally,
    verticalArrangement = Arrangement.Center
) {
    Card(
        modifier = Modifier
            .padding(8.dp)
            .align(alignment = Alignment.CenterHorizontally),
        colors      = CardDefaults.cardColors(containerColor =
Color(0xFFE3F2FD)),
        shape = CircleShape,
        elevation = CardDefaults.cardElevation(defaultElevation =
2.dp)
    ) {
        Icon(
            imageVector = Icons.Default.Person,
            contentDescription = "User Icon",
            tint = Color(0xFF2196F3),
            modifier = Modifier
                .size(70.dp)
                .padding(16.dp),
        )
    }
    Spacer(modifier = Modifier.height(16.dp))
    Text(
        text = "USER 1",
        fontSize = 24.sp,
        fontWeight = FontWeight.Bold,
        color = Color.Black
    )
    Spacer(modifier = Modifier.height(8.dp))
    Card(
        shape = RoundedCornerShape(16.dp),
        colors      = CardDefaults.cardColors(containerColor =
Color(0xFFFF5F5F))
    ) {
        Text(
            text = "ID: 46a08ef58a11b7f86bcb9a8d97c4ac95",
            fontSize = 14.sp,
            color = Color.Gray,
            modifier = Modifier.padding(8.dp)
        )
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        }
    }
}

@Composable
fun ContactUI() {
    val contactDetails = listOf(
        Triple("Phone", "+91 9512518403", Icons.Default.Phone),
        Triple("Email", "user1@gmail.com", Icons.Default.Mail),
        Triple("Registered", "Today", Icons.Default.CalendarToday)
    )

    Card(
        modifier = Modifier
            .padding(8.dp)
            .fillMaxWidth(),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp)
    ) {
        Column(
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth(),
        ) {
            Text(
                text = "Contact Information",
                fontSize = 16.sp,
                fontWeight = FontWeight.SemiBold,
            )

            Spacer(modifier = Modifier.height(16.dp))

            contactDetails.forEachIndexed { index, (label, value, icon) ->
                Row(
                    verticalAlignment = Alignment.CenterVertically
                ) {
                    Card(
                        shape = RoundedCornerShape(8.dp),
                        colors = CardDefaults.cardColors(containerColor =
Color(0xFFFF5F5F))
                    ) {
                        Icon(
                            imageVector = icon,
                            contentDescription = "$label Icon",
                            tint = Color.Gray,
                        )
                    }
                }
            }
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

Smart Tourist Safety Monitoring & Incident Response

```
        .fillMaxWidth(),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 4.dp)
    ) {
    Column(
        modifier = Modifier
            .padding(16.dp)
            .fillMaxWidth()
    ) {
        Text(
            text = "Location Settings",
            fontSize = 18.sp,
            fontWeight = FontWeight.SemiBold,
            color = Color.Black
        )
        Spacer(modifier = Modifier.height(16.dp))

        contactDetails.forEach { (label, value, icon) ->
        Row(
            verticalAlignment = Alignment.CenterVertically,
            modifier = Modifier
                .fillMaxWidth()
                .padding(vertical = 8.dp)
        ) {
            Card(
                shape = RoundedCornerShape(8.dp),
                colors = CardDefaults.cardColors(containerColor =
Color(0xFFFF5F5F))
            ) {
                Icon(
                    imageVector = icon,
                    contentDescription = "$label Icon",
                    tint = Color.Gray,
                    modifier = Modifier
                        .size(30.dp)
                        .padding(4.dp)
                )
            }
            Spacer(modifier = Modifier.width(16.dp))

            Column {
                Text(
                    text = label,
                    fontSize = 14.sp,
```

Smart Tourist Safety Monitoring & Incident Response

```
        color = Color.DarkGray
    )
    Text(
        text = value,
        fontSize = 16.sp,
        color = statusColors[value] ?: Color.Black
    )
}
}
}
}
```

```
@Preview(showBackground = true)
```

@Composable

```
fun ProfileScreenPreview() {
```

ProfileScreen()

}

SafetyTipsScreen.kt

package com.bytemap.a24172012068.assignment.screen

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons)
import androidx.compose.material.icons.filled./*
import androidx.compose.material3./*
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.vector.ImageVector
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
```

```
data class SafetyTip(  
    val title: String,  
    val description: String,  
    val icon: ImageVector
```

Smart Tourist Safety Monitoring & Incident Response

