

# Android Programming

Meruyert Oshayeva

## Hi, Teacher!

Here is my practice task 4

In this task I made a Story Carousel like in Instagram

```
Usage
@Composable
fun StoriesCarousel(isDarkMode: Boolean, onStoryClick: (Int) -> Unit) {
    LazyRow(
        modifier = Modifier.fillMaxWidth().padding(vertical = 8.dp),
        contentPadding = PaddingValues(horizontal = 12.dp),
        horizontalArrangement = Arrangement.spacedBy(space = 10.dp)
    ) {
        items(
            items = storyList,
            key = { it.id }
        ) { story ->
            StoryItem(story, isDarkMode, onStoryClick)
        }
    }
}
```

I made sure to view stories in full screen for a better experience

```

@Composable
fun FullScreenStoryViewer(
    stories: List<Story>,
    initialStoryId: Int,
    onClose: () -> Unit
) {
    val scope = rememberCoroutineScope()
    val slideDuration = 1000
    val initialStoryIndex = stories.indexOfFirst { it.id == initialStoryId }.coerceAtLeast(minimumValue = 0)

    var currentStoryIndex by rememberSaveable { mutableStateOf(value = initialStoryIndex) }
    var currentSlideIndex by rememberSaveable { mutableStateOf(value = 0) }

    val safeCurrentStory = stories.getOrNull(currentStoryIndex) ?: run { onClose(); return }
    val currentStorySlides = safeCurrentStory.storyContentIds.size
    val currentSlideContentId = safeCurrentStory.storyContentIds.getOrNull(currentSlideIndex) ?: run { onClose(); return }

    val progressAnimatable = remember { Animatable(initialValue = 0f) }

    val navigateStory: (Boolean) -> Unit = { isForward ->
        scope.launch {
            progressAnimatable.stop()
        }
    }
}

```

You can also view next story or go back by tapping the right and left side accordingly:

```

if (isForward) {
    if (currentSlideIndex < currentStorySlides - 1) {
        currentSlideIndex += 1
    } else if (currentStoryIndex < stories.size - 1) {
        currentStoryIndex += 1
        currentSlideIndex = 0
    } else {
        onClose()
    }
} else {
    if (currentSlideIndex > 0) {
        currentSlideIndex -= 1
    } else if (currentStoryIndex > 0) {
        currentStoryIndex -= 1
        currentSlideIndex = stories[currentStoryIndex].storyContentIds.size - 1
    }
}
}

```

The next thing I added is follower list, by clicking its icon, a list will appear where you can unfollow or remove those people

```

Column(Modifier.animateContentSize()) {
    followers.forEach { follower ->
        val isBeingRemoved = removedFollowerIds.value.contains(follower.id)

        val onRemoveFollower: () -> Unit = {
            scope.launch {
                val removedId = follower.id
                removedFollowerIds.value = removedFollowerIds.value + removedId

                delay( timeMillis = 300)

                followers = followers.filter { it.id != removedId }
                removedFollowerIds.value = removedFollowerIds.value - removedId

                val result = snackbarHostState.showSnackbar(
                    message = "${follower.name} removed.",
                    actionLabel = "Undo",
                    duration = SnackbarDuration.Short
                )

                if (result == SnackbarResult.ActionPerformed) {
                    followers = followers.plus( element = follower).sortedBy { it.id }
                }
            }
        }
    }
}

```

And I also added the close button, so you can hide the list when not needed

```

Text(
    text = "Followers (${followers.size})",
    style = MaterialTheme.typography.titleMedium,
    fontWeight = FontWeight.Bold,
    color = textBaseColor,
    modifier = Modifier.padding(start = 8.dp)
)
IconButton(onClick = onHideFollowers) {
    Icon(
        imageVector = Icons.Default.Close,
        contentDescription = "Hide Followers List",
        tint = textBaseColor
    )
}

