Mobile Web Application Development Portfolio Submission

Student Name, Surname: Mohammed Adnan Osman

Id: 33114153

Course: Computer Science

Module name: Mobile Web Application Development

Module Leader: Dr. Cain Kazimoglu

Word Count: 7994

Video Component Link: https://youtu.be/j2JcHil00XE

Table of Contents

Chapter 1: Introduction

This report covers seven mobile web application tasks, applying techniques learned from lectures, seminars, and independent research. Each app uses XML for layouts and Java for coding in Android Studio, with Figma for wireframes. The report documents processes, challenges, and solutions. All tasks were tested for functionality, meeting assignment requirements and occasionally exceeding them. Gemini assistance in comments . ChatGPT was used, with specific areas of its contribution outlined in the report. I sought to go above and beyond in these tasks :

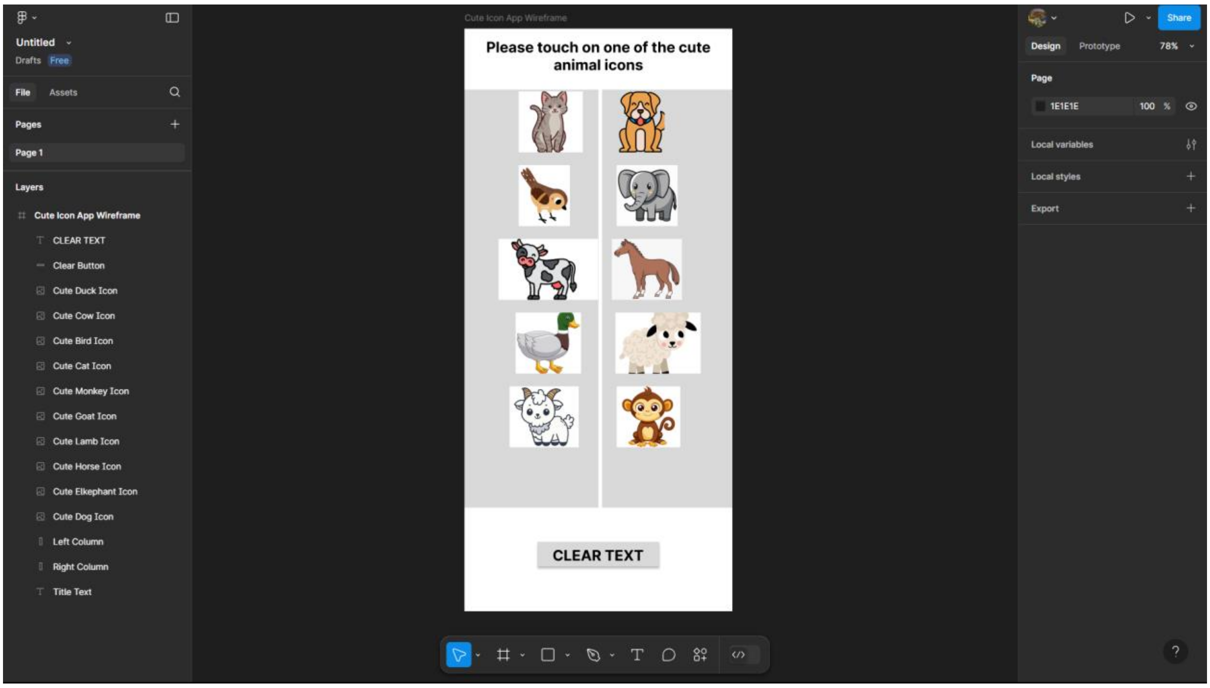
1. Cute Icons : Shake animations , Real–World animal sound effects
2. Gym Guide : Dynamic diagrams
3. Man City Merchandise Ordering : -
4. Fab 4 in Test Cricket : External-Intent
5. CryptoStock Login : -
6. Athlete Connect Contacts : -
7. TravelNow : -

Chapter 2: Wireframes and User Experience

2.1

Task 1: Cute Icons (Wireframe)

The wireframe consists of a two-column layout where the animals are displayed symmetrically opposite each other with 5 in each column. A header at the top provides clear instructions, while the ‘cleat text’ button at the bottom allowed users to go to the default page. This wireframe was essential for pre-development planning, ensuring a clear layout to keep in mind.

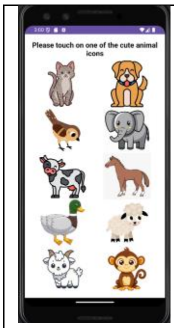
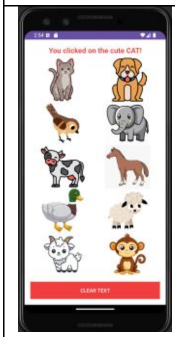
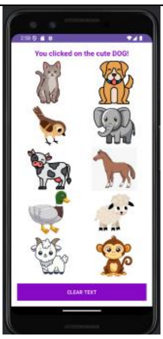
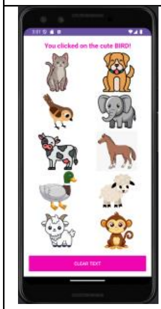
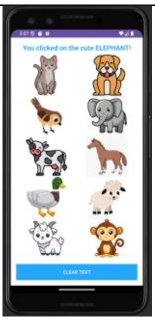
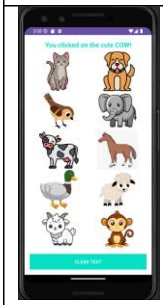
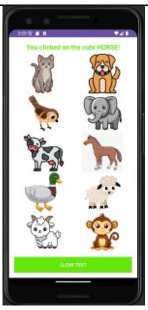


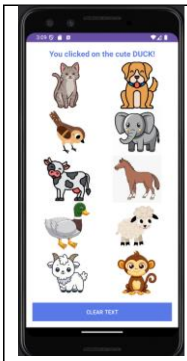
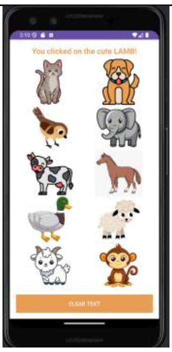
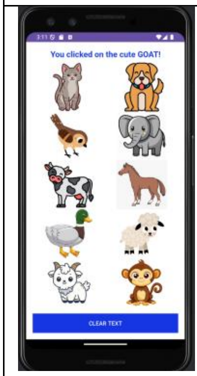
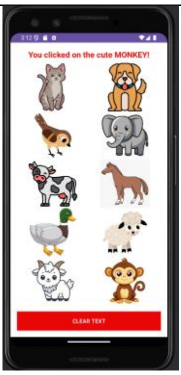
Please touch on one of the cute animal icons

Figure 1a: Main = T1

Task 1: Cute Icons (User Experience)

Users can click on animal icons which changes the colour and text, providing immediate visual feedback. The hidden reset button became visible after interactions, clicking it goes back to the main screen. To make the app more engaging, I added a shake animation and real-animal sound once the icons were clicked. These features helped to enhance user experience by making the app more engaging .

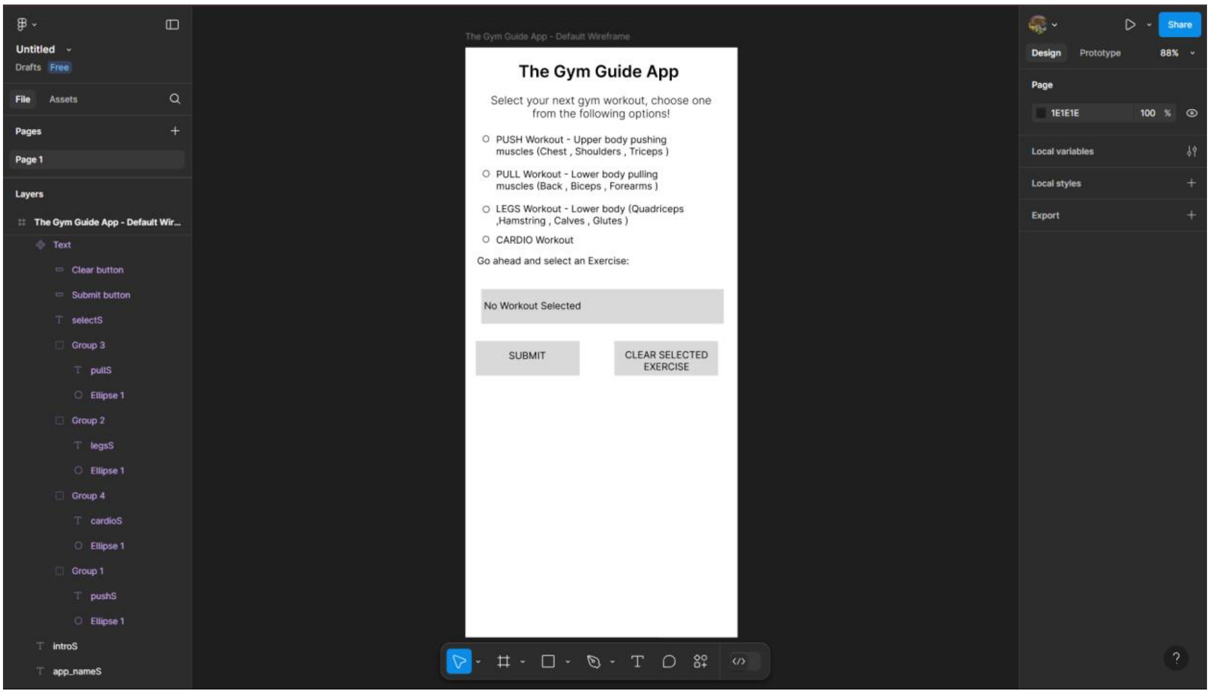
 Please touch on one of the cute animal icons *Figure 2: Main T1*	
 You clicked on the cute CAT! CLEAR TEXT *Figure 3:Cat Icon*	 You clicked on the cute DOG! CLEAR TEXT *Figure 4: Dog Icon*
 You clicked on the cute BIRD! CLEAR TEXT *Figure 5: Bird Icon*	 You clicked on the cute ELEPHANT! CLEAR TEXT *Figure 6: Elephant Icon*
 You clicked on the cute COW! CLEAR TEXT *Figure 7: Cow Icon*	 You clicked on the cute HORSE! CLEAR TEXT *Figure 8: Horse Icon*

 You clicked on the cute DUCK! *Figure 9: Duck Icon*	 You clicked on the cute LAMB! *Figure 10: Lamb Icon*
 You clicked on the cute GOAT! *Figure 11: Goat Icon*	 You clicked on the cute MONKEY! *Figure 12: Monkey Icon*

2.2

Task 2: The Gym Guide App (Wireframe)

This wireframe contains radio buttons for selecting workout splits, a spinner that dynamically updates the exercise list based on the selected split, and a section to display exercise diagrams below the buttons. The two buttons are also included to submit the selected exercise or to clear it and choose again.



The Gym Guide App

Select your next gym workout, choose one from the following options!

○ PUSH Workout - Upper body pushing muscles (Chest , Shoulders , Triceps)

○ PULL Workout - Lower body pulling muscles (Back , Biceps , Forearms)

○ LEGS Workout - Lower body (Quadriceps ,Hamstring , Calves , Glutes)

○ CARDIO Workout

Go ahead and select an Exercise:

No Workout Selected

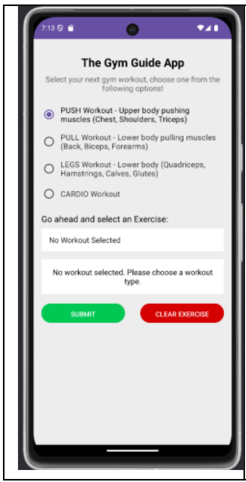
SUBMIT CLEAR SELECTED EXERCISE

Figure 13: Main – T2

Task 2: The Gym Guide (User Experience)

Selecting a workout split with radio buttons updates the spinner dynamically, displaying the corresponding lists of exercises. Once selected, a toast message confirms the chosen exercise, and the app also displays visual diagrams for each exercise. These dynamic diagrams help the user understand how to do the exercise , adding clarity to the app.

Table 1: PUSH



The Gym Guide App

Select your next gym workout, choose one from the following options!

● PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps)

○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms)

○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes)

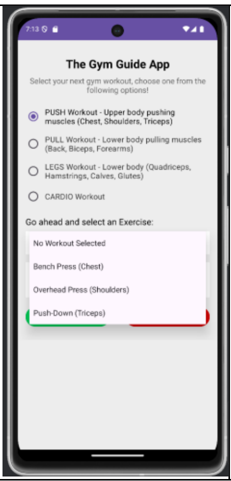
○ CARDIO Workout

Go ahead and select an Exercise:

No Workout Selected

No workout selected. Please choose a workout type.

SUBMIT CLEAR EXERCISE



The Gym Guide App

Select your next gym workout, choose one from the following options!

● PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps)

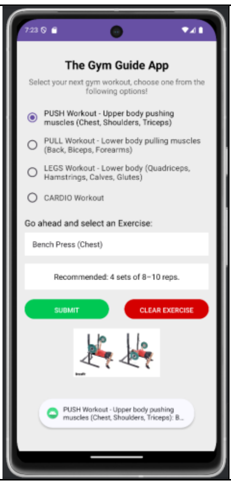
○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms)

○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes)

○ CARDIO Workout

Go ahead and select an Exercise:

No Workout Selected
Bench Press (Chest)
Overhead Press (Shoulders)
Push-Down (Triceps)



The Gym Guide App

Select your next gym workout, choose one from the following options!

● PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps)

○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms)

○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes)

○ CARDIO Workout

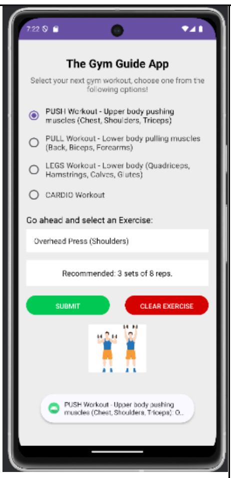
Go ahead and select an Exercise:

Bench Press (Chest)

Recommended: 4 sets of 8–10 reps.

SUBMIT CLEAR EXERCISE

PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps): B...



The Gym Guide App

Select your next gym workout, choose one from the following options!

● PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps)

○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms)

○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes)

○ CARDIO Workout

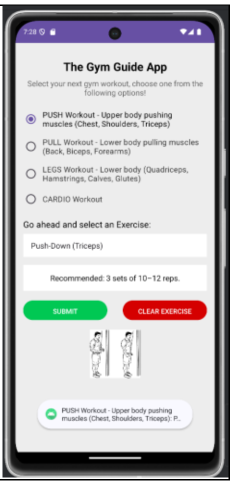
Go ahead and select an Exercise:

Overhead Press (Shoulders)

Recommended: 3 sets of 8 reps.

SUBMIT CLEAR EXERCISE

PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps): O...



The Gym Guide App

Select your next gym workout, choose one from the following options!

● PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps)

○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms)

○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes)

○ CARDIO Workout

Go ahead and select an Exercise:

Push-Down (Triceps)

Recommended: 3 sets of 10–12 reps.

SUBMIT CLEAR EXERCISE

PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps): P...

Table 2: PULL

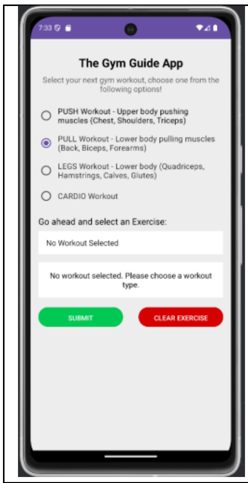
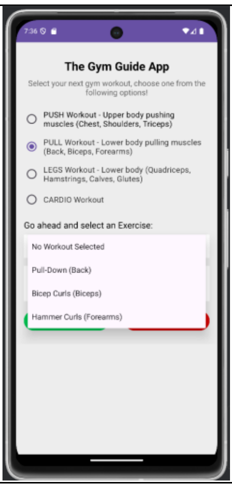
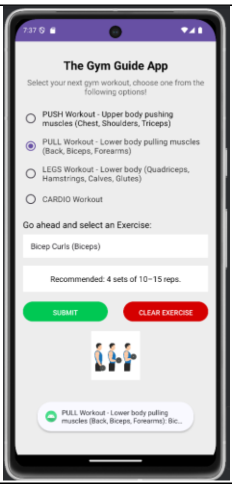
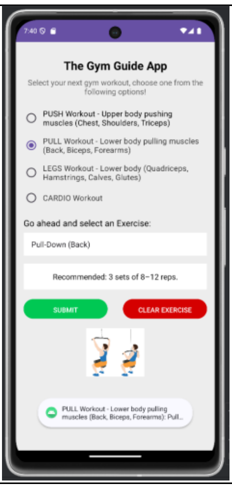
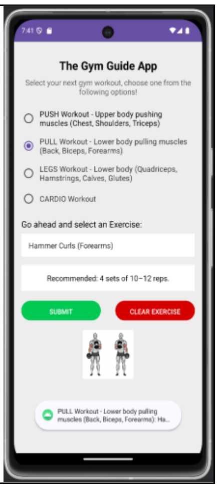
 7:33 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ● PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: No Workout Selected No workout selected. Please choose a workout type. SUBMIT CLEAR EXERCISE	 7:36 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ● PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: No Workout Selected Pull-Down (Back) Bicep Curls (Biceps) Hammer Curls (Forearms)	 7:37 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ● PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: Bicep Curls (Biceps) Recommended: 4 sets of 10–15 reps. SUBMIT CLEAR EXERCISE PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms): Bic...	 7:40 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ● PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: Pull-Down (Back) Recommended: 3 sets of 8–12 reps. SUBMIT CLEAR EXERCISE PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms): Pull...	 7:41 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ● PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: Hammer Curls (Forearms) Recommended: 4 sets of 10–12 reps. SUBMIT CLEAR EXERCISE PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms): Ha...