```
    val color: Color
)

@Composable
fun SafetyTipsScreen(navController: NavController) {
    val safetyTips = listOf(
        SafetyTip(
            "Stay Alert",
            "Always be aware of your surroundings and trust your instincts. If something feels wrong, it probably is.",
            Icons.Default.Visibility,
            Color(0xFF4CAF50)
        ),
        SafetyTip(
            "Share Location",
            "Keep your location sharing enabled and let trusted contacts know where you're going.",
            Icons.Default.LocationOn,
            Color(0xFF2196F3)
        ),
        SafetyTip(
            "Emergency Contacts",
            "Keep emergency contacts easily accessible and know how to reach them quickly.",
            Icons.Default.Contacts,
            Color(0xFFFF9800)
        ),
        SafetyTip(
            "Avoid Dark Areas",
            "Stick to well-lit, populated areas especially during night time. Plan your routes in advance.",
            Icons.Default.LightMode,
            Color(0xFF9C27B0)
        ),
        SafetyTip(
            "Trust Your Phone",
            "Keep your phone charged and easily accessible. Use emergency features when needed.",
            Icons.Default.Phone,
            Color(0xFFF44336)
        ),
        SafetyTip(
            "Travel in Groups",
            "When possible, travel with friends or family. There's safety in numbers.",
            Icons.Default.Group,
        )
    )
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        Color(0xFF607D8B)
    )
)

Column(
    modifier = Modifier
        .fillMaxSize()
        .background(Color(0xFFF8F9FA))
        .verticalScroll(rememberScrollState())
        .padding(16.dp)
) {
    Card(
        modifier = Modifier.fillMaxWidth(),
        shape = RoundedCornerShape(16.dp),
        colors      =      CardDefaults.cardColors(containerColor      =
Color.White),
        elevation   =   CardDefaults.cardElevation(defaultElevation   =
2.dp)
    ) {
        Column(
            modifier = Modifier.padding(20.dp),
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Icon(
                imageVector = Icons.Default.Security,
                contentDescription = "Safety Icon",
                tint = Color(0xFF4CAF50),
                modifier = Modifier.size(48.dp)
            )
            Spacer(modifier = Modifier.height(12.dp))
            Text(
                text = "Safety Tips",
                fontSize = 24.sp,
                fontWeight = FontWeight.Bold,
                color = Color(0xFF2C3E50)
            )
            Text(
                text = "Essential safety guidelines to keep you
protected",
                fontSize = 14.sp,
                color = Color(0xFF7F8C8D),
                textAlign
                    androidx.compose.ui.text.style.TextAlign.Center
            )
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
Spacer(modifier = Modifier.height(20.dp))

Column(
    verticalArrangement = Arrangement.spacedBy(12.dp)
) {
    safetyTips.forEach { tip ->
        SafetyTipCard(tip = tip)
    }
}

Spacer(modifier = Modifier.height(20.dp))

Card(
    modifier = Modifier.fillMaxWidth(),
    shape = RoundedCornerShape(16.dp),
    colors = CardDefaults.cardColors(containerColor =
Color(0xFFE53E3E)),
    elevation = CardDefaults.cardElevation(defaultElevation =
4.dp)
) {
    Column(
        modifier = Modifier
            .fillMaxWidth()
            .padding(20.dp),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Icon(
            imageVector = Icons.Default.Warning,
            contentDescription = "Emergency Icon",
            tint = Color.White,
            modifier = Modifier.size(32.dp)
        )
        Spacer(modifier = Modifier.height(8.dp))
        Text(
            text = "In Emergency?",
            color = Color.White,
            fontSize = 18.sp,
            fontWeight = FontWeight.Bold
        )
        Text(
            text = "Use the SOS button on the main screen",
            color = Color.White.copy(alpha = 0.9f),
            fontSize = 14.sp
        )
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        }
    }
}

@Composable
fun SafetyTipCard(tip: SafetyTip) {
    Card(
        modifier = Modifier.fillMaxWidth(),
        shape = RoundedCornerShape(12.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 2.dp)
    ) {
        Row(
            modifier = Modifier.padding(16.dp),
            verticalAlignment = Alignment.CenterVertically
        ) {
            Card(
                shape = RoundedCornerShape(8.dp),
                colors = CardDefaults.cardColors(containerColor = tip.color.copy(alpha = 0.1f)),
                modifier = Modifier.size(48.dp)
            ) {
                Box(
                    modifier = Modifier.fillMaxSize(),
                    contentAlignment = Alignment.Center
                ) {
                    Icon(
                        imageVector = tip.icon,
                        contentDescription = "${tip.title} Icon",
                        tint = tip.color,
                        modifier = Modifier.size(24.dp)
                    )
                }
            }
        }