```

I also tried to animate it a little, as you can see:

```

AnimatedVisibility(
    visible = !isBeingRemoved,
    enter = fadeIn( animationSpec = tween( durationMillis = 150)),
    exit = fadeOut( animationSpec = tween( durationMillis = 300)) + shrinkVertically( animationSpec = tween( durationMillis = 300))
) {
    Column {
        FollowerItem(
            follower = follower,
            isDarkMode = isDarkMode,
            onRemove = onRemoveFollower
        )
        HorizontalDivider(color = Color.Gray.copy(alpha = 0.2f), modifier = Modifier.padding(horizontal = 16.dp))
    }
}

```

And the last thing I added is this box in Sections Drawer, so it doesn't look as blank as in the previous code

```
Icon(  
    imageVector = Icons.Default.Star,  
    contentDescription = "Profile Star",  
    tint = Color.White,  
    modifier = Modifier.size( size = 32.dp)  
)  
Spacer(modifier = Modifier.height( height = 8.dp))  
Text(  
    text = "M's Profile Navigator",  
    style = MaterialTheme.typography.titleMedium,  
    fontWeight = FontWeight.ExtraBold,  
    color = Color.White  
)  
}
```

And I also made sure to use LazyColumn, as you see in this part (and LazyRow is used in Story carousel, if you didn't notice):

```

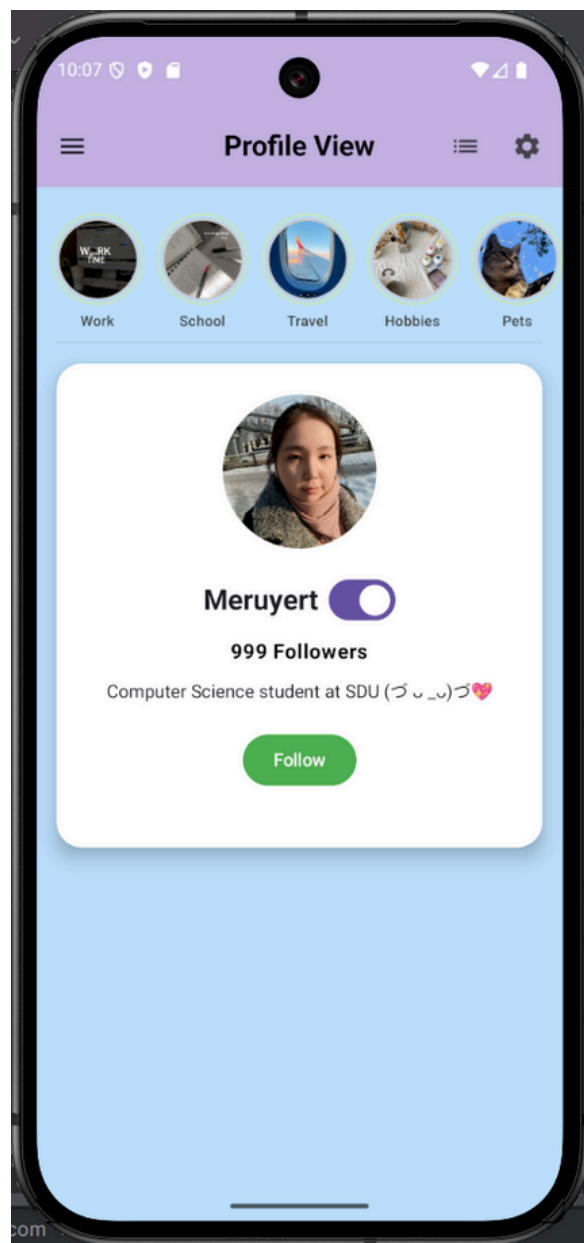
LazyColumn(
    modifier = modifier.fillMaxSize(),
    contentPadding = PaddingValues(vertical = 12.dp),
    state = lazyListState
) {
