Demo Sheet

Please use this form to plan and organise your demo to make sure you are making the most of the time. Please provide your marker with a completed copy at the start of the presentation – it will not be directly marked but will help make the marking process more quicker and more transparent.

# Context

Provide the context of your application – what problem you are trying to solve and an overview of how your application solves it.

The Glow Up Health and Wellbeing App is designed to support Sustainable Development Goal (SDG) 3: ensuring healthy lives and promoting well-being for all ages. The app aims to strengthen health systems by addressing key health challenges, including maternal and child health, disease prevention and mental well-being.

Many individuals struggle with maintaining a healthy lifestyle due to a lack of structured guidance and accessible resources. Glow Up serves as a digital health companion, offering users insights, progress tracking, and expert-backed resources to encourage and support healthier lifestyle choices. Through interactive features, the app promotes physical and mental wellness, empowering users to make informed health decisions.

# Demo of working app

Use the table below to list each feature you have developed, the UI elements and the ‘working parts’ – backend/hardware/systems apps… It is recommended to list them in an order that makes sense for a walk-through of your app during the demo.

|  |  |  |
| --- | --- | --- |
| Feature | UI elements | Working parts |
| Images Imported and used for all pages | Logo, Navbar buttons, Trackers, Stories, Quotes, etc. | Importing Images and Using them in the App |
| Top navbar with logo, settings and account icon | Logo, Settings, Account icons | Functional buttons that go to their respective pages |
| Bottom navbar with selectable buttons that change state when selected | Buttons with Captions | On click of button goes to their respective page and changes the state of the button to show which page you are on |
| Carousel with alternating images every 4 seconds | Carousel, Quotes, Icons | Automatically changing carousel with smooth animation every 4 seconds. Can be manually dragged to the next |
| Trackers form that tracks data and inserts into database | Colourful PNG Images functioning as buttons, Side scroll. Form to add data to database | Trackers go their respective tracker pages and can be scrollable to the side to access all trackers. Data can be inputted through form. When submitting the form, user feedback is given that data has been saved to database |
| Accessibility Options | Toggle buttons, dropdown | Dark mode changes screen to black and white, high contrast mode makes text colours darker to make them more visible. Font size can be adjusted to make content larger and more readable. Enable button animations is on by default for the buttons but can be turned off if old Android phones do not support this feature |
| Tracker data displayed using data from database | Side scroll, Today’s Date, Bar Chart displaying tracker data for this week with title, scale and legend. Tips shown as tabs with icons | Takes data from the logged in user for each of the trackers and shows this week’s data on the bar chart. Averages are calculated for users to compare their progress |
| Bar Charts displaying data from each tracker | Colour-coded, Titles, Legends, Scale | Displays this week’s data as a data chart to make progress more visually appealing using data taken from trackers in database |
| Community Chat enabling users to send messages by inserting into database | Icons, Textbox, Dates/ Times | Logged in user is able to send messages to the community chat, it displays their name, what they said, the date and time it was sent. Creating and sending a message is inserting it into the database and displaying as a chat message. All users can see everyone’s message. Messages can only be removed by the creator. |
| Google Maps location of events | Google Maps Map, Form, Colour-coded by activity type | Events can be created and will create event in database to store longitude and latitude points for the pin of the map for different events. Data for the events are inserted into Firestore database using form. They are clickable and will automatically move to the location of the event. They can be removed by the creator and after the event ends, the event will be automatically removed. |
| Daily/ Weekly/ Monthly Challenges from database | Progress bar, Timer, Icons | Daily/ Weekly/ Monthly Challenges gives points on completion. They have their own timer, however, have not been fully implemented since this is more complex feature. Challenges database has been (NOT FULLY OPERATIONAL). |
| Monthly Leaderboard showing points of Top 3 users for this month | Icons, Colour-coding | Monthly Leaderboard shows the Top 3 users, their names and their points gained this month. |
| Rewards Page that redeems gift cards and logs transactions in database | Pop up boxes confirming user input | Rewards Page logs transactions in the Firestore database. Based on the user’s points, they can redeem gift cards and rewards. Points are deduced and code is sent to the user through email. |
| Login and Register creates user account | User Feedback success/ failure messages | Login, Register pages creates an email and password, generates unique user UID and inserted into Firebase Authentication database |
| Forms during registration stores user details and their goals in database | Forms, Colourful background | Forms during registration process prompts the user to add user details and their goals which are stored in Firestore database |
| Profile Page gives options to log out, delete account, update user details and goals, update their password | Forms, Colourful background, prompts to confirm action | Profile Page gives options to log out, delete account, update user details, update goals and update their password. This will send an email verification link to reset your password in the Firestore authentication. |

# Other considerations

Provide examples below of how you have considered/implemented the required aspects. 2 or 3 examples should be enough to demonstrate your knowledge and skills.

## Design Guidelines

1. Follows Material 3 standards, ensuring a consistent and modern UI design using Scaffold, App Bar, BottomNavigationBar and Cards.
2. Implemented a Bottom Navigation Bar to maintain smooth and predictable navigation between key sections.
3. Buttons, forms and interactive elements provide animations, tap highlights and real-time feedback to enhance the user experience.

## Accessibility

1. Users can toggle dark mode and high contrast settings via Shared Preferences, making the app more accessible.
2. All text and UI elements adjust properly when large text size settings are enabled, preventing overflow issues.
3. Implemented widgets to provide descriptive labels for buttons and interactive elements.

## Globalisation

1. The app uses clear and straightforward language to ensure that users can easily understand the content, regardless of their background.
2. The app is designed with simple navigation so users can easily find features, such as trackers, community chat and rewards.
3. Images and icons are used throughout the app to enhance understanding and provide a visual context, making the app more accessible to all users.

## Evidence of Testing

1. User tests, focusing on navigation, ease of use and feature accessibility. Users reported issues with challenges not working and some inconsistencies.
2. Verified that all Trackers correctly insert and retrieve data from Firestore, with bar charts reflecting accurate statistics
3. Identified and resolved chart overflow issues.

# Other Information

Use the space below to list anything else that you want to make sure you cover during the demo.

Material 3 Guidelines – The app uses Android Material Design 3 (Scaffold, App Bar, Cards, Navigation Bars) to maintain a modern and consistent look.

Stateless Widget - Used for **static UI components**, such as displaying user information and visual elements that don’t require state changes.

Stateful Widget – Used for dynamic elements like the navigation bar, trackers, chat messages, and interactive forms where user input changes are necessary.