Table 3: LEGS

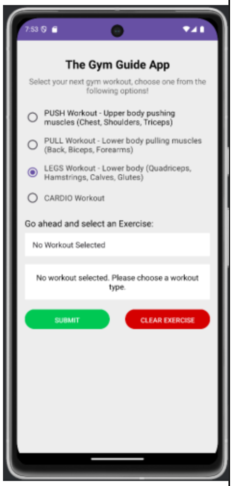
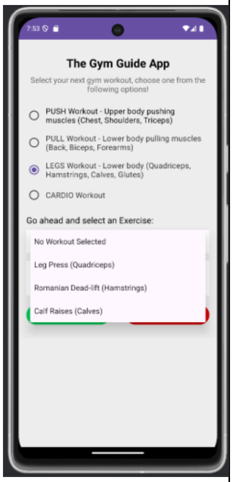
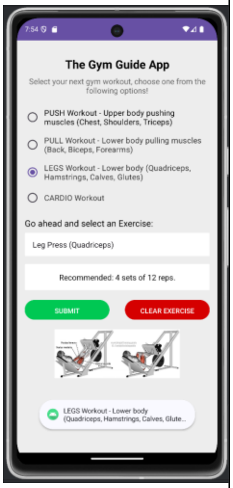
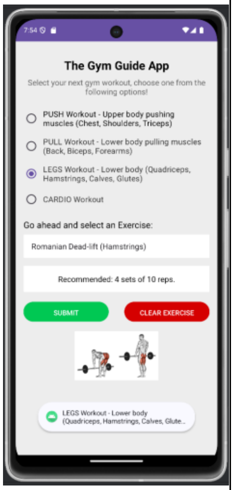
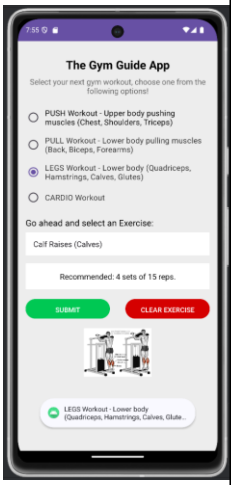
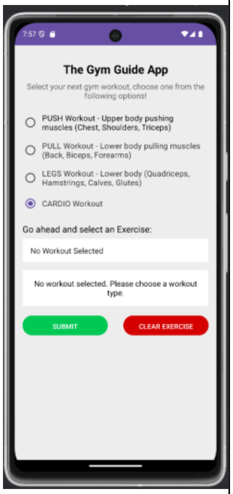
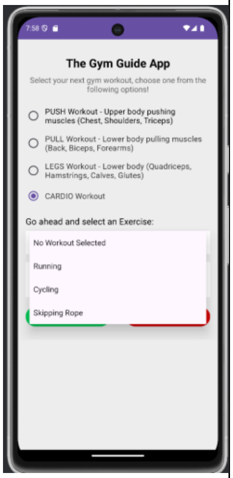
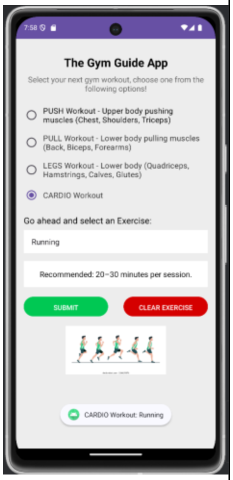
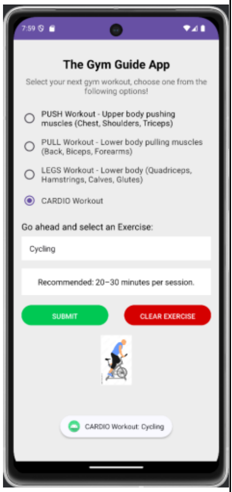
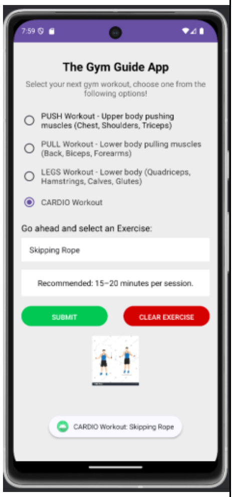
 7:53 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ● LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: No Workout Selected No workout selected. Please choose a workout type. SUBMIT CLEAR EXERCISE	 7:53 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ● LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: No Workout Selected Leg Press (Quadriceps) Romanian Dead-lift (Hamstrings) Calf Raises (Calves)	 7:54 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ● LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: Leg Press (Quadriceps) Recommended: 4 sets of 12 reps. SUBMIT CLEAR EXERCISE LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glute...	 7:54 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ● LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: Romanian Dead-lift (Hamstrings) Recommended: 4 sets of 10 reps. SUBMIT CLEAR EXERCISE LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glute...	 7:55 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ● LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ○ CARDIO Workout Go ahead and select an Exercise: Calf Raises (Calves) Recommended: 4 sets of 15 reps. SUBMIT CLEAR EXERCISE LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glute...

Table 4: CARDIO

 7:57 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ● CARDIO Workout Go ahead and select an Exercise: No Workout Selected No workout selected. Please choose a workout type. SUBMIT CLEAR EXERCISE	 7:58 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ● CARDIO Workout Go ahead and select an Exercise: No Workout Selected Running Cycling Skipping Rope	 7:58 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ● CARDIO Workout Go ahead and select an Exercise: Running Recommended: 20–30 minutes per session. SUBMIT CLEAR EXERCISE CARDIO Workout: Running	 7:59 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ● CARDIO Workout Go ahead and select an Exercise: Cycling Recommended: 20–30 minutes per session. SUBMIT CLEAR EXERCISE CARDIO Workout: Cycling	 7:59 The Gym Guide App Select your next gym workout, choose one from the following options! ○ PUSH Workout - Upper body pushing muscles (Chest, Shoulders, Triceps) ○ PULL Workout - Lower body pulling muscles (Back, Biceps, Forearms) ○ LEGS Workout - Lower body (Quadriceps, Hamstrings, Calves, Glutes) ● CARDIO Workout Go ahead and select an Exercise: Skipping Rope Recommended: 15–20 minutes per session. SUBMIT CLEAR EXERCISE CARDIO Workout: Skipping Rope

2.3

Task 3 : Man City Merchandise Order (Wireframe)

This app wireframe design follows a two-column grid layout for the items. Below the "Merchandise Selected" section, images dynamically update based on the selected items, giving users immediate feedback on their choices. A toast message is planned to display when the order is confirmed, providing the total cost which including the default delivery fee.

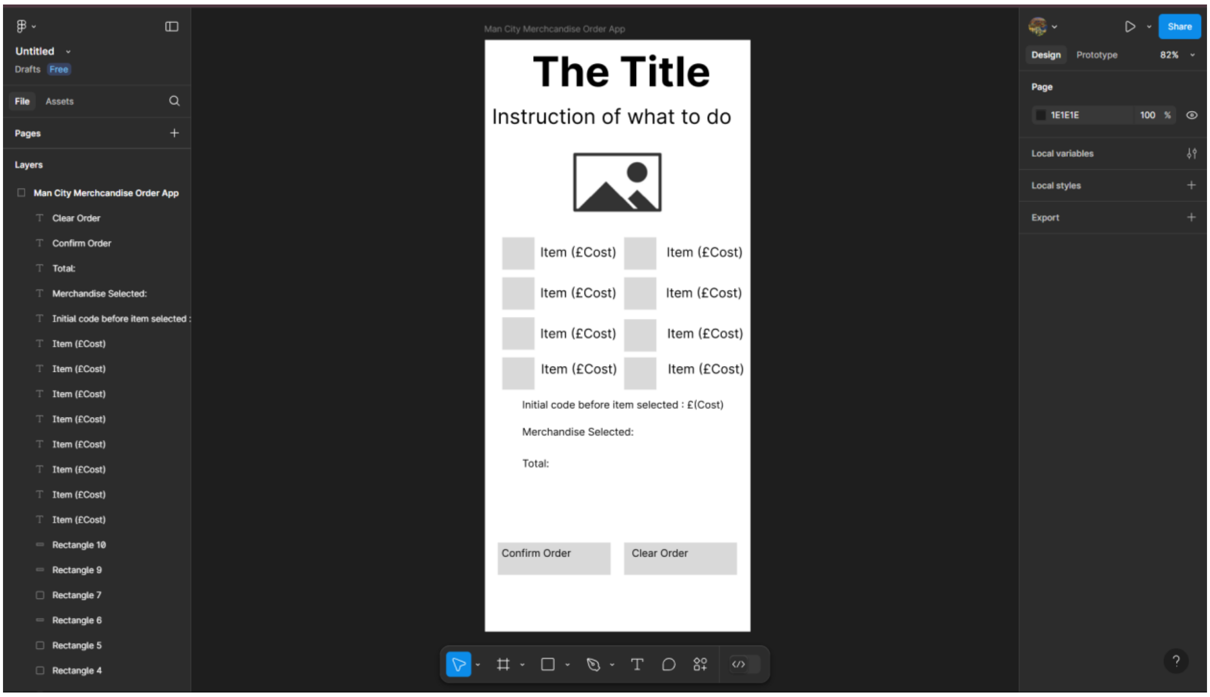


Figure 14: Main T3

Task 3 : Man City Merchandise Order (User Experience)

The app features checkboxes for item selection, dynamically updating the total cost and images in real-time. A confirm button displays a toast message with the total cost, while a reset button clears selections, returning to the default page. Users can visually see selected items, enhancing clarity. An additional image was added: one for football match-specific kits and another for non-match items like a scarf. This design ensures users can easily navigate and understand their selections while maintaining an engaging and user-friendly interface.

Table 5: Example 1 of User Experience

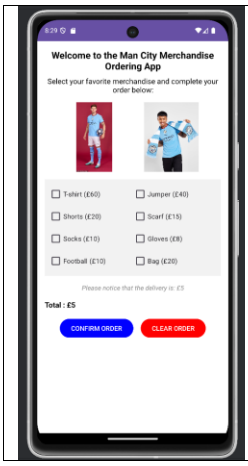
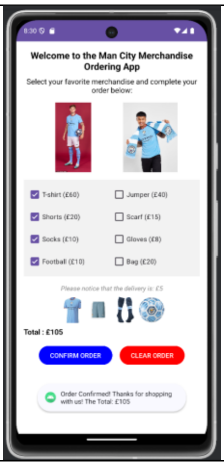
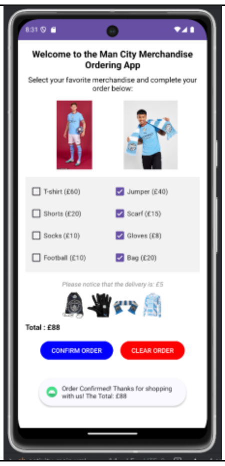
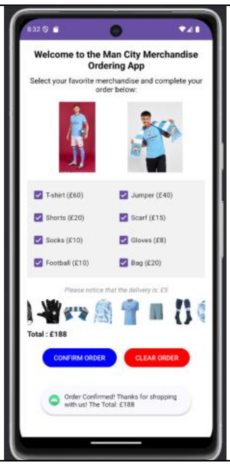
 8:29 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☐ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☐ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £5 CONFIRM ORDER CLEAR ORDER	 8:30 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☑ T-shirt (£60) ☐ Jumper (£40) ☑ Shorts (£20) ☐ Scarf (£15) ☑ Socks (£10) ☐ Gloves (£8) ☑ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £105 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £105	 8:31 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☑ Jumper (£40) ☐ Shorts (£20) ☑ Scarf (£15) ☐ Socks (£10) ☑ Gloves (£8) ☐ Football (£10) ☑ Bag (£20) Please notice that the delivery is: £5 Total : £88 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £88	 8:32 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☑ T-shirt (£60) ☑ Jumper (£40) ☑ Shorts (£20) ☑ Scarf (£15) ☑ Socks (£10) ☑ Gloves (£8) ☑ Football (£10) ☑ Bag (£20) Please notice that the delivery is: £5 Total : £188 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £188

Table 6: Example 2 of User Experience

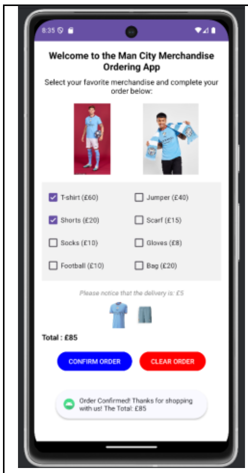
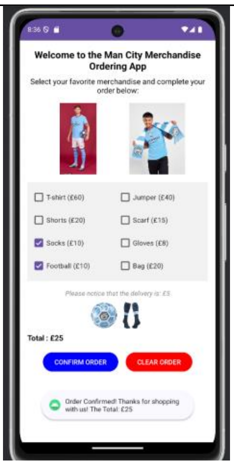
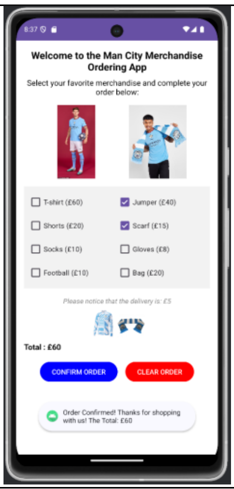
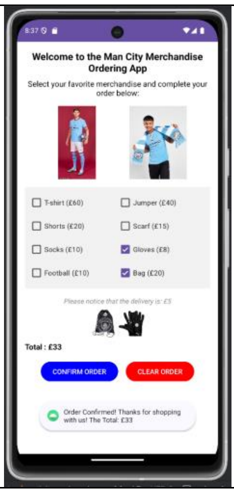
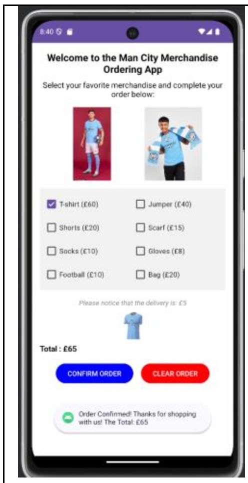
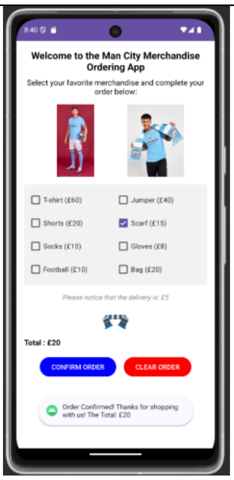
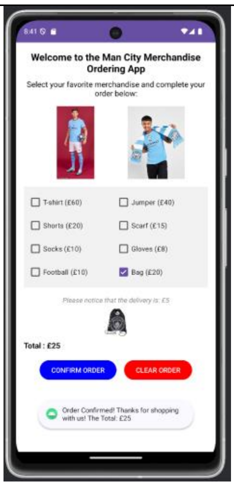
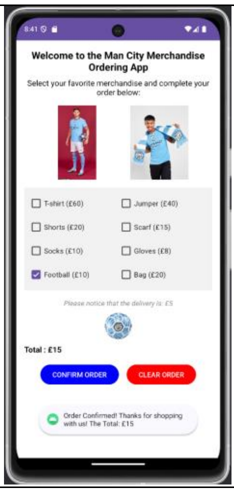
 8:35 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☑ T-shirt (£60) ☐ Jumper (£40) ☑ Shorts (£20) ☐ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☐ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £85 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £85	 8:36 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☐ Scarf (£15) ☑ Socks (£10) ☐ Gloves (£8) ☑ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £25 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £25	 8:37 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☑ Jumper (£40) ☐ Shorts (£20) ☑ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☐ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £60 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £60	 8:37 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☐ Scarf (£15) ☐ Socks (£10) ☑ Gloves (£8) ☐ Football (£10) ☑ Bag (£20) Please notice that the delivery is: £5 Total : £33 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £33

Table 7: Example 3 of User Experience

 8:40 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☑ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☐ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☐ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £65 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £65	 8:40 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☑ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☐ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £20 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £20	 8:41 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☐ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☐ Football (£10) ☑ Bag (£20) Please notice that the delivery is: £5 Total : £25 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £25	 8:41 Welcome to the Man City Merchandise Ordering App Select your favorite merchandise and complete your order below: ☐ T-shirt (£60) ☐ Jumper (£40) ☐ Shorts (£20) ☐ Scarf (£15) ☐ Socks (£10) ☐ Gloves (£8) ☑ Football (£10) ☐ Bag (£20) Please notice that the delivery is: £5 Total : £15 CONFIRM ORDER CLEAR ORDER Order Confirmed! Thanks for shopping with us! The Total: £15

2.4

Task 4 : Fab 4 in Test Cricket (Wireframe)

The first wireframe shows the main screen, which has the app title, a brief introduction, and four player images with a "Swipe Right" button to horizontally navigate right. The second wireframe focused on the player profiles, displaying the player's details, and buttons for navigating to the next or previous player. Images at the bottom allowed users to switch profiles directly.