    item(key = "StoriesCarousel") {
        StoriesCarousel(isDarkMode = isDarkMode, onStoryClick = onStoryClick)
        HorizontalDivider(color = Color.Gray.copy(alpha = 0.2f), modifier = Modifier.padding(horizontal = 16.dp))
        Spacer(modifier = Modifier.height( height = 16.dp))
    }

    item(key = "ProfileHeader") {
        Card(
            modifier = Modifier
                .fillMaxWidth()
                .padding(horizontal = 16.dp)
                .padding(bottom = 12.dp),
            shape = RoundedCornerShape( size = 20.dp),
            elevation = CardDefaults.cardElevation( defaultElevation = cardElevation),
            colors = CardDefaults.cardColors(containerColor = cardColor)
        ) {
            Column(
                horizontalAlignment = Alignment.CenterHorizontally,
                modifier = Modifier.padding( all = 24.dp)
            ) {
                Image(
                    painter = painterResource(id = R.drawable.ic_avatar),
                    contentDescription = "Profile Image",
                    modifier = Modifier

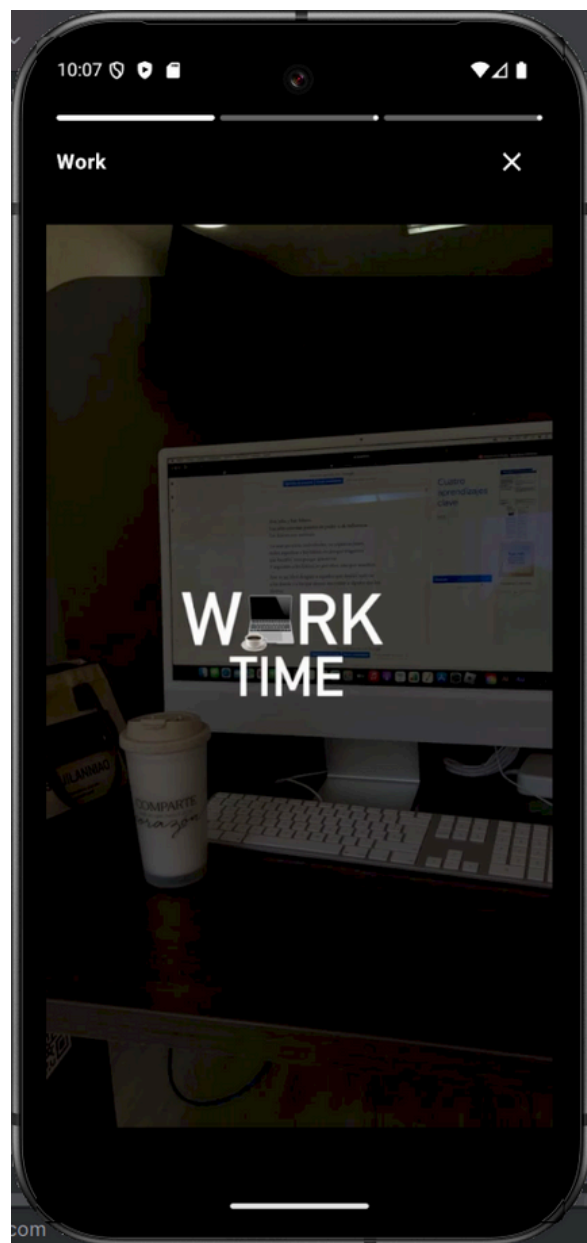
```

I tried to make the swipe to remove actions, but for some reason it gave me tons of errors no matter what, seemed like issue with the material3 packages itself, so instead I made sure to include hide the follower list action, animated it and tried to make the stories better and more functional.