        Spacer(modifier = Modifier.width(16.dp))

        Column(
            modifier = Modifier.weight(1f)
        ) {
            Text(
                text = tip.title,
                fontSize = 16.sp,
                fontWeight = FontWeight.Bold,
                color = Color(0xFF2C3E50)
            )
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        Spacer(modifier = Modifier.height(4.dp))
        Text(
            text = tip.description,
            fontSize = 14.sp,
            color = Color(0xFF7F8C8D),
            lineHeight = 20.sp
        )
    }
}
}

@Preview(showBackground = true)
@Composable
fun SafetyTipsScreenPreview() {
    SafetyTipsScreen(navController =
        androidx.navigation.compose.rememberNavController())
}

SettingsScreen.kt
package com.bytemap.a24172012068_assignment.screen

import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.icons(Icons
import androidx.compose.material.icons.filled.*
import androidx.compose.material3.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.graphics.vector.ImageVector
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController

data class SettingsItem(
    val title: String,
    val subtitle: String,
    val icon: ImageVector,
    val iconColor: Color,
    val onClick: () -> Unit = {}
```

Smart Tourist Safety Monitoring & Incident Response

```
)  
  
@Composable  
fun SettingsScreen(navController: NavController) {  
    var locationEnabled by remember { mutableStateOf(true) }  
    var notificationsEnabled by remember { mutableStateOf(true) }  
    var autoSOSEnabled by remember { mutableStateOf(false) }  
    var darkModeEnabled by remember { mutableStateOf(false) }  
    var biometricEnabled by remember { mutableStateOf(true) }  
  
    Column(  
        modifier = Modifier  
            .fillMaxSize()  
            .background(Color(0xFF5F7FA))  
            .verticalScroll(rememberScrollState())  
    ) {  
        SettingsHeader()  
  
        Spacer(modifier = Modifier.height(16.dp))  
  
        SettingsSection(  
            title = "Safety Settings",  
            items = listOf(  
                SettingsItem(  
                    title = "Location Services",  
                    subtitle = "Share your location for emergency response",  
                    icon = Icons.Default.LocationOn,  
                    iconColor = Color(0xFF2196F3),  
                    onClick = { locationEnabled = !locationEnabled }  
                ),  
                SettingsItem(  
                    title = "Auto SOS",  
                    subtitle = "Automatically send SOS in critical situations",  
                    icon = Icons.Default.Warning,  
                    iconColor = Color(0xFFE53E3E),  
                    onClick = { autoSOSEnabled = !autoSOSEnabled }  
                ),  
                SettingsItem(  
                    title = "Biometric Lock",  
                    subtitle = "Secure app with fingerprint or face unlock",  
                    icon = Icons.Default.Fingerprint,  
                    iconColor = Color(0xFF4CAF50),  
                    onClick = { biometricEnabled = !biometricEnabled }  
                )  
            )  
    }  
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        )
    )
)

Spacer(modifier = Modifier.height(8.dp))

SettingsSection(
    title = "Notifications",
    items = listOf(
        SettingsItem(
            title = "Push Notifications",
            subtitle = "Receive alerts and safety updates",
            icon = Icons.Default.Notifications,
            iconColor = Color(0xFFFF9800),
            onClick = { notificationsEnabled = !notificationsEnabled }
        ),
        SettingsItem(
            title = "Emergency Alerts",
            subtitle = "Critical safety notifications",
            icon = Icons.Default.NotificationImportant,
            iconColor = Color(0xFFE53E3E),
            onClick = { }
        )
    )
)

Spacer(modifier = Modifier.height(8.dp))

SettingsSection(
    title = "Privacy & Security",
    items = listOf(
        SettingsItem(
            title = "Data Privacy",
            subtitle = "Manage your personal data",
            icon = Icons.Default.PrivacyTip,
            iconColor = Color(0xFF9C27B0),
            onClick = { }
        ),
        SettingsItem(
            title = "Account Security",
            subtitle = "Password and security settings",
            icon = Icons.Default.Security,
            iconColor = Color(0xFF607D8B),
            onClick = { }
        )
    )
)
```

Smart Tourist Safety Monitoring & Incident Response