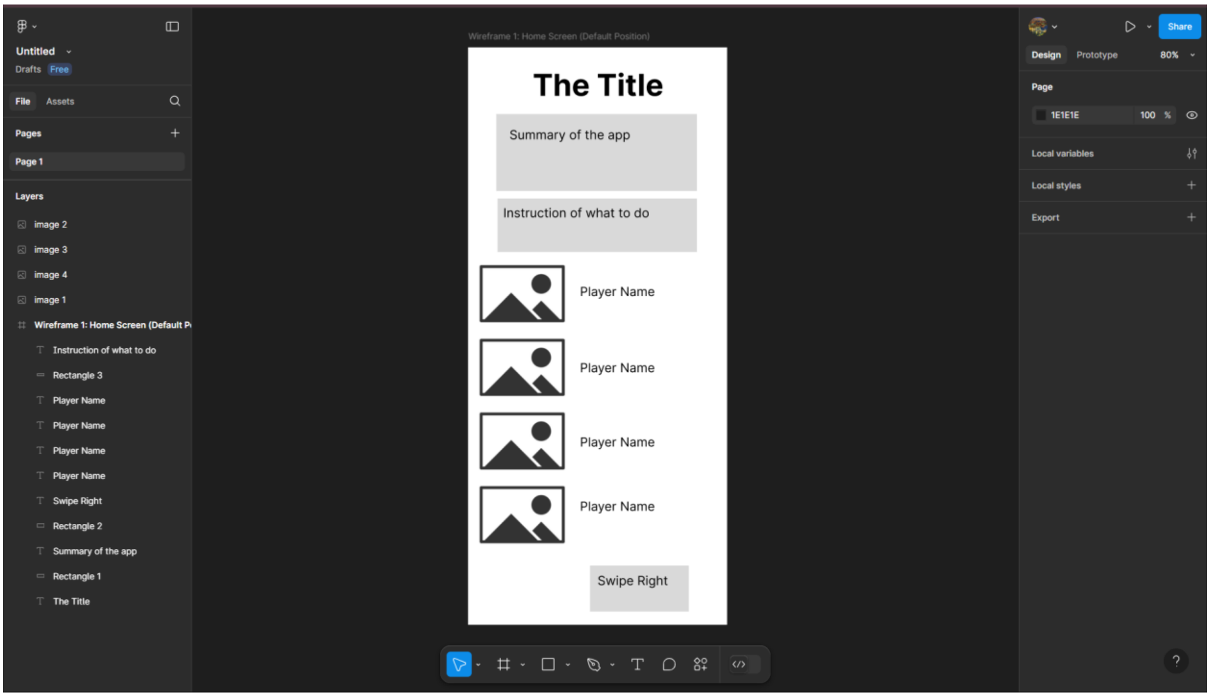


Figure 15: Default Page – T4 W1

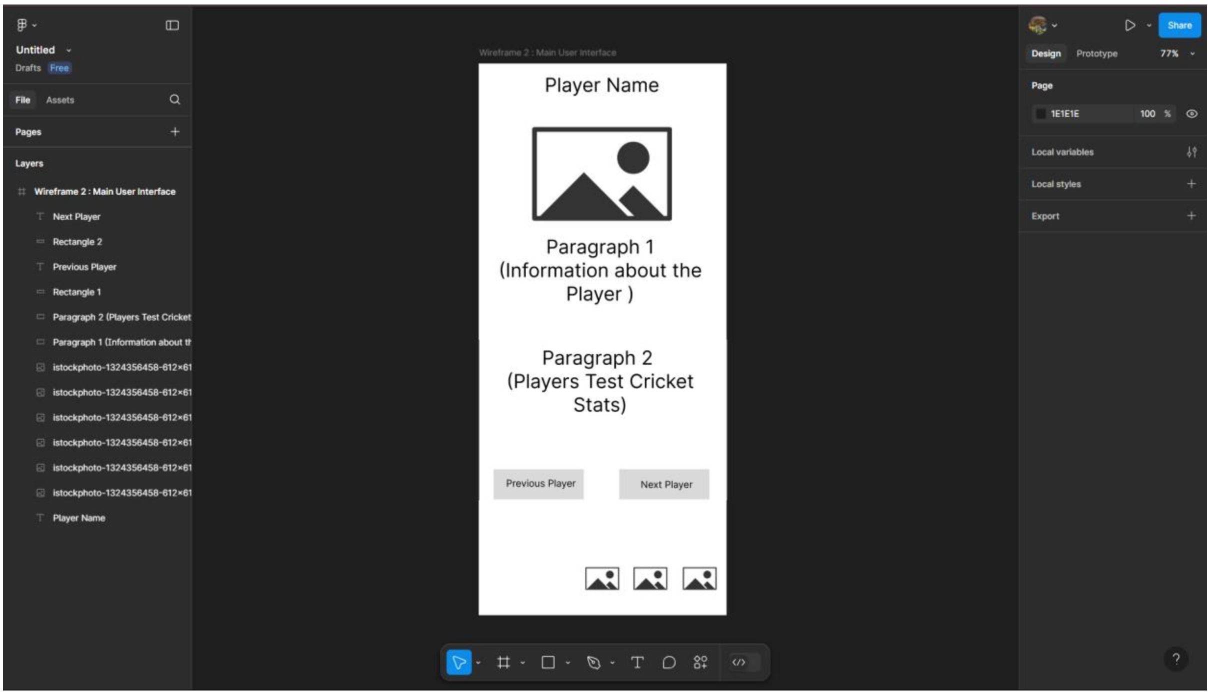
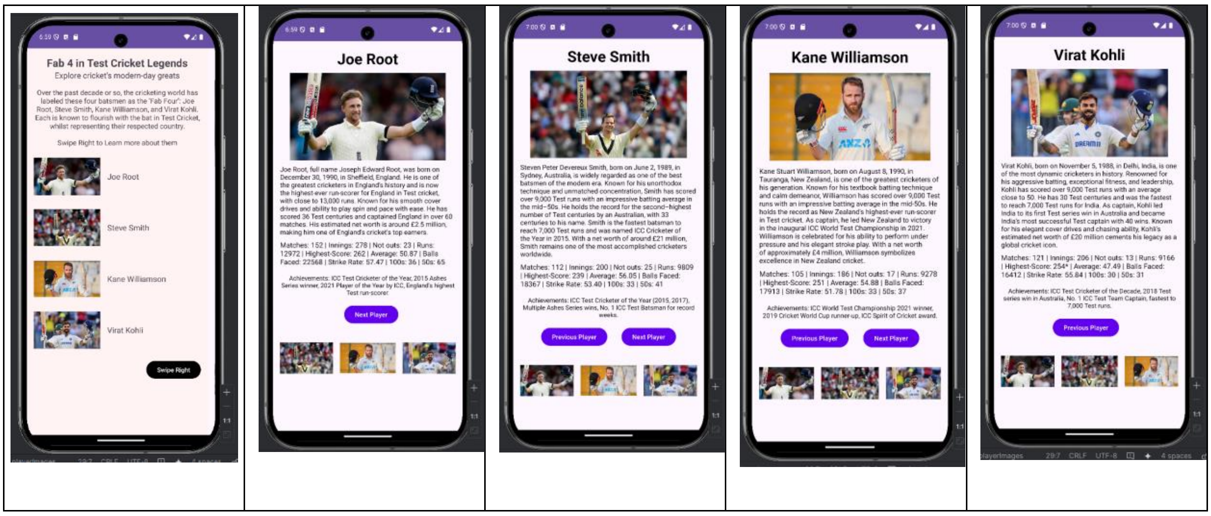

Untitled
Drafts Free
File Assets
Pages
Layers
Wireframe 2 : Main User Interface
Next Player
Rectangle 2
Previous Player
Rectangle 1
Paragraph 2 (Players Test Cricket
Paragraph 1 (Information about th
istockphoto-1324356458-612×61
istockphoto-1324356458-612×61
istockphoto-1324356458-612×61
istockphoto-1324356458-612×61
istockphoto-1324356458-612×61
istockphoto-1324356458-612×61
Player Name
Wireframe 2 : Main User Interface
Player Name
Paragraph 1 (Information about the Player)
Paragraph 2 (Players Test Cricket Stats)
Previous Player
Next Player
Share
Design Prototype 77%
Page
1E1E1E 100 %
Local variables
Local styles
Export

Figure 16: Player Page – T4 W2

Task 4 : Fab 4 in Test Cricket (User Experience)

Once opening the app, users first see a main screen with clear options to explore individual player profiles. The transition to a player's profile was done by a horizontal scroll. This design follows the same layouts across all screens. Additionally, the integration of the bottom images allowed users to access players pages directly without going through all screens.



Fab 4 in Test Cricket Legends

Explore cricket's modern-day greats

Over the past decade or so, the cricketing world has labeled these four batsmen as the 'Fab Four': Joe Root, Steve Smith, Kane Williamson, and Virat Kohli. Each is known to flourish with the bat in Test Cricket, whilst representing their respected country.

Swipe Right to Learn more about them

Joe Root

Joe Root, full name Joseph Edward Root, was born on December 30, 1990, in Sheffield, England. He is one of the greatest cricketers in England's history and is now the highest-ever run-scorer for England in Test cricket, with close to 13,000 runs. Known for his smooth cover drives and ability to play spin and pace with ease. He has scored 36 Test centuries and captained England in over 60 matches. His estimated net worth is around £2.5 million, making him one of England's cricket's top earners.

Matches: 152 | Innings: 278 | Not outs: 23 | Runs: 12972 | Highest-Score: 262 | Average: 50.87 | Balls Faced: 22568 | Strike Rate: 57.47 | 100s: 36 | 50s: 65

Achievements: ICC Test Cricketer of the Year, 2015 Ashes Series winner, 2021 Player of the Year by ICC, England's highest Test run-scorer.

Steve Smith

Steven Peter Devereux Smith, born on June 2, 1989, in Sydney, Australia, is widely regarded as one of the best batsmen of the modern era. Known for his unorthodox technique and unmatched concentration, Smith has scored over 9,000 Test runs with an impressive batting average in the mid−50s. He holds the record for the second−highest number of Test centuries by an Australian, with 33 centuries to his name. Smith is the fastest batsman to reach 7,000 Test runs and was named ICC Cricketer of the Year in 2015. With a net worth of around £21 million, Smith remains one of the most accomplished cricketers worldwide.

Matches: 112 | Innings: 200 | Not outs: 25 | Runs: 9809 | Highest-Score: 239 | Average: 56.05 | Balls Faced: 18367 | Strike Rate: 53.40 | 100s: 33 | 50s: 41

Achievements: ICC Test Cricketer of the Year (2015, 2017), Multiple Ashes Series wins, No. 1 ICC Test Batsman for record weeks.

Kane Williamson

Kane Stuart Williamson, born on August 8, 1990, in Tauranga, New Zealand, is one of the greatest cricketers of his generation. Known for his textbook batting technique and calm demeanor, Williamson has scored over 9,000 Test runs with an impressive batting average in the mid-50s. He holds the record as New Zealand's highest-ever run-scorer in Test cricket. As captain, he led New Zealand to victory in the inaugural ICC World Test Championship in 2021. Williamson is celebrated for his ability to perform under pressure and his elegant stroke play. With a net worth of approximately £4 million, Williamson symbolizes excellence in New Zealand cricket.

Matches: 105 | Innings: 186 | Not outs: 17 | Runs: 9278 | Highest-Score: 251 | Average: 54.88 | Balls Faced: 17913 | Strike Rate: 51.78 | 100s: 33 | 50s: 37

Achievements: ICC World Test Championship 2021 winner, 2019 Cricket World Cup runner-up, ICC Spirit of Cricket award.

Virat Kohli

Virat Kohli, born on November 5, 1988, in Delhi, India, is one of the most dynamic cricketers in history. Renowned for his aggressive batting, exceptional fitness, and leadership, Kohli has scored over 9,000 Test runs with an average close to 50. He has 30 Test centuries and was the fastest to reach 7,000 Test runs for India. As captain, Kohli led India to its first Test series win in Australia and became India's most successful Test captain with 40 wins. Known for his elegant cover drives and chasing ability, Kohli's estimated net worth of £20 million cements his legacy as a global cricket icon.

Matches: 121 | Innings: 206 | Not outs: 13 | Runs: 9166 | Highest-Score: 254* | Average: 47.49 | Balls Faced: 16412 | Strike Rate: 55.84 | 100s: 30 | 50s: 31

Achievements: ICC Test Cricketer of the Decade, 2018 Test series win in Australia, No. 1 ICC Test Team Captain, fastest to 7,000 Test runs.

Figure 17: User Experience : T4

2.5

Task 5 : CryptoStock Login (Wireframe)

The login screen included essential elements such as input fields for the username and password, a login button to click to go to the welcome screen or to register is not yet done so. For the registration screen, the wireframe displayed fields for the user's details, along with button to submit the form, clearing the form, and navigating back. The welcome screen shows the user's key information conveying a successful login.