Here is the result:

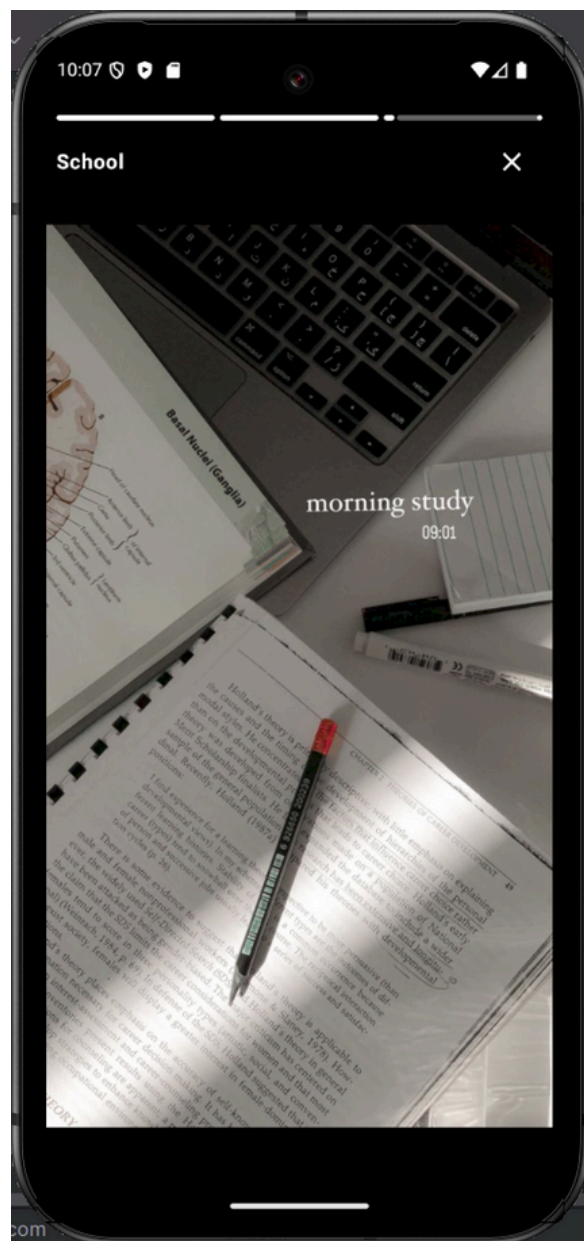


You can already see the previews for stories, and here is how it looks like when you click them