```
        )
    )

    Spacer(modifier = Modifier.height(8.dp))

    SettingsSection(
        title = "App Settings",
        items = listOf(
            SettingsItem(
                title = "Dark Mode",
                subtitle = "Switch between light and dark themes",
                icon = Icons.Default.DarkMode,
                iconColor = Color(0xFF424242),
                onClick = { darkModeEnabled = !darkModeEnabled }
            ),
            SettingsItem(
                title = "Language",
                subtitle = "English",
                icon = Icons.Default.Language,
                iconColor = Color(0xFF2196F3),
                onClick = { }
            ),
            SettingsItem(
                title = "About",
                subtitle = "App version and information",
                icon = Icons.Default.Info,
                iconColor = Color(0xFF757575),
                onClick = { }
            )
        )
    )

    Spacer(modifier = Modifier.height(16.dp))

    LogoutButton()

    Spacer(modifier = Modifier.height(32.dp))
}

}

@Composable
fun SettingsHeader() {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 16.dp, vertical = 8.dp),

```

Smart Tourist Safety Monitoring & Incident Response

```
        shape = RoundedCornerShape(16.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 2.dp)
    ) {
    Row(
        modifier = Modifier
            .fillMaxWidth()
            .padding(20.dp),
        verticalAlignment = Alignment.CenterVertically
    ) {
        Box(
            modifier = Modifier
                .size(60.dp)
                .background(
                    color = Color(0xFF1937B3),
                    shape = RoundedCornerShape(12.dp)
                ),
            contentAlignment = Alignment.Center
        ) {
            Icon(
                imageVector = Icons.Default.Settings,
                contentDescription = "Settings",
                tint = Color.White,
                modifier = Modifier.size(28.dp)
            )
        }
    }

    Spacer(modifier = Modifier.width(16.dp))

    Column {
        Text(
            text = "Settings",
            fontSize = 24.sp,
            fontWeight = FontWeight.Bold,
            color = Color(0xFF2C3E50)
        )
        Text(
            text = "Customize your SafeHorizon experience",
            fontSize = 14.sp,
            color = Color(0xFF7F8C8D)
        )
    }
}
}
```

Smart Tourist Safety Monitoring & Incident Response

```
@Composable
fun SettingsSection(
    title: String,
    items: List<SettingsItem>
) {
    Column(
        modifier = Modifier.padding(horizontal = 16.dp)
    ) {
        Text(
            text = title,
            fontSize = 16.sp,
            fontWeight = FontWeight.SemiBold,
            color = Color(0xFF2C3E50),
            modifier = Modifier.padding(horizontal = 4.dp, vertical =
8.dp)
        )

        Card(
            shape = RoundedCornerShape(16.dp),
            colors = CardDefaults.cardColors(containerColor =
Color.White),
            elevation = CardDefaults.cardElevation(defaultElevation =
2.dp)
        ) {
            Column {
                items.forEachIndexed { index, item ->
                    SettingsItemRow(
                        item = item,
                        showDivider = index < items.size - 1
                    )
                }
            }
        }
    }
}

@Composable
fun SettingsItemRow(
    item: SettingsItem,
    showDivider: Boolean
) {
    Column {
        Row(
            modifier = Modifier
                .fillMaxWidth()
                .padding(16.dp),
            verticalArrangement = Arrangement.Center
        ) {
            item.icon
            Text(item.title)
            if (showDivider) {
                Spacer(modifier = Modifier.width(16.dp))
                Divider(
                    thickness = 1.dp,
                    color = Color(0xFF2C3E50)
                )
            }
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
        verticalAlignment = Alignment.CenterVertically
    ) {
    Box(
        modifier = Modifier
            .size(40.dp)
            .background(
                color = item.iconColor.copy(alpha = 0.1f),
                shape = RoundedCornerShape(8.dp)
            ),
        contentAlignment = Alignment.Center
    ) {
        Icon(
            imageVector = item.icon,
            contentDescription = item.title,
            tint = item.iconColor,
            modifier = Modifier.size(20.dp)
        )
    }
}

Spacer(modifier = Modifier.width(16.dp))