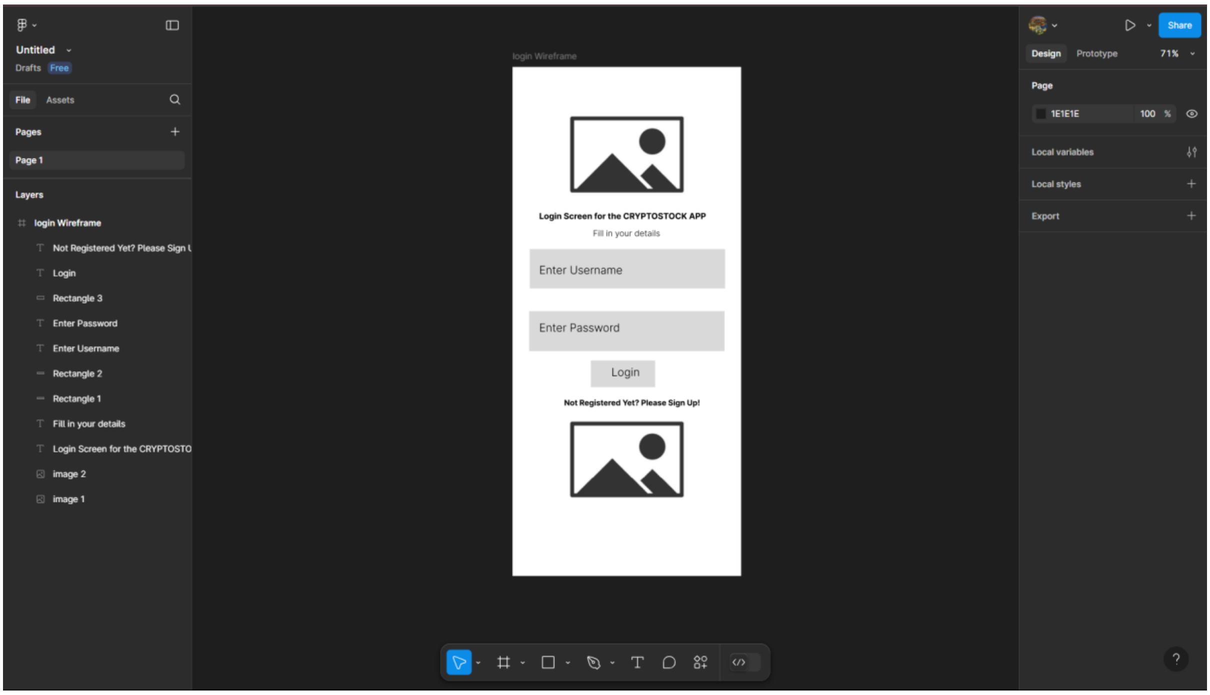


Figure 18: Login Page – T5 W1

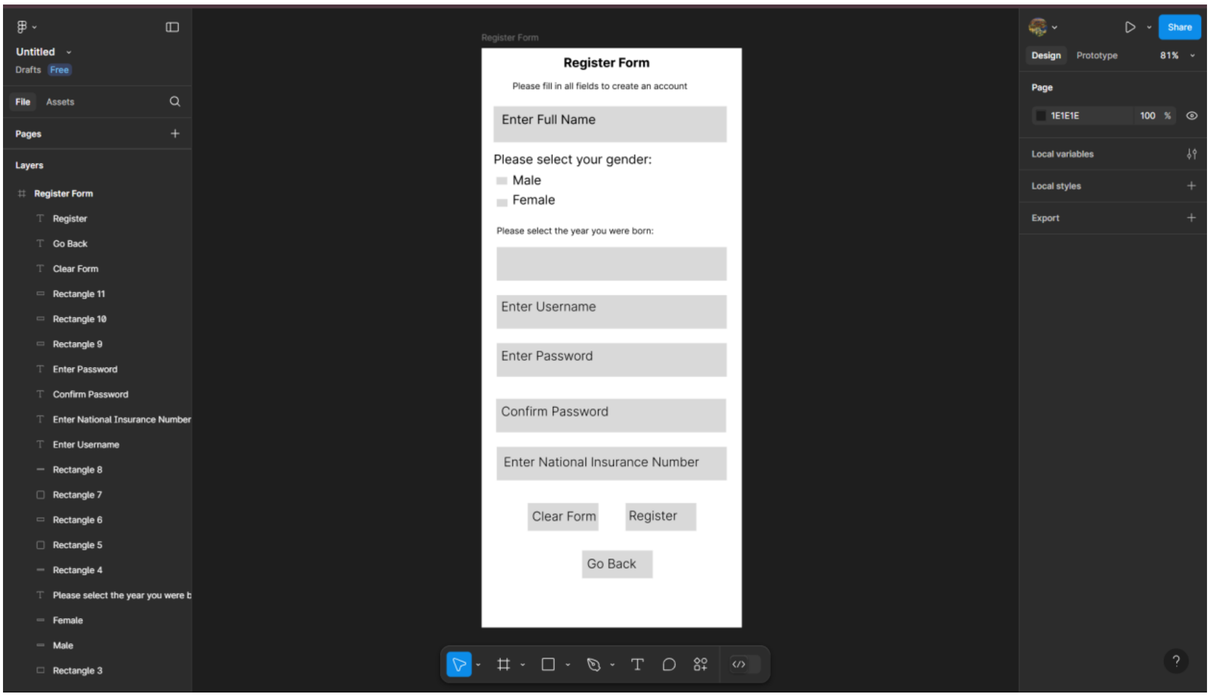
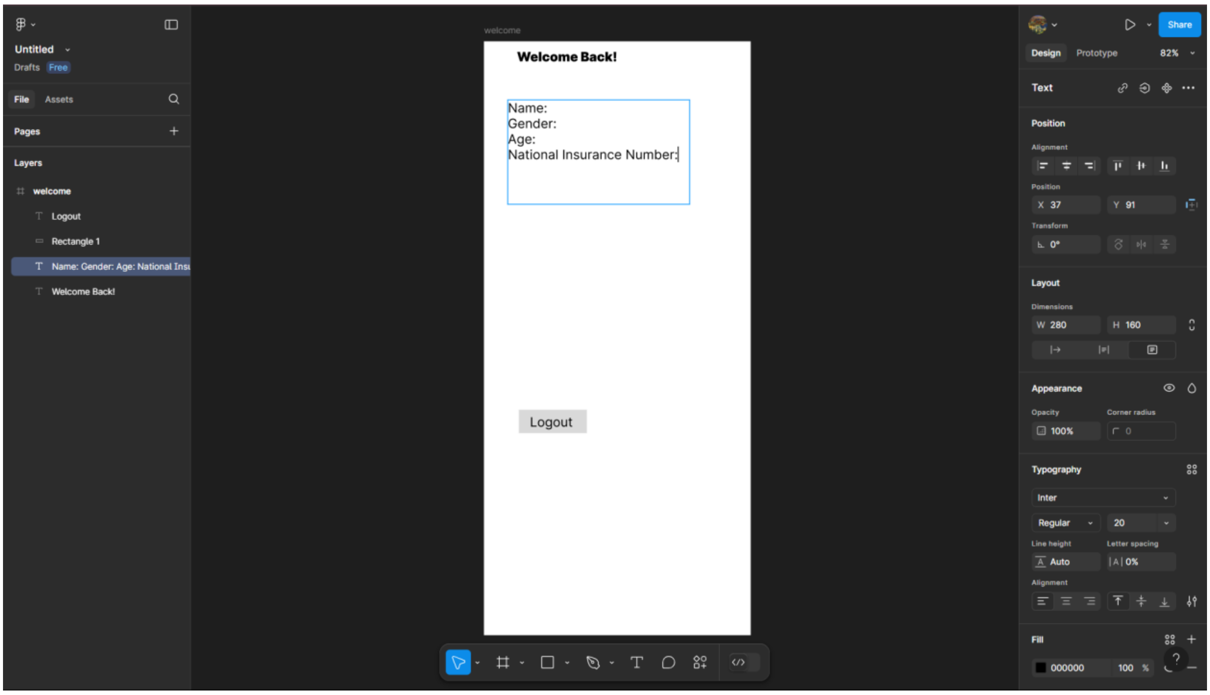


Figure 19: Register Form – T5 W2



welcome

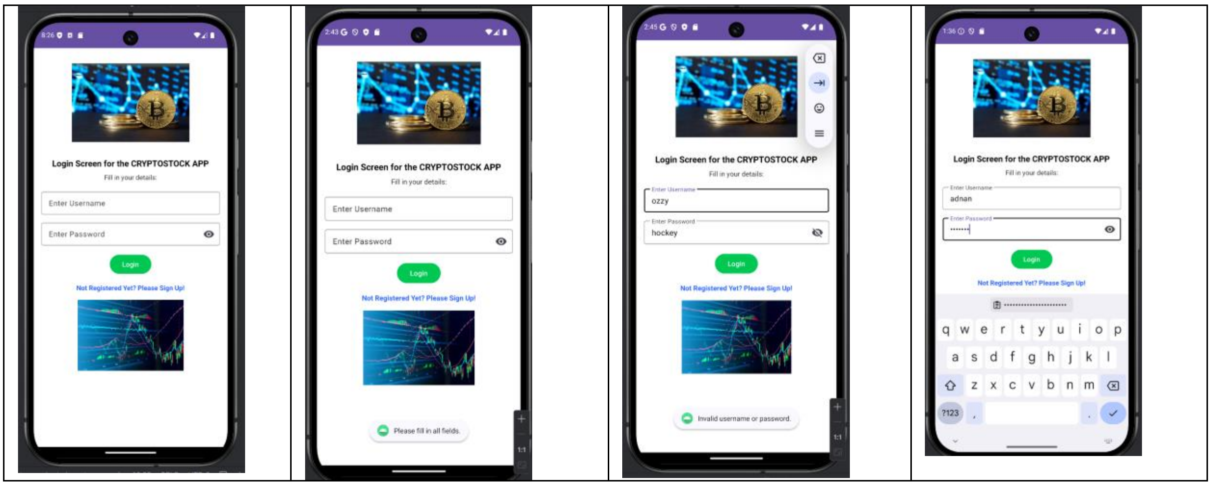
Welcome Back!

Name:
Gender:
Age:
National Insurance Number:

Figure 20: Welcome – T5 W3

Task 5 : CryptoStock Login (User Experience)

The login screen provides immediate feedback for incorrect details or fields that haven't been completed. The registration screen uses validation checks for every field, including matching passwords and selecting birthyear from the spinner, preventing incomplete or wrong registrations. The welcome page displays the details of the user which shows completed and successful registration. To help with the aesthetics of the app I added background images of stocks. Toasts messages would show errors or success, ensuring the user knew about their login in state.


Login Screen for the CRYPTOSTOCK APP
Fill in your details:
Enter Username
Enter Password
Login
Not Registered Yet? Please Sign Up!
ozzy
hockey
adnan

Please fill in all fields.

Invalid username or password.

Figure 21: Login Page - Task 5 User Experience

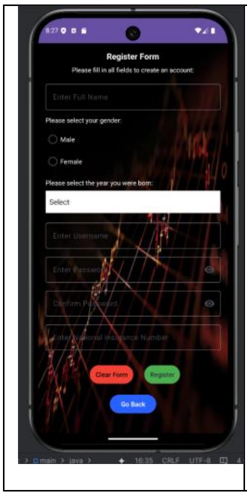
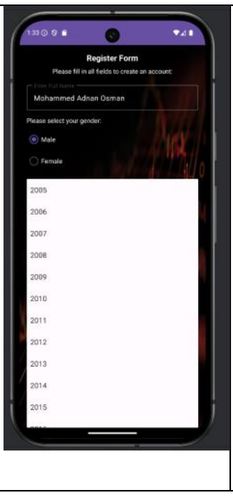
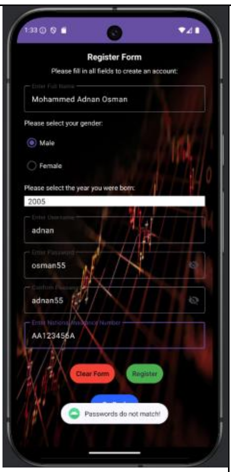
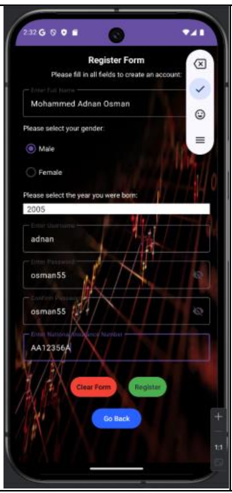
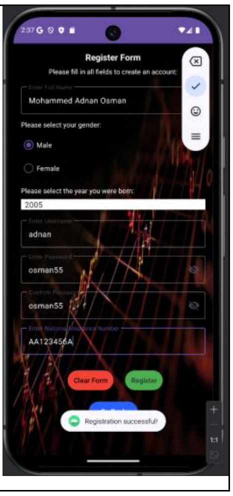
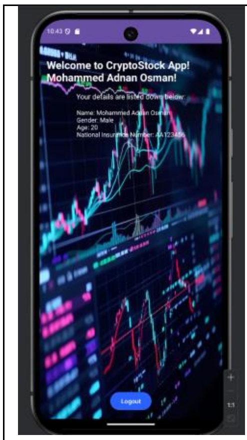
 Register Form Please fill in all fields to create an account: Enter Full Name Please select your gender: ○ Male ○ Female Please select the year you were born: Select Enter Username Enter Password Confirm Password Enter National Insurance Number Clear Form Register Go Back	 Register Form Please fill in all fields to create an account: Enter Full Name Mohammed Adnan Osman Please select your gender: ● Male ○ Female 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015	 Register Form Please fill in all fields to create an account: Enter Full Name Mohammed Adnan Osman Please select your gender: ● Male ○ Female Please select the year you were born: 2005 Enter Username adnan Enter Password osman55 Confirm Password adnan55 Enter National Insurance Number AA123456A Clear Form Register Passwords do not match!	 Register Form Please fill in all fields to create an account: Enter Full Name Mohammed Adnan Osman Please select your gender: ● Male ○ Female Please select the year you were born: 2005 Enter Username adnan Enter Password osman55 Confirm Password osman55 Enter National Insurance Number AA12356A Clear Form Register Go Back	 Register Form Please fill in all fields to create an account: Enter Full Name Mohammed Adnan Osman Please select your gender: ● Male ○ Female Please select the year you were born: 2005 Enter Username adnan Enter Password osman55 Confirm Password osman55 Enter National Insurance Number AA123456A Clear Form Register Registration successful!

Table 8: Registration Page - Task 5 User Experience



Welcome to CryptoStock App!
Mohammed Adnan Osman!

Your details are listed down below:

Name: Mohammed Adnan Osman
Gender: Male
Age: 20
National Insurance Number: AA123456

Logout

Figure 22: Welcome Page - Task 5 User Experience

2.6

Task 6 : Athlete Connect Contacts (Wireframe)

The main wireframe includes a search bar at the top for filtering contacts by the letters the user searches and a structured list below displaying the athletes' attributes. The search wireframe demonstrates how the app dynamically updates to show common letters in the name. The call-to-contact wireframe shows what will happen after clicking the contact such as initiating calls, sending messages.