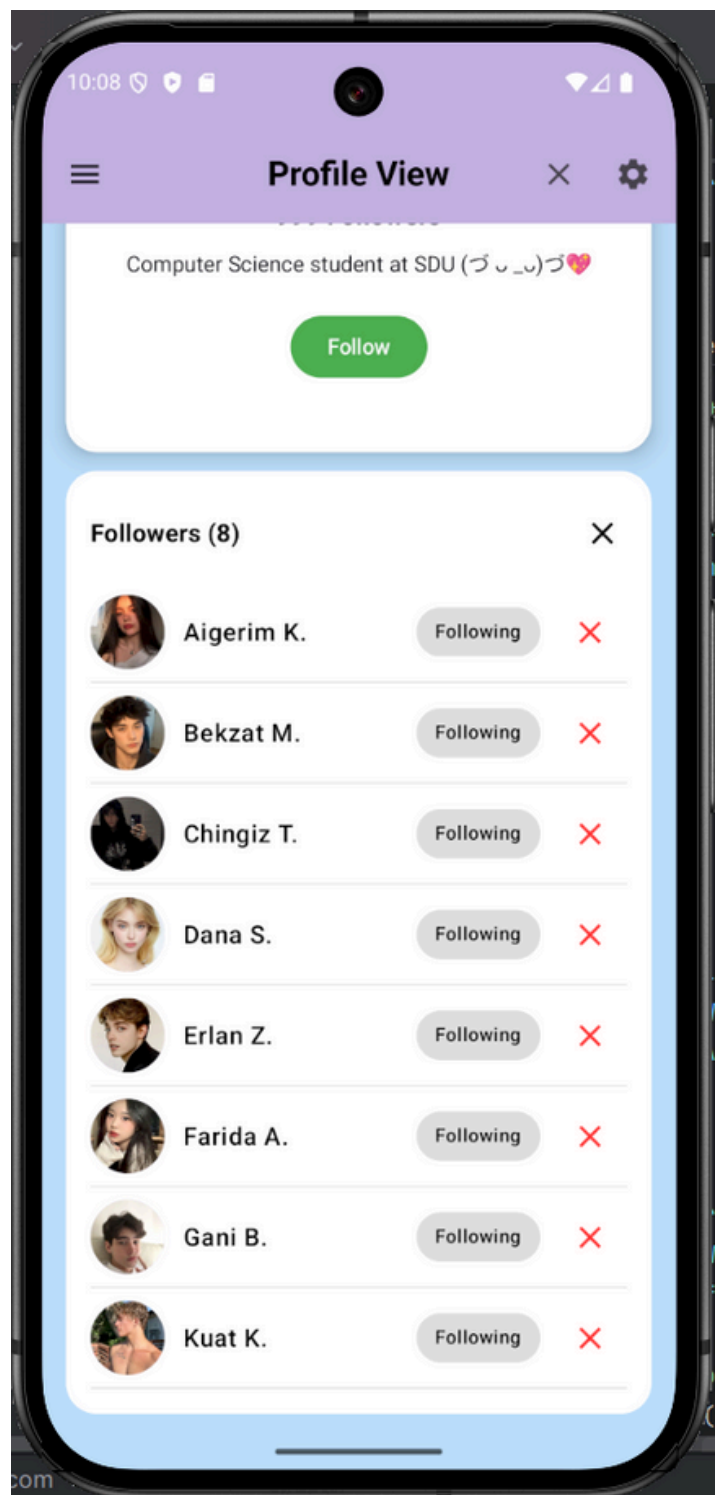


You can also one tap and go to the next slide, so each 3 seconds it'll automatically go to the next slide, the white line at the top is also animated so you can see after what time it'll change to another picture or it'll close by itself when you reach the end of all stories, and most importantly I included close button so you can exit whenever you want