Column(
    modifier = Modifier.weight(1f)
) {
    Text(
        text = item.title,
        fontSize = 16.sp,
        fontWeight = FontWeight.Medium,
        color = Color(0xFF2C3E50)
    )
    Text(
        text = item.subtitle,
        fontSize = 13.sp,
        color = Color(0xFF7F8C8D)
    )
}

Icon(
    imageVector = Icons.Default.ChevronRight,
    contentDescription = "Navigate",
    tint = Color(0xFFBDC3C7),
    modifier = Modifier.size(20.dp)
)
}

if (showDivider) {
```

Smart Tourist Safety Monitoring & Incident Response

```
        HorizontalDivider(
            modifier = Modifier.padding(horizontal = 16.dp),
            color = Color(0xFFE5E7EB)
        )
    }
}

@Composable
fun LogoutButton() {
    Card(
        modifier = Modifier
            .fillMaxWidth()
            .padding(horizontal = 16.dp),
        shape = RoundedCornerShape(16.dp),
        colors = CardDefaults.cardColors(containerColor = Color.White),
        elevation = CardDefaults.cardElevation(defaultElevation = 2.dp)
    ) {
        Row(
            modifier = Modifier
                .fillMaxWidth()
                .padding(16.dp),
            verticalAlignment = Alignment.CenterVertically
        ) {
            Box(
                modifier = Modifier
                    .size(40.dp)
                    .background(
                        color = Color(0xFFE5E3E).copy(alpha = 0.1f),
                        shape = RoundedCornerShape(8.dp)
                    ),
                contentAlignment = Alignment.Center
            ) {
                Icon(
                    imageVector = Icons.Default.ExitToApp,
                    contentDescription = "Logout",
                    tint = Color(0xFFE5E3E),
                    modifier = Modifier.size(20.dp)
                )
            }
        }

        Spacer(modifier = Modifier.width(16.dp))

        Column(
            modifier = Modifier.weight(1f)
        ) {
```

Smart Tourist Safety Monitoring & Incident Response

```
        Text(
            text = "Logout",
            fontSize = 16.sp,
            fontWeight = FontWeight.Medium,
            color = Color(0xFFE53E3E)
        )
        Text(
            text = "Sign out of your account",
            fontSize = 13.sp,
            color = Color(0xFF7F8C8D)
        )
    }
}

Icon(
    imageVector = Icons.Default.ChevronRight,
    contentDescription = "Navigate",
    tint = Color(0xFFBDC3C7),
    modifier = Modifier.size(20.dp)
)
}
}
}
}
```

```
@Preview(showBackground = true)
@Composable
fun SettingsScreenPreview() {
    MaterialTheme {
        SettingsScreen(navController =
    androidx.navigation.compose.rememberNavController())
    }
}
```

TripMonitorScreen.kt

```
package com.bytemap.a24172012068_assignment.screen
```

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material3.*
import androidx.compose.material3.TabRowDefaults.tabIndicatorOffset
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
```