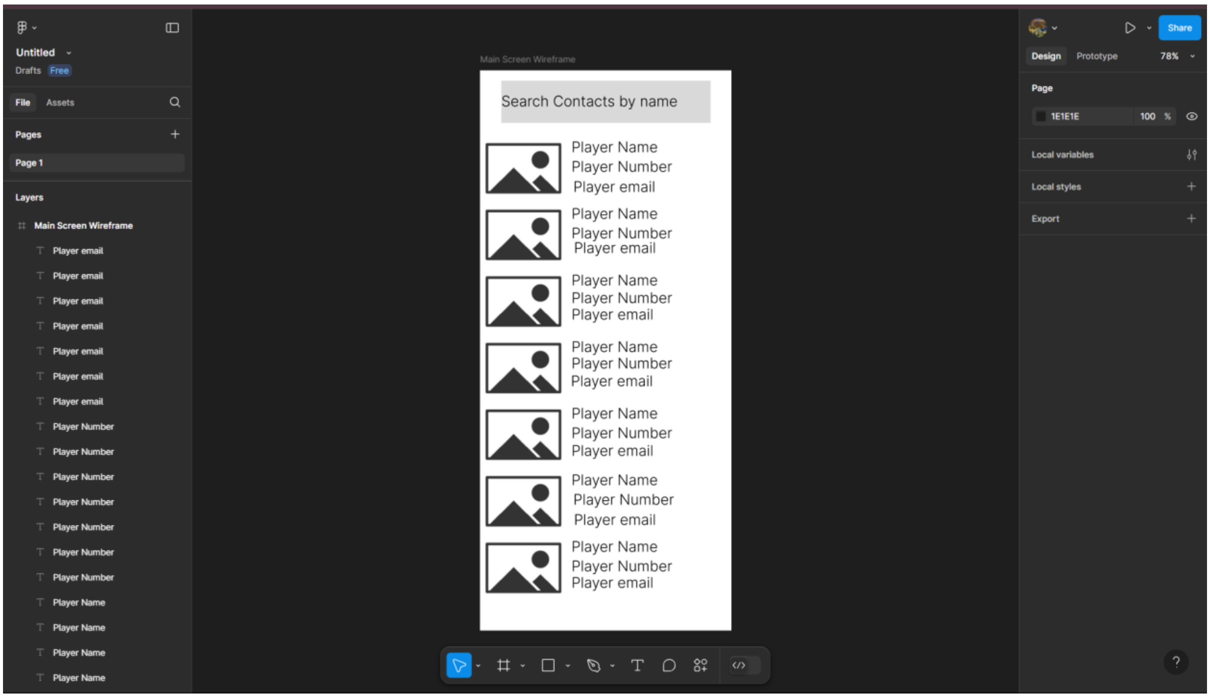


Figure 23: Main – T6 W1

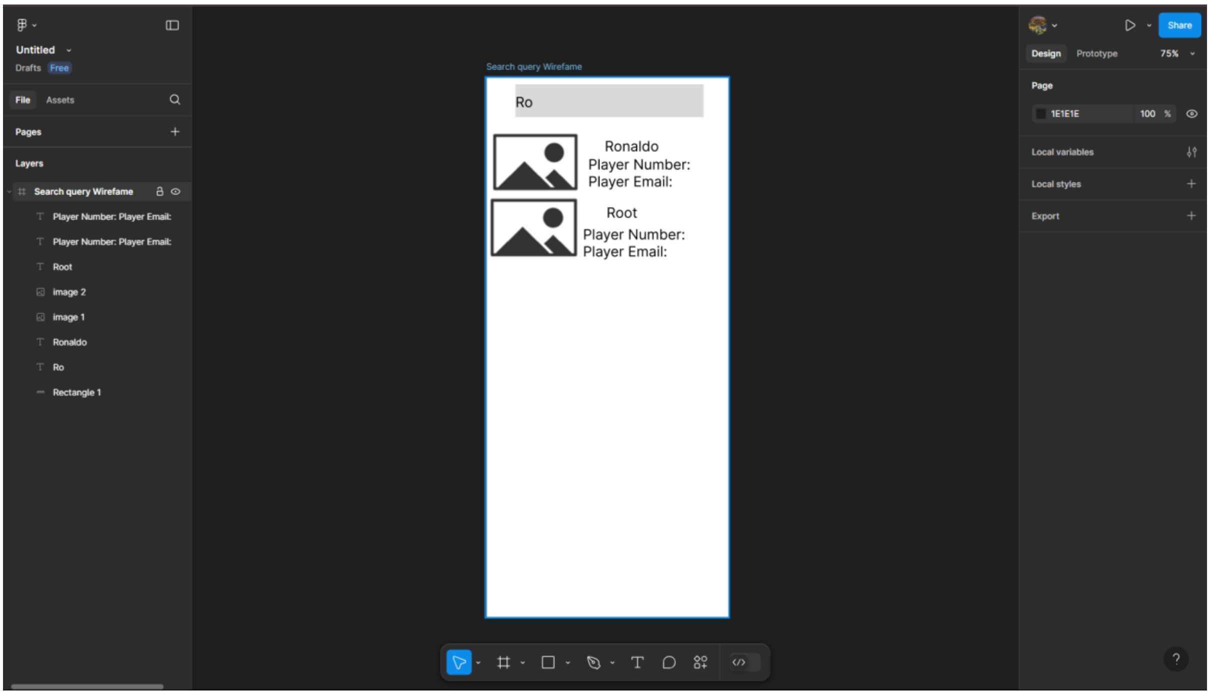


Figure 24: Search query – T6 W2

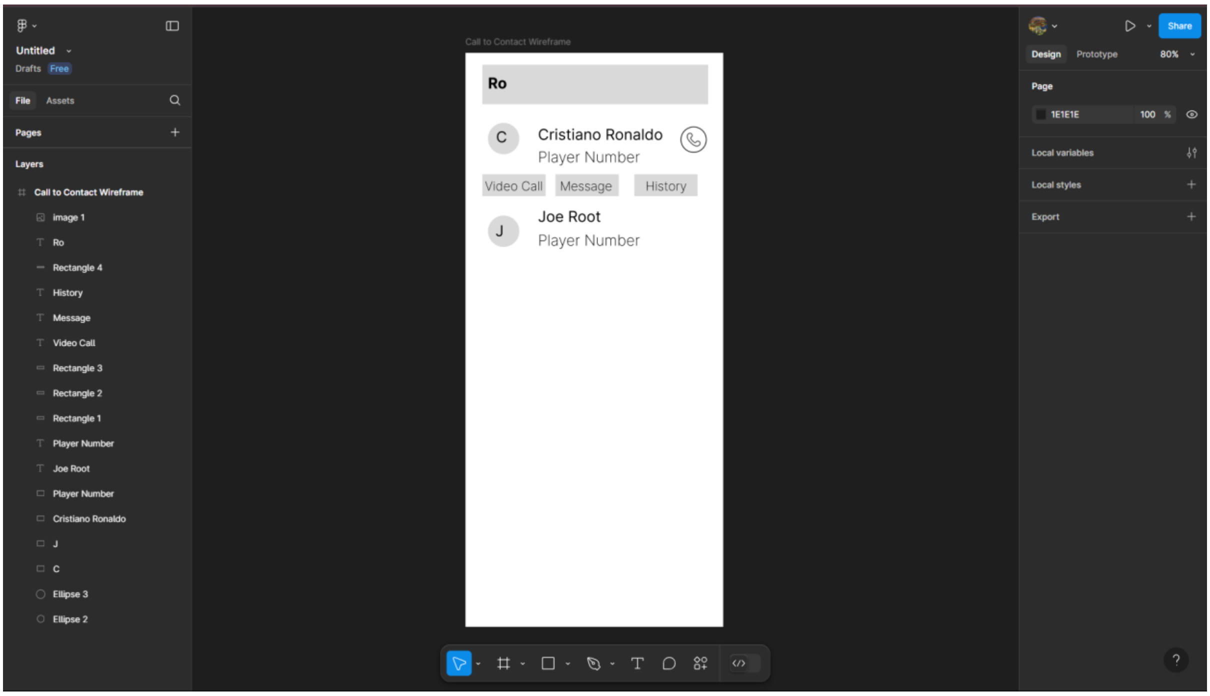


Figure 25: Call to Contact T6 W3

Task 6 : Athlete Connect Contacts (User Experience)

The main screen lists contacts with details and includes a search bar for finding specific contacts by name. An additional profession detail was added for each athlete. Selecting a contact allows users to call or message. The final implementation aligns with the wireframes, offering a simple, user-friendly layout .

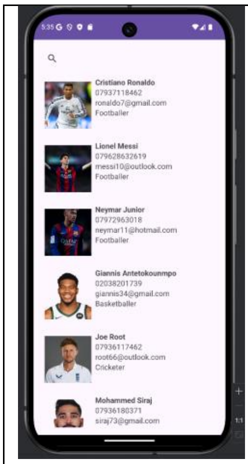
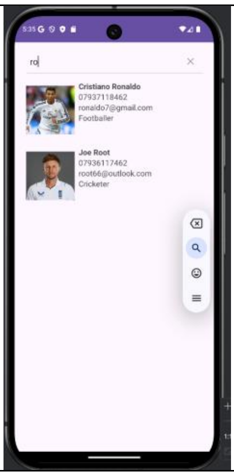
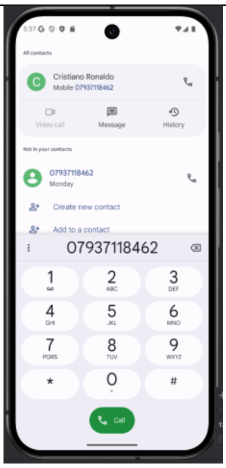
		

Table 9: Task 6 Wireframes working on app (User Experience)

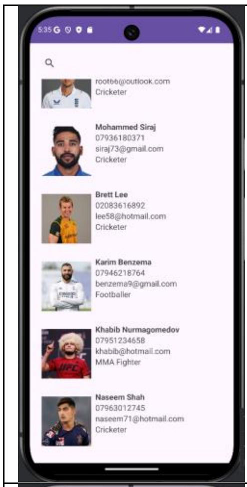
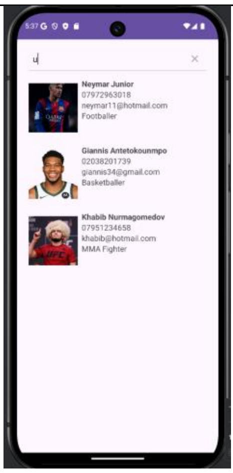
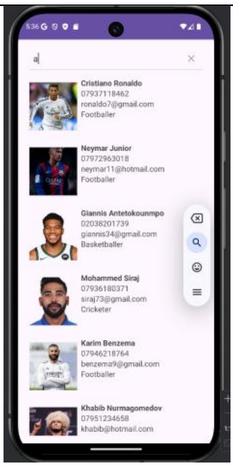
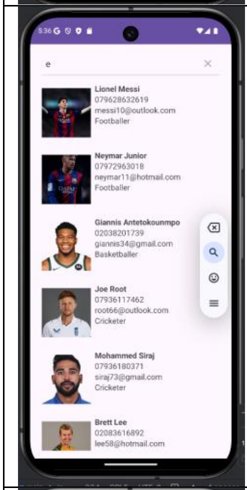
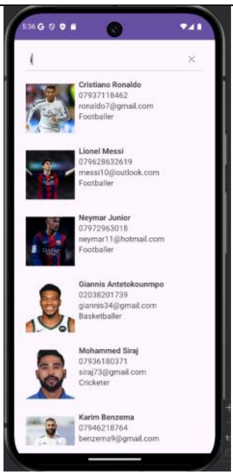
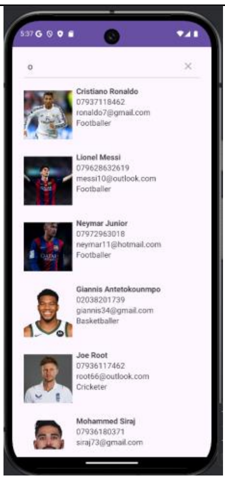
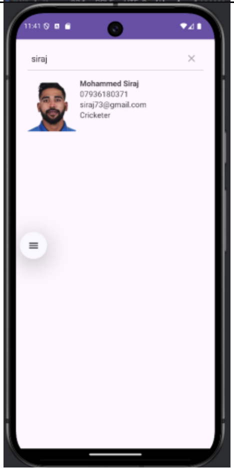
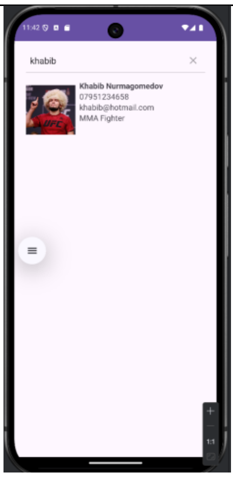
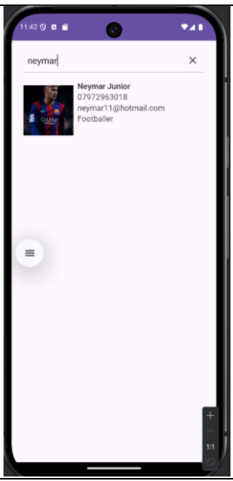
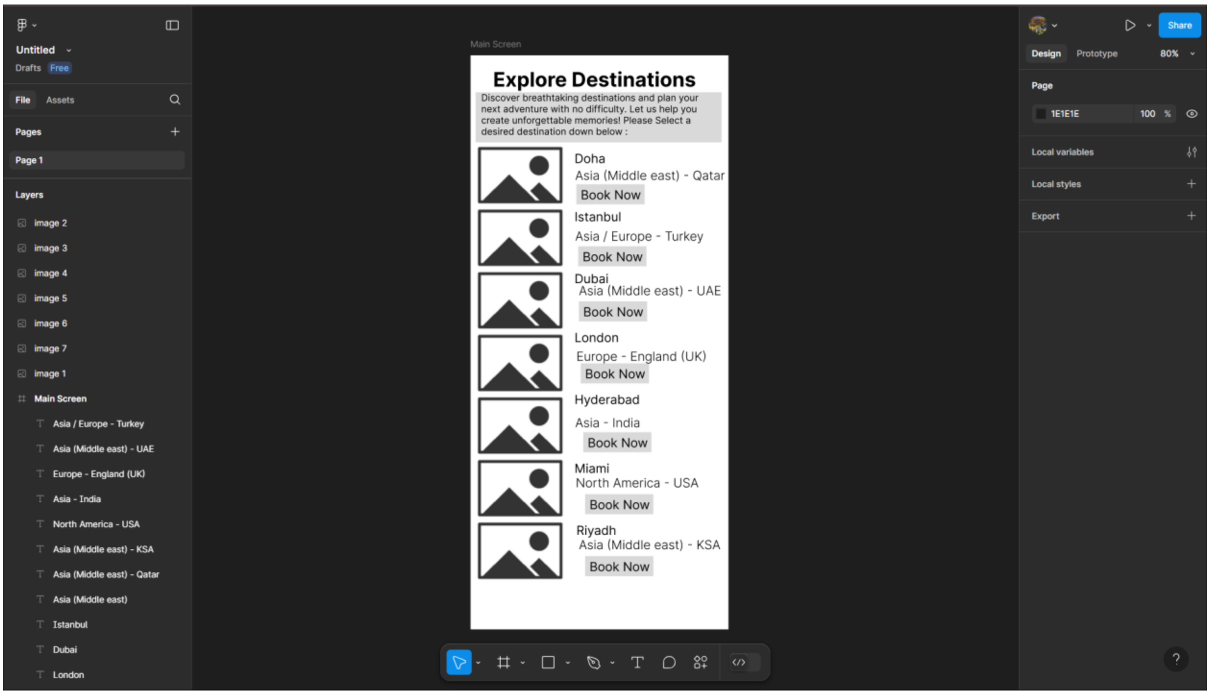
 root66@outlook.com Cricketer Mohammed Siraj 07936180371 siraj73@gmail.com Cricketer Brett Lee 02083616892 lee58@hotmail.com Cricketer Karim Benzema 07946218764 benzema9@gmail.com Footballer Khabib Nurmagomedov 07951234658 khabib@hotmail.com MMA Fighter Naseem Shah 07963012745 naseem71@hotmail.com Cricketer	 u Neymar Junior 07972963018 neymar11@hotmail.com Footballer Giannis Antetokounmpo 02038201739 giannis34@gmail.com Basketballer Khabib Nurmagomedov 07951234658 khabib@hotmail.com MMA Fighter	 a Cristiano Ronaldo 07937118462 ronaldo7@gmail.com Footballer Neymar Junior 07972963018 neymar11@hotmail.com Footballer Giannis Antetokounmpo 02038201739 giannis34@gmail.com Basketballer Mohammed Siraj 07936180371 siraj73@gmail.com Cricketer Karim Benzema 07946218764 benzema9@gmail.com Footballer Khabib Nurmagomedov 07951234658 khabib@hotmail.com
 e Lionel Messi 079628632619 messi10@outlook.com Footballer Neymar Junior 07972963018 neymar11@hotmail.com Footballer Giannis Antetokounmpo 02038201739 giannis34@gmail.com Basketballer Joe Root 07936117462 root66@outlook.com Cricketer Mohammed Siraj 07936180371 siraj73@gmail.com Cricketer Brett Lee 02083616892 lee58@hotmail.com	 i Cristiano Ronaldo 07937118462 ronaldo7@gmail.com Footballer Lionel Messi 079628632619 messi10@outlook.com Footballer Neymar Junior 07972963018 neymar11@hotmail.com Footballer Giannis Antetokounmpo 02038201739 giannis34@gmail.com Basketballer Mohammed Siraj 07936180371 siraj73@gmail.com Cricketer Karim Benzema 07946218764 benzema9@gmail.com	 o Cristiano Ronaldo 07937118462 ronaldo7@gmail.com Footballer Lionel Messi 079628632619 messi10@outlook.com Footballer Neymar Junior 07972963018 neymar11@hotmail.com Footballer Giannis Antetokounmpo 02038201739 giannis34@gmail.com Basketballer Joe Root 07936117462 root66@outlook.com Cricketer Mohammed Siraj 07936180371 siraj73@gmail.com
 siraj Mohammed Siraj 07936180371 siraj73@gmail.com Cricketer	 khabib Khabib Nurmagomedov 07951234658 khabib@hotmail.com MMA Fighter	 neymar Neymar Junior 07972963018 neymar11@hotmail.com Footballer

Table 10: Testing Search Query Task 6

2.7

Task 7 : TravelNow (Wireframe)

The app starts with a scrollable list of cities. Clicking “Book Now” takes users to a calendar to select departure and return dates. Users can then proceed to checkout or explore other city bookings. At checkout, a successful booking message appears, and “Close App” ends the program. Initially, images were in portrait , but I changed them to landscape for better user experience as that’s what the better pictures were in.