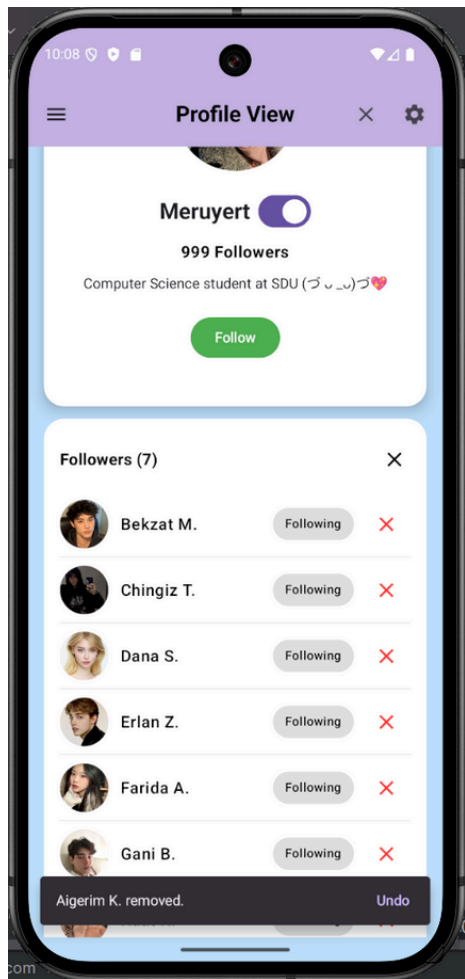




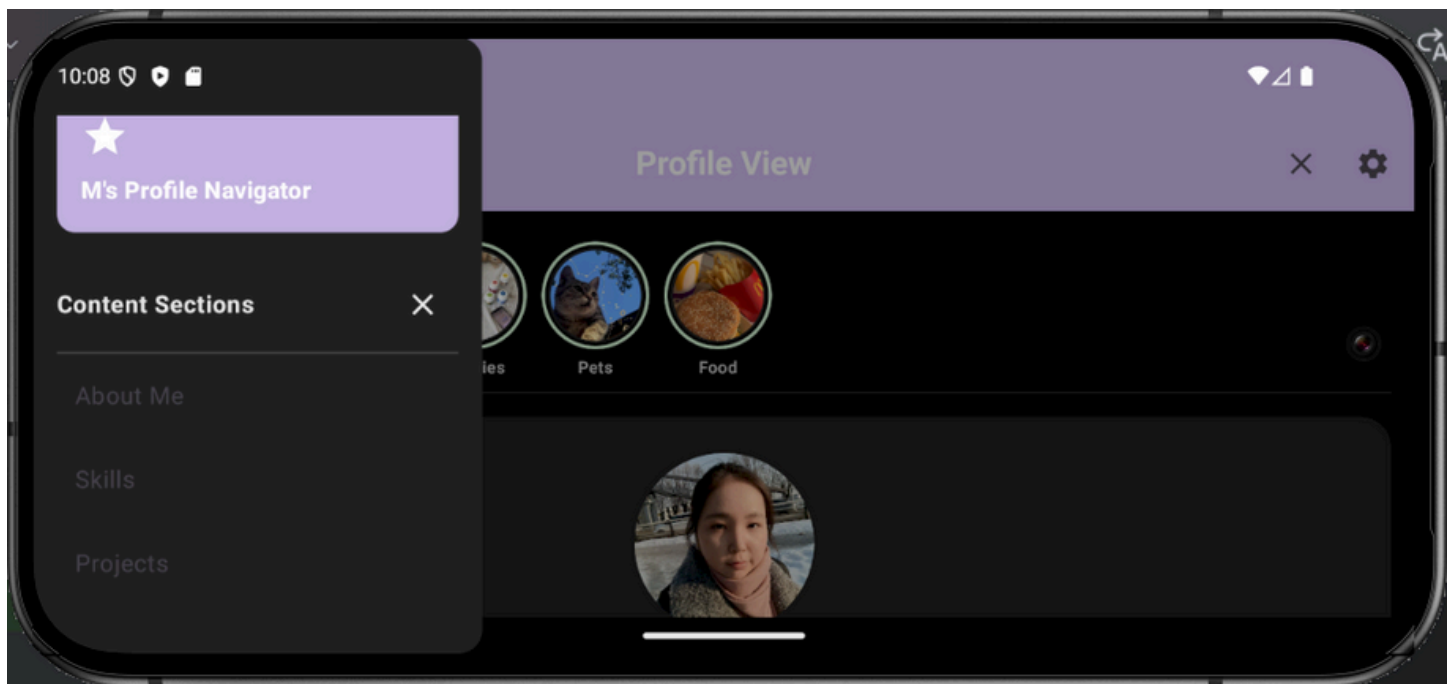
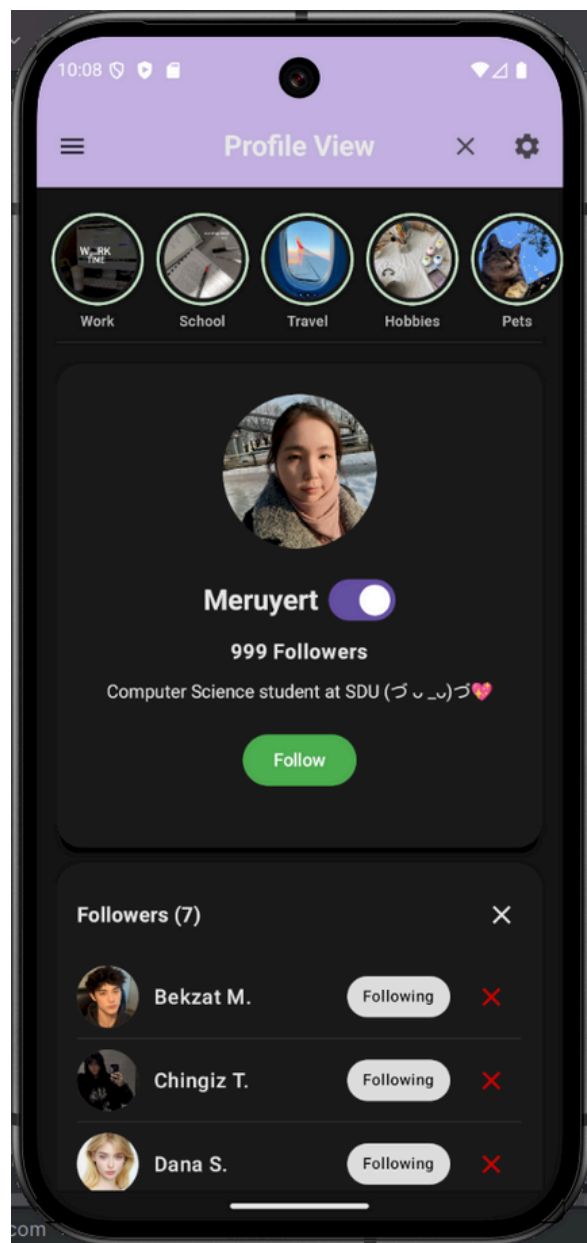
Here is the follower list, you can access it by clicking the list button near settings