Smart Tourist Safety Monitoring & Incident Response

```
import androidx.navigation NavController
import androidx.navigation.compose.rememberNavController

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun TripMonitorScreen(navController: NavController) {
    var selectedTabIndex by remember { mutableStateOf(0) }
    val tabs = listOf("Our Trip", "Activities")

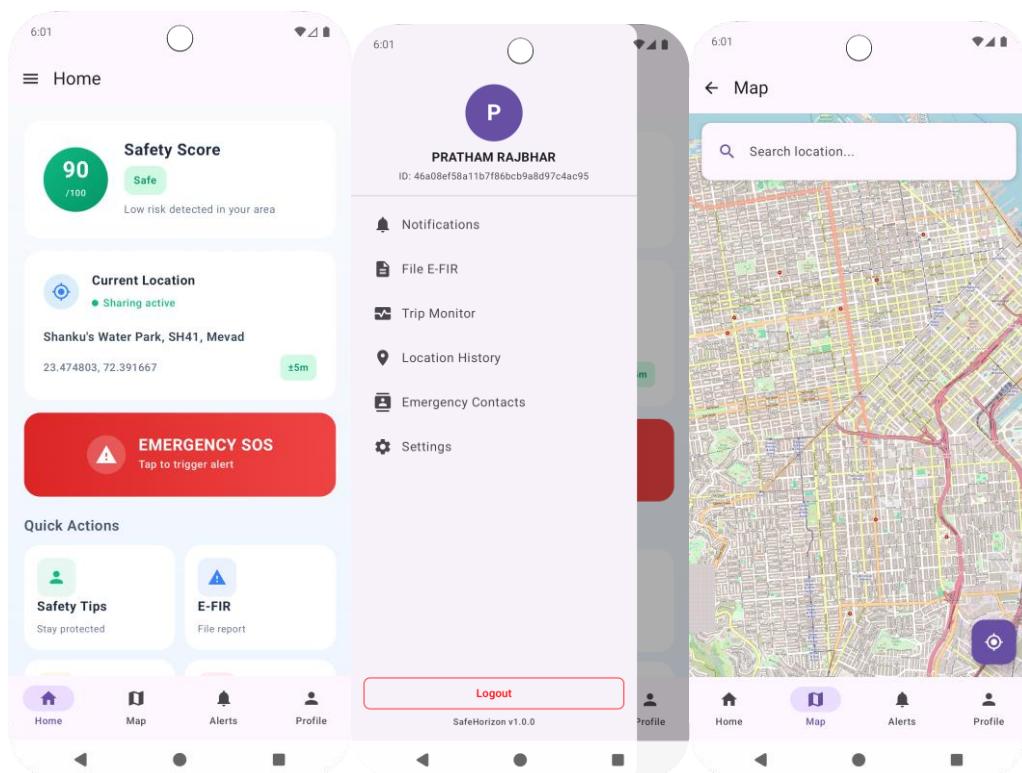
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(MaterialTheme.colorScheme.background)
    ) {
        TabRow(
            selectedTabIndex = selectedTabIndex,
            containerColor = MaterialTheme.colorScheme.surface,
            contentColor = MaterialTheme.colorScheme.primary,
            divider = { Divider(color = Color.LightGray, thickness = 1.dp) }
        ),
        indicator = { tabPositions ->
            TabRowDefaults.Indicator(
                Modifier.tabIndicatorOffset(tabPositions[selectedTabIndex]),
                color = MaterialTheme.colorScheme.primary
            )
        }
    ) {
        tabs.forEachIndexed { index, title ->
            Tab(
                selected = selectedTabIndex == index,
                onClick = { selectedTabIndex = index },
                text = {
                    Text(
                        text = title,
                        fontWeight = if (selectedTabIndex == index)
                            FontWeight.Bold else FontWeight.Normal,
                        color = if (selectedTabIndex == index)
                            MaterialTheme.colorScheme.primary
                        else
                            Color.Gray
                    )
                }
            )
        }
    }
}
```

Smart Tourist Safety Monitoring & Incident Response

```
Box(  
    modifier = Modifier  
        .fillMaxSize()  
        .padding(16.dp),  
    contentAlignment = Alignment.Center  
) {  
    when (selectedTabIndex) {  
        0 -> OurTripScreen()  
        1 -> ActivitiesScreen()  
    }  
}  
}  
}
```

```
@Preview(showBackground = true)  
@Composable  
fun TripMonitorScreenPreview() {  
    TripMonitorScreen(navController = rememberNavController())  
}
```

Output :



Smart Tourist Safety Monitoring & Incident Response

The image displays the Smart Tourist Safety Monitoring & Incident Response application interface, featuring several key screens:

- Alerts Screen:** Shows four alerts: Heavy Snowfall Warning, Wildfire Alert, Health Advisory, and Traffic Accident Alert.
- Profile Screen:** Displays user information for **USER 1**, including ID (46a08ef58a11b7f86bcb9a8d97c4ac95), contact details (Phone: +91 9512518403, Email: user1@gmail.com), and registration date (Today).
- File E-FIR Screen:** Allows users to file an incident report. It includes sections for Incident Type (Harassment, Theft, Assault, Fraud, Emergency, Other), Other incidents, Incident Description (with a text input field and character count), Select Date & Time, Location (with a checkbox for Use current location), and a map view.
- Trip Monitor Screen:** Shows the current trip status with the address: Shanku's Water Park, SH41, Coordinates: 23.474803, 72.391667, Speed: 0.0 km/h, and Updates: 1 auto-updates completed. It also includes fields for setting a destination and monitoring status.
- Emergency Contacts Screen:** Lists contacts: PAPA (8745214752, Parent) and MOTHER (7547862145, Parent).
- Settings Screen:** Provides options for Safety Settings (Location Services, Auto SOS, Biometric Lock), Notifications (Push Notifications, Emergency Alerts), and Privacy & Security (Data Privacy).