Explore Destinations

Discover breathtaking destinations and plan your next adventure with no difficulty. Let us help you create unforgettable memories! Please Select a desired destination down below :

Doha
Asia (Middle east) - Qatar
Book Now

Istanbul
Asia / Europe - Turkey
Book Now

Dubai
Asia (Middle east) - UAE
Book Now

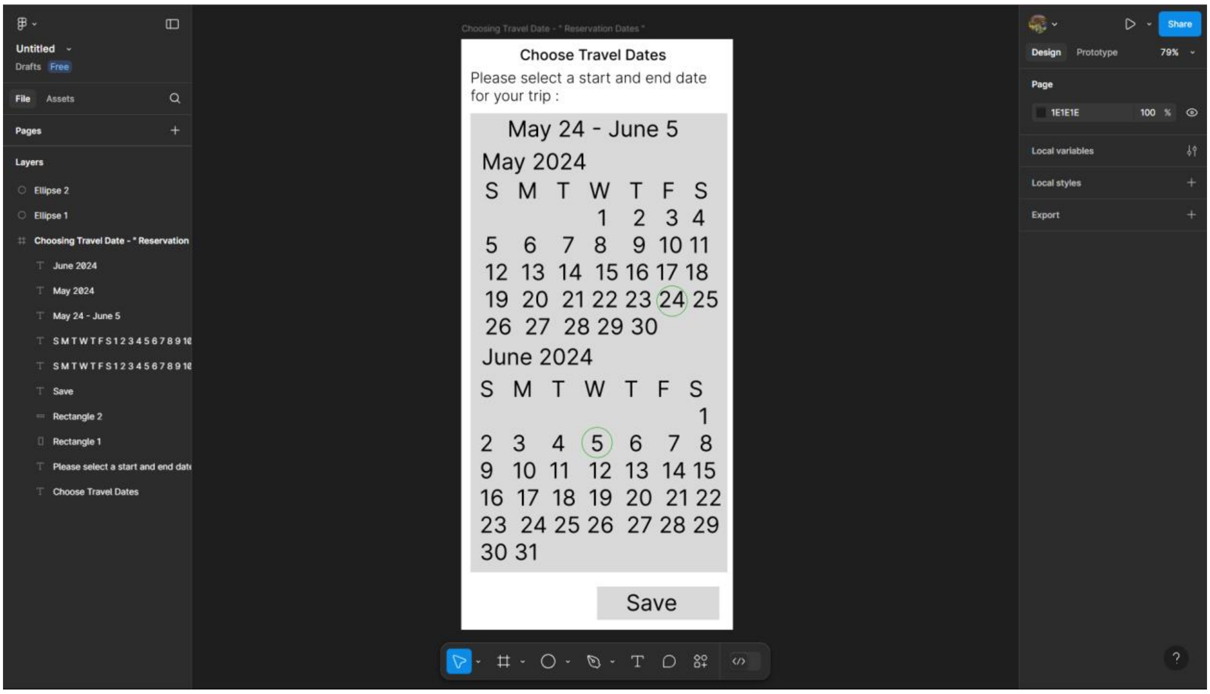
London
Europe - England (UK)
Book Now

Hyderabad
Asia - India
Book Now

Miami
North America - USA
Book Now

Riyadh
Asia (Middle east) - KSA
Book Now

Figure 26: Main T7 W1



Choose Travel Dates

Please select a start and end date for your trip :

May 24 - June 5

May 2024

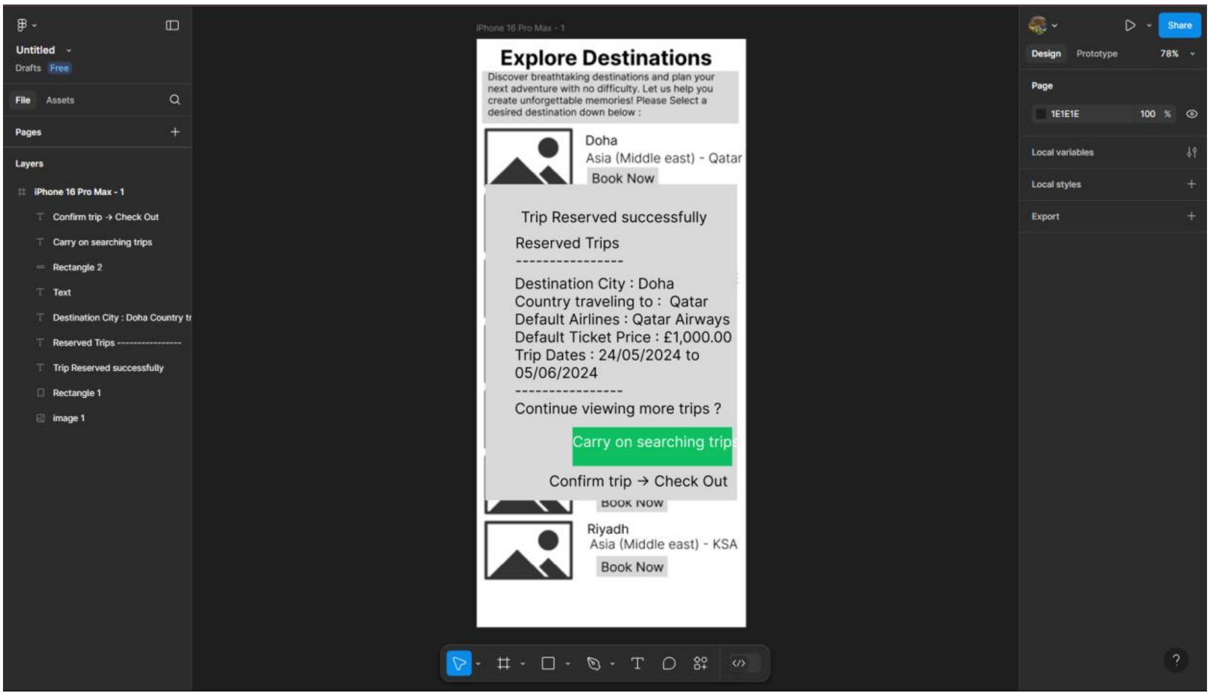
```
S  M  T  W  T  F  S
         1  2  3  4
5  6  7  8  9  10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

June 2024

```
S  M  T  W  T  F  S
                  1
2  3  4  5  6  7  8
9  10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

Save

Figure 27: Calendar T7 W2



Explore Destinations

Discover breathtaking destinations and plan your next adventure with no difficulty. Let us help you create unforgettable memories! Please Select a desired destination down below :

Doha
Asia (Middle east) - Qatar
Book Now

Trip Reserved successfully

Reserved Trips

Destination City : Doha
Country traveling to : Qatar
Default Airlines : Qatar Airways
Default Ticket Price : £1,000.00
Trip Dates : 24/05/2024 to 05/06/2024

Continue viewing more trips ?

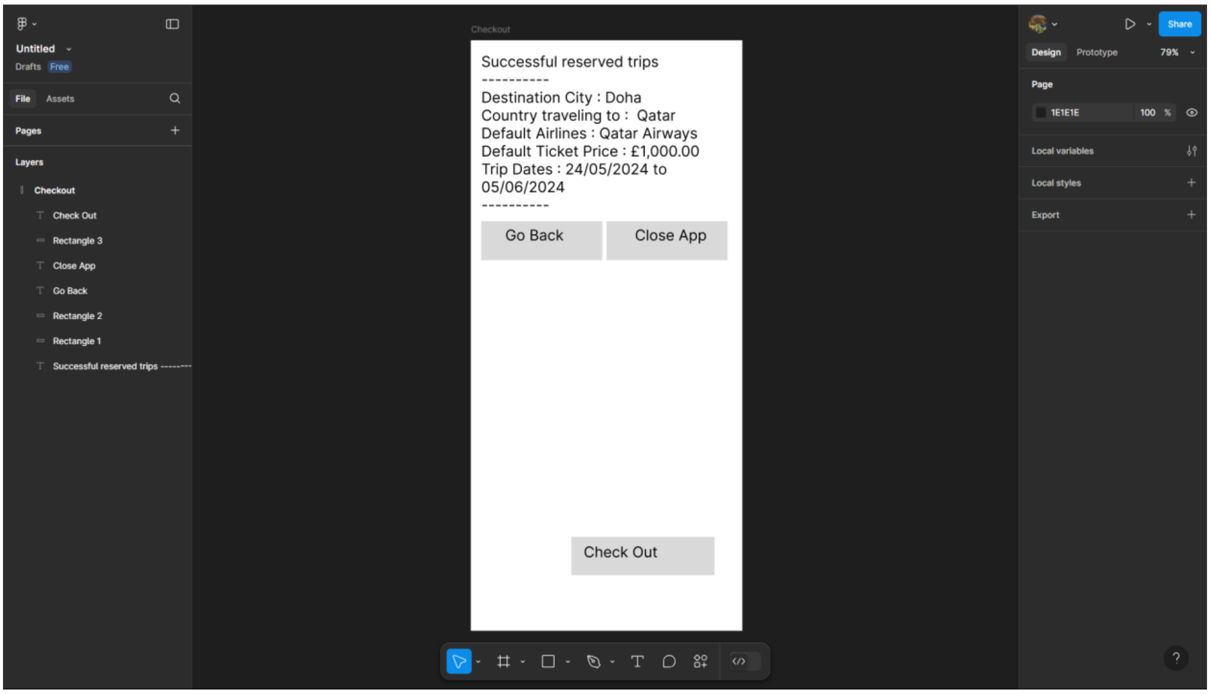
Carry on searching trip

Confirm trip → Check Out

Book Now

Riyadh
Asia (Middle east) - KSA
Book Now

Figure 28: Reserved Booking T7 W3



Successful reserved trips

Destination City : Doha
Country traveling to : Qatar
Default Airlines : Qatar Airways
Default Ticket Price : £1,000.00
Trip Dates : 24/05/2024 to 05/06/2024

Go Back Close App

Check Out

Figure 29: Successful Checkout T7 W4

Task 7 : TravelNow (User Experience)

The home screen displays a list of cities in a scrollable RecyclerView, with a “Book Now” button allowing users to select travel dates via a calendar. Users can add dates directly or scroll through the calendar. After selecting dates, they can return to the main screen to view other cities or proceed to the checkout page. A separate page for date selection was created to ensure modularity, making the task manageable as it was my first attempt this sort of practise, resulting in slight differences from the wireframes but same functionality.

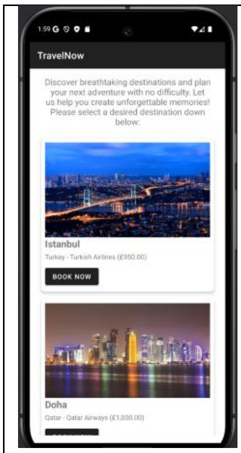
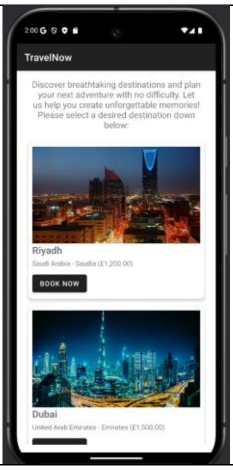
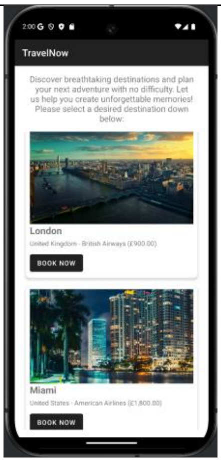
 TravelNow Discover breathtaking destinations and plan your next adventure with no difficulty. Let us help you create unforgettable memories! Please select a desired destination down below: Istanbul Turkey - Turkish Airlines (£950.00) BOOK NOW Doha Qatar - Qatar Airways (£1,000.00)	 TravelNow Discover breathtaking destinations and plan your next adventure with no difficulty. Let us help you create unforgettable memories! Please select a desired destination down below: Riyadh Saudi Arabia - Saudia (£1,200.00) BOOK NOW Dubai United Arab Emirates - Emirates (£1,500.00)	 TravelNow Discover breathtaking destinations and plan your next adventure with no difficulty. Let us help you create unforgettable memories! Please select a desired destination down below: London United Kingdom - British Airways (£900.00) BOOK NOW Miami United States - American Airlines (£1,800.00) BOOK NOW

Table 11: Main Screen , Cities to choose from – T7

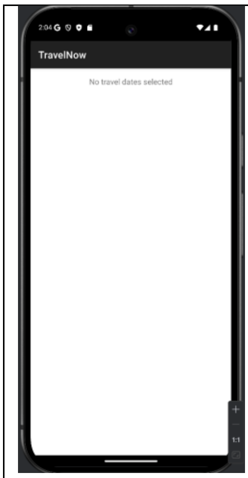
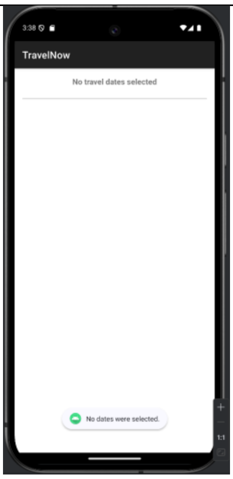
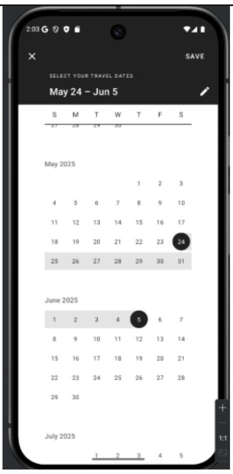
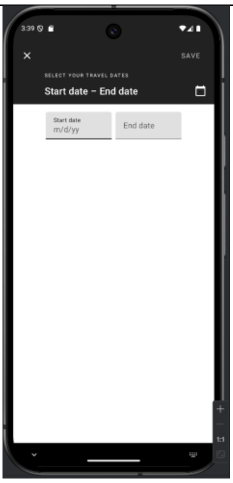
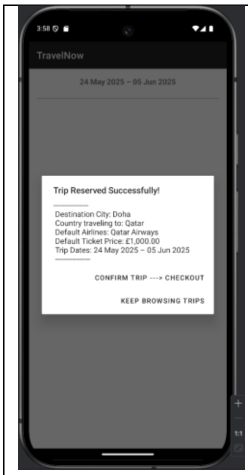
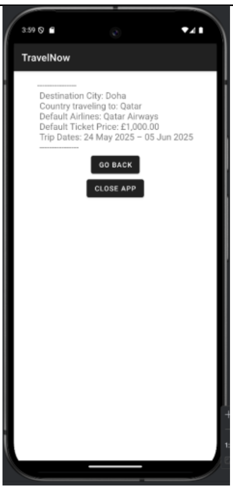
 TravelNow No travel dates selected	 TravelNow No travel dates selected No dates were selected.	 SAVE SELECT YOUR TRAVEL DATES May 24 – Jun 5 May 2025 June 2025 July 2025	 SAVE SELECT YOUR TRAVEL DATES Start date – End date Start date m/d/yy End date *Explained in 5.1*

Table 12: Choose Dates Screen , Calendar – T7

 TravelNow 24 May 2025 – 05 Jun 2025 Trip Reserved Successfully! Destination City: Doha Country traveling to: Qatar Default Airlines: Qatar Airways Default Ticket Price: £1,000.00 Trip Dates: 24 May 2025 – 05 Jun 2025 CONFIRM TRIP ---> CHECKOUT KEEP BROWSING TRIPS *Figure 30: Successfully reserved trip - T7*	 TravelNow Destination City: Doha Country traveling to: Qatar Default Airlines: Qatar Airways Default Ticket Price: £1,000.00 Trip Dates: 24 May 2025 – 05 Jun 2025 GO BACK CLOSE APP *Figure 31: Checkout - T7*

Chapter 3 : Back – End Development

3.1 Task 1: Cute Icons

The back-end development implemented animations, sound playback, and text updates efficiently. Initially, repetitive code for icons was used but became tedious and not efficient so created a reusable setIconListener method. This method combined animations, sounds, and text updates, improving code maintainability by accepting unique icon identifiers. The shake animation logic, initially suggested by ChatGPT as a separate method, was modified into setIconListener, simplifying functionality. Sound playback was handled using the MediaPlayer class, allowing animal sounds to play when icons were clicked. These improvements reduced redundancy and enhanced user interaction.

3.2 Task 2: The Gym Guide

A key issue with handling radio buttons arose due to using a switch statement, which caused compilation errors as the generated ID couldn't be used as a constant. This was resolved by replacing the switch with an if-else statement.

(Before)

```
switch (selectedId) {
     case R.id.pushRB:
selectedWorkout = "Push Workout";
     break;
```

```
//...
```

(After)

```
if (sameId == R.id.pushRB) {
   ListOfExercises = addDefaultPlaceholder(getResources().getStringArray(R.array.pushSA));
   workoutsInfo = getString(R.string.pushIS);
} else if (sameId == R.id.pullRB) {
   ListOfExercises = addDefaultPlaceholder(getResources().getStringArray(R.array.pullSA));
```

```
//...
```

Initially, the spinner lacked a dynamic placeholder, defaulting to the first exercise. Adding a placeholder like "No Exercise Selected" ensured clarity. Diagrams presented another challenge as they stayed visible when switching workouts. Implementing a reset cleared or updated diagrams dynamically based on the selection.

Event listeners handled interactions efficiently. The RadioGroup updated the spinner and work out details, while the spinner’s OnItemSelectedListener dynamically displayed exercise diagrams. Buttons for submitting or clearing selections were integrated with toast messages to confirm user actions. The modular design facilitates future updates.

3.3 Task 3 : The Man City Merchandise Order

Initially, each checkbox had its own setOnCheckedChangeListener, leading to duplicate code making it longer to code. This approach was changed using a onCheckboxClicked method, dynamically handling all checkboxes using IDs. Centring the merch into one method made adding items a lot easier.

Another critical feature was the dynamic calculation of the total cost. The app included a default delivery fee, which was shows with the updated total. Logical operations were used to change the cost in real time based on the checkboxes clicked.

Dynamic image updates were another important component. Based on user interactions, images were added or removed dynamically from the layout. This was done using methods like addView and removeView.

3.4 Task 4 : Fab 4 in Test Cricket

The Player class stored attributes like names, images, descriptions, and achievements, enabling easy updates to player information. Arrays were used for dynamic data binding, allowing the app to display content based on the selected player's index during horizontal scrolling. Navigation between pages used buttons to update the player index and arrays of image IDs for quick profile access via click listeners.