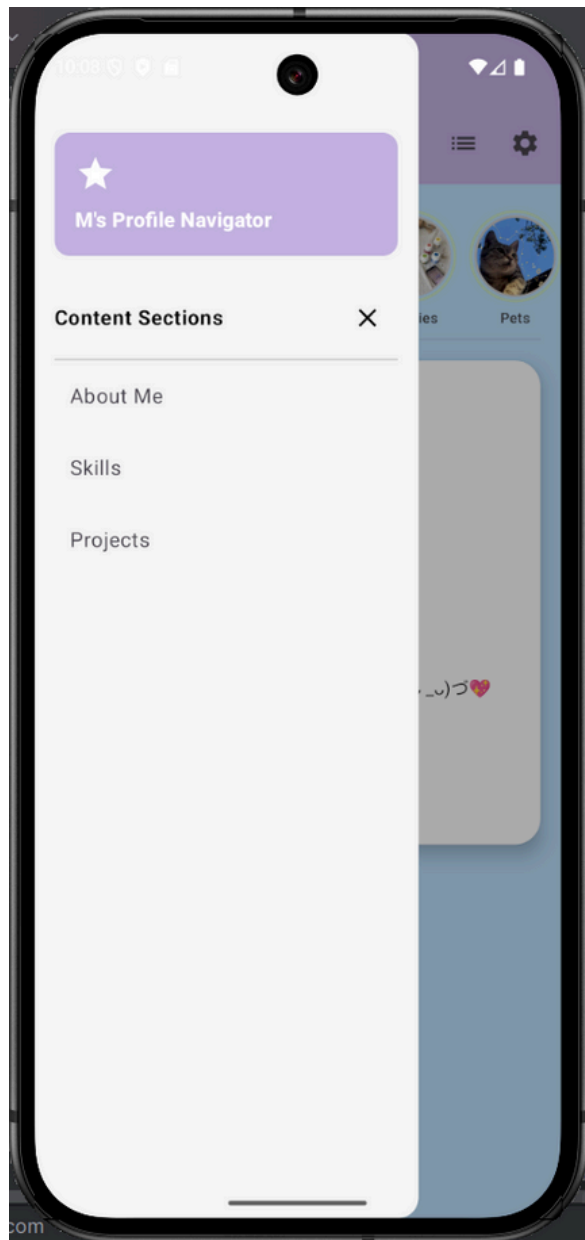
You can also hide it using the “X” button near the “Followers (7).....” or at the top bar, it also tells you when you remove someone from your followers list and show the option “Undo” in case if you accidentally pressed it



It also works in Dark Mode and even when you rotate it, you can still scroll through the main page and drawer:



And this is the box I told you earlier, so now it looks more pleasant to eyes when you open the drawer



And that's all that I did for this week's practice task.  
Thank you!