External intents facilitated transitions between activities, with intent.putExtra() used to pass the current player index. Initially lacking error handling, ChatGPT suggested improvements, ensuring easy and correct changes. Implementing both horizontal scrolling and linked images at the bottom of the screen was challenging .

Overcoming navigation errors and enhancing dynamic updates deepened my understanding of Android programming. These skills will be applicable in future projects for university or work, particularly those requiring dynamic user interface updates.

3.5 Task 5 : CryptoStock Login

The MainActivity.java manages login functionality, using input validation to ensure username and password fields are filled. It retrieves stored details via SharedPreferences, verifying them in handleLogin() before navigating to the welcome screen with navigateToWelcome().

The RegistrationActivity.java handles user registration, validating fields and using a dynamic spinner to generate years (1950 to the current year) via the Java Calendar class. Incomplete fields trigger a toast error message. The users data is saved in SharedPreferences with unique keys like _nin. The clearForm() method allows users to reset the form.

The WelcomeScreen.java retrieves user details from SharedPreferences to dynamically display personalized information. Age calculation subtracts the birth year from the current year, ensuring accurate results.

3.6 Task 6 : Athlete Connects Contact

The MainActivity class initialises the RecyclerView and connects it to the AthleteAdapter, dynamically displaying athletes' details. It implements a SearchView for real-time filtering when user enters the players name. Athletes' details are loaded using a predefined list, with names, numbers, and images from the strings.xml and drawable file.

The AthleteAdapter class connects the Athlete data model to the RecyclerView. It dynamically shows attributes such as names, phone numbers, and images using the onBindViewHolder method. The adapter also has the filter to manage search queries. Additionally, it has Intent.ACTION_DIAL, enabling users to initiate calls directly from the list, enhancing interactivity and user engagement.

The Athlete class is the data model which has the essential attributes such as name, phone number, email, profession, and profile image resource ID. Each attribute is accessible through getter methods like getName() and getPhone(). This modular structure me to manage the details.

3.7 Task 7 : TravelNow

A key feature was fetching JSON data from an external API , handled by the GetJsonData class. Initially, challenges arose with FileNotFoundException due to API misconfigurations which AI assisted me to solve here. The solution was to have error-handling components. The RecyclerView adapter was between the parsed JSON data and the user interface components. This was achieved by iterating over the JSON array, mapping data into City objects, and dynamically updating the RecyclerView using the adapter. The MainActivity also incorporated OnClickListener interfaces to handle navigation between the screens, including the date picker and checkout summary. Lecture slides recommended using glide to be integrated to load images efficiently into the RecyclerView.

Chapter 4: AI and CoPilot Tools

4.1 Task 1: Cute Icons

ChatGPT helped to implement the shake.xml file for animations and suggested using AnimationUtils.loadAnimation for integrating XML-based animations and the MediaPlayer class for sound playback. I modified the logic by embedding animations directly in the setIconListener method and renaming variables. These enhancements improved the app's visual appeal and user experience efficiently.

4.2 Task 2: The Gym Guide

ChatGPT was used here and there , but a significant issue was with the spinner . Its default selection was of the first exercise, potentially confusing users. ChatGPT suggested dynamically adding a placeholder to the exercise list, prompting explicit user selection with a method like:

```
private String[] addDefaultPlaceholder(String[] exercises) {
    String[] updatedExercises = new String[exercises.length + 1];
    //..
  }
```

Another issue was previous diagrams still showing when switching workout splits. ChatGPT recommended resetting the diagram to null when no valid resource ID was available:

```
if (diagramResId != 0) {

  diagram.setImageResource(diagramResId);

} else {

  diagram.setImageDrawable(null); // Clear image if no diagram is available

}
```

4.3 Task 3 : The Man City Merchandise Order

ChatGPT was used here and there but mainly in using addView() and removeView() for dynamic changes and setTag() to assign tags to ImageView objects for efficient identification and removal. These suggestions, combined with my modifications, enhanced my ability to manage dynamic changes , improving both functionality and design while deepening my understanding of dynamic UI development in mobile apps.

4.4 Task 4 : Fab 4 in Test Cricket

The lecture slides covered basic Android functionalities but lacked details on features like external intents, requiring additional research. ChatGPT assisted in implementing intent.putExtra() for passing player indices between activities, ensuring correct navigation. It also suggested using error handles, including null checks and integration of finish(). ChatGPT’s assistance on dynamic information, such as arrays and null checks like if (nameTextView != null), improved functionality. These insights helped resolve minor errors and enhanced my understanding of Android programming, particularly in managing intents and dynamic data effectively.

4.5 Task 5 : CryptoStocks Login

ChatGPT significantly aided in implementing SharedPreferences for securely storing user information like usernames and passwords. It said to use key-value pairs across activities, which, combined with lecture slides, helped the process. For example, while ChatGPT suggested a foundational method for saving user details, I customised it by creating unique keys (e.g., username + "_fN"). This assistance reduced development time and enhanced my understanding of SharedPreferences, especially as it was my first implementation.

4.6 Task 6 : Athlete Connector Contact

ChatGPT was used in this task for example, the AthleteAdapter's data-binding logic and filtering functionality were assisted from AI and using the lecture slides. In the MainActivity, AI helped in the integration of RecyclerView .I needed assistance as I wasn't grasping how to implement it from the lecture slides .

4.7 Task 7 : TravelNow

ChatGPT was used in debugging JSON parsing issues as it was my first time implementing it and was finding it hard to combine the knowledge of the lecture slide to my app so bridging that gap AI was used. For example, I used the slides for the GetJsonData class and linking it to the MainActivity, but AI assisted in integrated it to work properly without errors. It also helped when implementing Glide for the image loading. So it helped in not just helping me understand how to implement Json data but solved the problems I encountered when making the app.

Chapter 5: Reflection and Discussion

5.1 Completion of Work

I completed all seven tasks, meeting all requirements and changed themes to reduce similarity. However, in the TravelNow App, the calendar displays dates in the American format (MM/DD/YYYY) instead of (DD/MM/YYYY) . Despite searching online and implementing the code it didn't work. This minor issue doesn't affect the app's core functionality, and all requirements and practices for each task have been met. I have completed the whole report.

5.2 Bugs and Challenges

In the Gym Guide , switching workouts initially failed to clear previous diagrams, causing confusion, but this was resolved. The Man City Merchandise App had an issue with dynamic image updates not syncing with item sections, which was fixed by correcting listener handling. Lastly, the Fab 4 in Test Cricket App had a navigation bug causing index mismatches between player profiles, but debugging resolved it. These solutions fixed the program .

5.3 Above and Beyond

The Cute Icons App features shaken animations and real animal sounds for an engaging experience. The Gym Guide App includes dynamic diagrams to visually demonstrate exercises, improving clarity. In the Fab 4 in Test Cricket App, an external intent is used between player profiles. While I couldn't go above and beyond in all tasks due to time constraints, I plan to expand on other projects post-assessment to further improve my skills.

5.4 Self-Critical Evaluation

I completed all 7 tasks but had to rush the last two due to time constraints. I applied for an extension on 17th December but received no response, so I adapted using lecture slides as usual and YouTube and used AI more extensively. Early tasks were easier , aided by doing seminar activities. This module felt highly relevant to real-world employment, building on my first-year Java knowledge. Creating apps was enjoyable, by how Java and XML layouts create real apps. I'm glad I studied this module.

Chapter 6: Conclusion

Developing seven mobile applications increased my knowledge Java in Android Studio, making this work a success by meeting task requirements and adding valuable skills to get a job. External tools like ChatGPT, Stack Overflow, and lecture slides were crucial in resolving errors and ensuring functionality. If I had more time, I would have explored techniques like using SQLite to store user details in the login app or implementing validation for the National Insurance number. Looking ahead, I plan to enhance this application further while building upon this foundation in mobile development. This module has changed my view on coding, as seeing it applied in real apps has motivated me to pursue additional projects independently.

References

Urban Programmer (2016) '*Android Studio Tutorial 6: Media Player*' Available at: https://www.youtube.com/watch?v=wFpJgVBqNC0&t=149s (Last Access : 3 December 2024)

Caleb Curry (2020) '*Android App Development in Java All-in-One Tutorial Series (4 HOURS!)*' , YouTube Available at: https://www.youtube.com/watch?v=tZvjSl9dswg&t=1321s (Last Access : 9 December 2024)

OpenAI (2024) 'Discussion on Android Development', ChatGPT. Available at: https://chat.openai.com (Accessed in between: 03 December 2024 – 05 January 2025)

Kazimoglu, Cain. (no date) *User Interface (UI) Components* University of West London. Available at: file:///C:/Users/mosma/AppData/Local/Temp/OneNote/16.0/Exported/%7B9651C8A4-F82A-4E1A-A829-8F59996B092A%7D/NT/11/Mobile_W04.UI_Components_Part2%20(1).pdf (Last Accessed: 12 December 2024).

Stack Overflow (2018) 'How to solve error: "Resource IDs cannot be used in switch statement in Android library modules" Available at : https://stackoverflow.com/questions/51401999/how-to-solve-error-resource-ids-cannot-be-used-in-switch-statement-in-android (Last Accessed : 12 December 2024).

Kazimoglu, Cain, (no date). *User Interface (UI) Components – PART 1*. University of West London. Available at: file:///C:/Users/mosma/AppData/Local/Temp/OneNote/16.0/Exported/%7BBAED810A-49EF-4834-860A-E6B32633E366%7D/NT/1/Mobile_W03.UI_Components_Part1.pdf (Last Accessed : 12 December 2024]

Kazimoglu, Cain ,(no date). *Intents : Implicit and Explicit Intents* University of West London Available at:

file:///C:/Users/mosma/AppData/Local/Temp/OneNote/16.0/Exported/%7B16CB27D6-F7DB-46C1-9A73-90B24F72DC3C%7D/NT/15/Mobile_W05.Intents.pdf (Last Accessed : 04 January 2025)

Kazimoglu, Cain ,(no date). *Shared Preferences, Files, and SQL Statements (SQLite)* University of West London Available at: Mobile_W08.Files_JSON_SQL.pdf (Last Accessed : 04 January 2025)

Appendix (Back-End Code)

Task 1: Cute Icons App

```
package com.example.cute_icons_app;

import android.graphics.Color;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  private TextView iconS;
  private Button buttonD;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // Set the layout file
    setContentView(R.layout.activity_main);
    // Initialize the views
    iconS = findViewById(R.id.IconStrings);
    buttonD = findViewById(R.id.ButtonDefault);
    // Set up the listeners for the animal icons
    iconUI();
    // Set up the listener for the Clear Text button
    buttonD.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        ReturnToDefault();
      }
    });
  }

  private void iconUI() {
    // Pass the icon ID, text ID, color ID, and sound ID to each listener
    setIconListener(R.id.catID, R.string.catS, R.color.catC, R.raw.cat);
    setIconListener(R.id.dogID, R.string.dogS, R.color.dogC, R.raw.dog);
    setIconListener(R.id.birdID, R.string.birdS, R.color.birdC, R.raw.birds);
    setIconListener(R.id.elephantID, R.string.elephantS, R.color.elephantC, R.raw.elephant);
    setIconListener(R.id.cowID, R.string.cowS, R.color.cowC, R.raw.cow);
    setIconListener(R.id.horseID, R.string.horseS, R.color.horseC, R.raw.horse);
    setIconListener(R.id.duckID, R.string.duckS, R.color.duckC, R.raw.duck);
    setIconListener(R.id.lambID, R.string.lambS, R.color.lambC, R.raw.lamb);
    setIconListener(R.id.goatID, R.string.goatS, R.color.goatC, R.raw.goat);
    setIconListener(R.id.monkeyID, R.string.monkeyS, R.color.monkeyC, R.raw.monkey);
  }

  private void setIconListener(int cuteIcon, int stringText, int colour, int soundNoise) {
    ImageView icon;
    icon = findViewById(cuteIcon);
    icon.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        // Making the animation of shake when user clicks on the icon
        Animation smallShake = AnimationUtils.loadAnimation(MainActivity.this,
R.anim.shake);
        icon.startAnimation(smallShake);

        // Change the text and colour on the top when new icon clicked
        ChangeUI(stringText, colour);

        // Play sound of animals when user clicks on the icon
        MediaPlayer mediaPlayer = MediaPlayer.create(MainActivity.this, soundNoise);
        mediaPlayer.start();

        // Release resources once sound completes
        mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
          @Override
          public void onCompletion(MediaPlayer mp) {
            mediaPlayer.release();
          }
        });
      }
    });
  }

  private void ChangeUI(int stringResId, int colorResId) {
    // Change the text and its color
    iconS.setText(stringResId);
```

```
        iconS.setTextColor(getResources().getColor(colorResId));

        // Change the button's background color and make it visible
        buttonD.setBackgroundTintList(getResources().getColorStateList(colorResId));
        buttonD.setVisibility(View.VISIBLE);
    }

    private void ReturnToDefault() {
        // Reset the text and color
        iconS.setText(R.string.defaultS);
        iconS.setTextColor(Color.BLACK);

        // Reset the button to its default state and hide it
        buttonD.setVisibility(View.GONE);
        buttonD.setBackgroundTintList(getResources().getColorStateList(R.color.white));
    }
}
```

Task 2: The Gym Guide

```
package com.example.thegymguideapp;

import com.example.thegymguideapp.R;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Spinner exercises;
    private RadioGroup workouts;
    private TextView routine;
    private ImageView diagram;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialise Spinner
        exercises = findViewById(R.id.exerciseSP);
        ArrayAdapter<String> spinnerAdapter = new ArrayAdapter<>(
            MainActivity.this,
            android.R.layout.simple_spinner_item,
            new String[]{getString(R.string.defaultS)}
        );

spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
;
        exercises.setAdapter(spinnerAdapter);

        // Initialise RadioGroup, TextView, and ImageView
        workouts = findViewById(R.id.workoutsRG);
        routine = findViewById(R.id.infoTV);
        diagram = findViewById(R.id.diagramIV);

        // Listener for RadioGroup
        workouts.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
          @Override
          public void onCheckedChanged(RadioGroup split, int sameId) {
            String[] ListOfExercises;
            String workoutsInfo;

            // Match exercises and descriptions based on selected workouts
            if (sameId == R.id.pushRB) {
              ListOfExercises =
addDefaultPlaceholder(getResources().getStringArray(R.array.pushSA));
              workoutsInfo = getString(R.string.pushIS);
            } else if (sameId == R.id.pullRB) {
              ListOfExercises =
addDefaultPlaceholder(getResources().getStringArray(R.array.pullSA));
              workoutsInfo = getString(R.string.pullIS);
            } else if (sameId == R.id.legsRB) {
              ListOfExercises =
addDefaultPlaceholder(getResources().getStringArray(R.array.legsSA));
              workoutsInfo = getString(R.string.legsIS);
            } else if (sameId == R.id.cardioRB) {
              ListOfExercises =
addDefaultPlaceholder(getResources().getStringArray(R.array.cardioSA));
              workoutsInfo = getString(R.string.cardioIS);
            } else {
              ListOfExercises = new String[]{getString(R.string.defaultS)};
              workoutsInfo = getString(R.string.defaultIS);
            }

            // Update Spinner with selected exercises
            ArrayAdapter<String> spinnerAdapter = new ArrayAdapter<>(
                MainActivity.this,
                android.R.layout.simple_spinner_item,
                ListOfExercises
            );

spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
;
            exercises.setAdapter(spinnerAdapter);

            // Display the workout information
            routine.setText(workoutsInfo);

            // Reset Spinner to default selection
            exercises.setSelection(0);
          }
        });

        // Listener for Spinner
        exercises.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
          @Override
          public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
            String ePicked = (String) parent.getItemAtPosition(position);
            String eInfo;
            int diagramResId = 0;

            // Connect exercise picked with information and image
            if (ePicked.equals(getString(R.string.defaultS))) {
              eInfo = getString(R.string.defaultIS);
            } else if (ePicked.equals("Bench Press (Chest)")) {
              eInfo = getString(R.string.bench_pressSR);
              diagramResId = R.drawable.bench_press;
            } else if (ePicked.equals("Overhead Press (Shoulders)")) {
              eInfo = getString(R.string.overhead_pressSR);
              diagramResId = R.drawable.overhead_press;
            } else if (ePicked.equals("Push-Down (Triceps)")) {
              eInfo = getString(R.string.push_downSR);
              diagramResId = R.drawable.push_down;
            } else if (ePicked.equals("Pull-Down (Back)")) {
              eInfo = getString(R.string.pull_downSR);
              diagramResId = R.drawable.pull_down;
            } else if (ePicked.equals("Bicep Curls (Biceps)")) {
              eInfo = getString(R.string.bicep_curlsSR);
              diagramResId = R.drawable.bicep_curls;
            } else if (ePicked.equals("Hammer Curls (Forearms)")) {
              eInfo = getString(R.string.hammer_curlsSR);
              diagramResId = R.drawable.hammer_curl;
            } else if (ePicked.equals("Leg Press (Quadriceps)")) {
              eInfo = getString(R.string.leg_pressSR);
              diagramResId = R.drawable.leg_press;
```

```
            } else if (ePicked.equals("Romanian Dead-lift (Hamstrings)")) {
                eInfo = getString(R.string.romanian_dead_liftSR);
                diagramResId = R.drawable.romanian_deadlift;
            } else if (ePicked.equals("Calf Raises (Calves)")) {
                eInfo = getString(R.string.calf_raisesSR);
                diagramResId = R.drawable.calf_raises;
            } else if (ePicked.equals("Running")) {
                eInfo = getString(R.string.runningSR);
                diagramResId = R.drawable.running;
            } else if (ePicked.equals("Cycling")) {
                eInfo = getString(R.string.cyclingSR);
                diagramResId = R.drawable.cycling;
            } else if (ePicked.equals("Skipping Rope")) {
                eInfo = getString(R.string.skipping_ropeSR);
                diagramResId = R.drawable.skipping_rope;
            } else {
                eInfo = getString(R.string.defaultIS);
            }

            // Update TextView with exercise info
            routine.setText(eInfo);

            // Update ImageView with exercise diagram
            if (diagramResId != 0) {
                diagram.setImageResource(diagramResId);
            } else {
                diagram.setImageDrawable(null); // Clear image if no image is found
            }
        }
        //Handle nothing selected
        @Override
        public void onNothingSelected(AdapterView<?> parent) {
            routine.setText(getString(R.string.defaultIS));
            diagram.setImageDrawable(null); // Clear image
        }
    });

    // Submit Button
    findViewById(R.id.submitB).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            int selectedId = workouts.getCheckedRadioButtonId();
            String workoutName;

            // Determine selected workout
            if (selectedId == R.id.pushRB) {
                workoutName = getString(R.string.pushS);
            } else if (selectedId == R.id.pullRB) {
                workoutName = getString(R.string.pullS);
            } else if (selectedId == R.id.legsRB) {
                workoutName = getString(R.string.legS);
            } else if (selectedId == R.id.cardioRB) {
                workoutName = getString(R.string.cardioS);
            } else {
                workoutName = getString(R.string.defaultS);
            }

            String selectedExercise = exercises.getSelectedItem().toString();
            showToast(workoutName + ": " + selectedExercise);
        }
    });

    // Clear Button
    findViewById(R.id.clearB).setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            workouts.clearCheck();
            exercises.setAdapter(new ArrayAdapter<>(
                    MainActivity.this,
                    android.R.layout.simple_spinner_item,
                    new String[]{getString(R.string.defaultS)}
            ));
            routine.setText(getString(R.string.defaultIS)); // Clear description
            diagram.setImageDrawable(null); // Clear image
        }
    });
  }

  private void showToast(String text) {
    Toast.makeText(this, text, Toast.LENGTH_LONG).show();
  }

  // Add a default placeholder to exercise array - ChatGPT wrote this part
  private String[] addDefaultPlaceholder(String[] exercises) {
    String[] updatedExercises = new String[exercises.length + 1];
    updatedExercises[0] = getString(R.string.defaultS);
    System.arraycopy(exercises, 0, updatedExercises, 1, exercises.length);
    return updatedExercises;
  }
}
```

Task 3 : The Man City Merchandise Ordering

```
package com.example.themancitymerchandiseorderapp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // Checkbox variables
    private CheckBox Tshirt, Jumper, Shorts, Scarf, Socks, Gloves, Football, Bag;

    private LinearLayout images;
    private TextView total;
    private Button confirm, clear;

    private final int Dcost = 5;
    private int totalCost;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize checkboxes
        Tshirt = findViewById(R.id.tShirtCB);
        Jumper = findViewById(R.id.jumperCB);
        Shorts = findViewById(R.id.shortsCB);
        Scarf = findViewById(R.id.scarfCB);
        Socks = findViewById(R.id.socksCB);
        Gloves = findViewById(R.id.glovesCB);
        Football = findViewById(R.id.footballCB);
        Bag = findViewById(R.id.bagCB);

        images = findViewById(R.id.imagesLL);
        total = findViewById(R.id.totalTV);
        confirm = findViewById(R.id.confirmB);
        clear = findViewById(R.id.clearB);

        // Set the initial total cost
        totalCost = Dcost;
        updateTotalCost();

        // Set listeners for buttons
        confirm.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                confirmOrder();
            }
        });

        clear.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                clearOrder();
            }
        });
    }

    // Method for checkbox
    public void onCheckboxClicked(View v) {
        boolean isChecked = ((CheckBox) v).isChecked();
        int cost = 0;
        int image = 0;

        if (v.getId() == R.id.tShirtCB) {
            cost = 60;
            image = R.drawable.tshirt;
        } else if (v.getId() == R.id.shortsCB) {
            cost = 20;
            image = R.drawable.shorts;
        } else if (v.getId() == R.id.socksCB) {
            cost = 10;
            image = R.drawable.socks;
        } else if (v.getId() == R.id.footballCB) {
            cost = 10;
            image = R.drawable.ball;
        } else if (v.getId() == R.id.jumperCB) {
            cost = 40;
            image = R.drawable.jumper;
        } else if (v.getId() == R.id.scarfCB) {
            cost = 15;
            image = R.drawable.scarf;
        } else if (v.getId() == R.id.glovesCB) {
            cost = 8;
            image = R.drawable.gloves;
        } else if (v.getId() == R.id.bagCB) {
            cost = 20;
            image = R.drawable.bag;
        }

        if (isChecked) {
            addImageToLayout(image);
            totalCost += cost;
        } else {
            removeImageFromLayout(image);
            totalCost -= cost;
        }
        updateTotalCost();
    }

    // Method to add images dynamically
    private void addImageToLayout(int imageResource) {
        ImageView imageView = new ImageView(this);
        imageView.setImageResource(imageResource);
        imageView.setTag(imageResource); // Use tag to identify image
        imageView.setLayoutParams(new LinearLayout.LayoutParams(150, 150));
        images.addView(imageView);
    }

    // Method to remove images
    private void removeImageFromLayout(int imageResource) {
        for (int i = 0; i < images.getChildCount(); i++) {
            ImageView imageView = (ImageView) images.getChildAt(i);
            if ((int) imageView.getTag() == imageResource) {
                images.removeView(imageView);
                break;
            }
        }
    }

    // Update total cost text
    private void updateTotalCost() {
        total.setText(getString(R.string.totalS, totalCost));
    }


    // Confirm order toast message
    private void confirmOrder() {
        Toast.makeText(this, getString(R.string.toastCS) + " Total: £" + totalCost,
Toast.LENGTH_LONG).show();
    }

    // Clear order: reset checkboxes, images, and cost
    private void clearOrder() {
        resetCheckboxes();
        images.removeAllViews();
        totalCost = Dcost;
        updateTotalCost();
    }
```

```
  // Helper method to uncheck all checkboxes
  private void resetCheckboxes() {
    Tshirt.setChecked(false);
    Jumper.setChecked(false);
    Shorts.setChecked(false);
    Scarf.setChecked(false);
    Socks.setChecked(false);
    Gloves.setChecked(false);
    Football.setChecked(false);
    Bag.setChecked(false);
  }
}
```

Task 4: Fab 4 in Test Cricket

```
package com.example.fab4intestcricket;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  private Button swipeRightButton;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.horizontal_scroll_view);

    swipeRightButton = findViewById(R.id.swipeRightButton);

    if (swipeRightButton != null) {
      swipeRightButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
          navigateToPlayer(0);
        }
      });
    }
  }

  private void navigateToPlayer(int playerIndex) {
    Intent intent = new Intent(MainActivity.this, PlayerActivity.class);
    intent.putExtra("playerIndex", playerIndex);
    startActivity(intent);
  }
}
```

```
package com.example.fab4intestcricket;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class PlayerActivity extends AppCompatActivity {

  // Define a Player class to encapsulate player data
  static class Player {
    int nameResId, imageResId, descriptionResId, statsResId, achievementsResId;

    Player(int name, int image, int description, int stats, int achievements) {
      this.nameResId = name;
      this.imageResId = image;
      this.descriptionResId = description;
      this.statsResId = stats;
      this.achievementsResId = achievements;
    }
  }

  private static final Player[] players = {
      new Player(R.string.root_name, R.drawable.root, R.string.root_description,
R.string.root_stats, R.string.root_achievements),
      new Player(R.string.smith_name, R.drawable.smith, R.string.smith_description,
R.string.smith_stats, R.string.smith_achievements),
      new Player(R.string.kane_name, R.drawable.williamson, R.string.kane_description,
R.string.kane_stats, R.string.kane_achievements),
      new Player(R.string.kohli_name, R.drawable.kohli, R.string.kohli_description,
R.string.kohli_stats, R.string.kohli_achievements)
  };

  private int playerIndex;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Retrieve the player's index passed via Intent
    playerIndex = getIntent().getIntExtra("playerIndex", 0);
```

```
        // Set the layout for the corresponding player
        int[] layouts = {R.layout.joe_root, R.layout.steve_smith, R.layout.kane_williamson,
R.layout.virat_kohli};
        if (playerIndex >= 0 && playerIndex < layouts.length) {
            setContentView(layouts[playerIndex]);
        } else {
            setContentView(R.layout.default_screen);
        }

        // Populate the screen with player details
        setPlayerDetails();

        // Initialize navigation buttons and bottom image click listeners
        setupNavigation();
    }

    private void setPlayerDetails() {
        Player currentPlayer = players[playerIndex];

        TextView nameTextView = findViewById(R.id.playerNameTextView);
        ImageView playerImageView = findViewById(R.id.playerImageView);
        TextView descriptionTextView = findViewById(R.id.playerDescriptionTextView);
        TextView statsTextView = findViewById(R.id.playerStatsTextView);
        TextView achievementsTextView = findViewById(R.id.playerAchievementsTextView);

        if (nameTextView != null) nameTextView.setText(currentPlayer.nameResId);
        if (playerImageView != null) playerImageView.setImageResource(currentPlayer.imageResId);
        if (descriptionTextView != null) descriptionTextView.setText(currentPlayer.descriptionResId);
        if (statsTextView != null) statsTextView.setText(currentPlayer.statsResId);
        if (achievementsTextView != null)
achievementsTextView.setText(currentPlayer.achievementsResId);
    }

    private void setupNavigation() {
        Button previousButton = findViewById(R.id.previousButton);
        Button nextButton = findViewById(R.id.nextButton);

        // Set listeners for navigation buttons if they exist in the layout
        setNavigationListener(previousButton, playerIndex - 1);
        setNavigationListener(nextButton, playerIndex + 1);

        // Setup click listeners for bottom navigation images
        setupBottomImageNavigation();
    }

    private void setNavigationListener(Button button, final int targetIndex) {
        if (button != null && targetIndex >= 0 && targetIndex < players.length) {
            button.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
            public void onClick(View v) {
                navigateToPlayer(targetIndex);
            }
        });
    }
}

private void setupBottomImageNavigation() {
    int[] imageIds = {R.id.imageRoot, R.id.imageSmith, R.id.imageWilliamson, R.id.imageKohli};

    for (int i = 0; i < imageIds.length; i++) {
        final int index = i;
        ImageView imageView = findViewById(imageIds[i]);

        if (imageView != null) {
            imageView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    navigateToPlayer(index);
                }
            });
        }
    }
}

private void navigateToPlayer(final int index) {
    Intent intent = new Intent(PlayerActivity.this, PlayerActivity.class);
    intent.putExtra("playerIndex", index);
    startActivity(intent);
    finish();
}
}
```

Task 5: CryptoStocks Login

```
package com.example.cryptostocksloginapp;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {

  private static final String PREFS_NAME = "CryptoStockPrefs";

  private EditText usET; //username
  private EditText paET; //password
  private Button loglB; //login button
  private TextView suTV; //singup

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.login);

    initializeComponentsAndListeners();
  }

  private void initializeComponentsAndListeners() {
    // Linking UI elements with Java variables
    usET = findViewById(R.id.usernameET);
    paET = findViewById(R.id.passwordET);
    loglB = findViewById(R.id.loginB);
    suTV = findViewById(R.id.signupTV);

    // Assigning hints
    usET.setHint(getString(R.string.usernameS));
    paET.setHint(getString(R.string.passwordS));
    loglB.setText(getString(R.string.loginS));
    suTV.setText(getString(R.string.signupS));


    loglB.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        handleLogin();
      }
    });

    suTV.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        navigateToSignup();
      }
    });
  }

  private void handleLogin() {
    // User input
```

```
        String uN = usET.getText().toString().trim();
        String pW = paET.getText().toString().trim();

        if (uN.isEmpty() || pW.isEmpty()) {
            displayMessage("Please fill in all fields."); // Custom method for toasts
            return;
        }

        // Get the stored password and compare
        SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
        String storedPassword = prefs.getString(uN + "_password", null);

        if (storedPassword != null && storedPassword.equals(pW)) {
            navigateToWelcome(uN);
        } else {
            displayMessage(getString(R.string.invalid_username_password));
        }
    }

    private void navigateToSignup() {
        // Direct navigation to RegistrationActivity
        Intent sUIntent = new Intent(MainActivity.this, RegistrationActivity.class);
        startActivity(sUIntent);
    }

    private void navigateToWelcome(String username) {
        // Direct navigation to WelcomeScreen
        Intent wWIntent = new Intent(MainActivity.this, WelcomeScreen.class);
        wWIntent.putExtra("username", username);
        startActivity(wWIntent);
        finish();
    }

    private void displayMessage(String message) {

        Toast.makeText(this, message, Toast.LENGTH_SHORT).show();
    }
}
```

```
package com.example.cryptostocksloginapp;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

public class RegistrationActivity extends AppCompatActivity {

  private EditText fullET, usernameET, passwordET, confirmET, niET;
  private RadioGroup genderRG;
  private Spinner birthSP;
  private Button registerB, clearB, backB;

  public static final String SHARED_PREFS = "CryptoStockPrefs";

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.registration);

    // Find views
    fullET = findViewById(R.id.fullET);
    usernameET = findViewById(R.id.usernameET);
    passwordET = findViewById(R.id.passwordET);
    confirmET = findViewById(R.id.confirmET);
    niET  = findViewById(R.id.niET);
    genderRG = findViewById(R.id.genderRG);
    birthSP = findViewById(R.id.birthSP);
    registerB = findViewById(R.id.registerB);
    clearB = findViewById(R.id.clearB);
    backB  = findViewById(R.id.backB);

    // Populate the Spinner with years from 1900 to the current year
    List<String> years = new ArrayList<>();
    years.add("Select"); // Placeholder item
    int currentYear = Calendar.getInstance().get(Calendar.YEAR);
    for (int year = 1950; year <= currentYear; year++) {
      years.add(String.valueOf(year));
    }

    ArrayAdapter<String> adapter = new ArrayAdapter<>(
        this, android.R.layout.simple_spinner_item, years);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    birthSP.setAdapter(adapter);
```

```
        // Register button
        registerB.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String fullName = fullET.getText().toString().trim();
                String username = usernameET.getText().toString().trim();
                String password = passwordET.getText().toString();
                String confirmPw = confirmET.getText().toString();
                String nin  = niET.getText().toString().trim();
                String year = birthSP.getSelectedItem().toString();
                int selectedGID = genderRG.getCheckedRadioButtonId();

                //validations:
                if (fullName.isEmpty() || username.isEmpty() || password.isEmpty() ||
confirmPw.isEmpty() || nin.isEmpty() || selectedGID == -1 ||
                        year.equals("Select"))
                {
                    Toast.makeText(RegistrationActivity.this,
                            "Please fill in all fields.",
                            Toast.LENGTH_SHORT).show();
                    return;
                }
                if (!password.equals(confirmPw)) {
                    Toast.makeText(RegistrationActivity.this,
                            "Passwords do not match!",
                            Toast.LENGTH_SHORT).show();
                    return;
                }

                // Figure out the gender text
                RadioButton selectedRadio = findViewById(selectedGID);
                String genderText = selectedRadio.getText().toString();

                // Save to SharedPreferences
                SharedPreferences prefs = getSharedPreferences(SHARED_PREFS, MODE_PRIVATE);
                SharedPreferences.Editor editor = prefs.edit();

                editor.putString(username + "_fN", fullName);
                editor.putString(username + "_gT",  genderText);
                editor.putString(username + "_y",    year);
                editor.putString(username + "_nin",     nin);

                // Also store the password so that login can check it
                editor.putString(username + "_password", password);

                editor.apply();
        Toast.makeText(RegistrationActivity.this,
            "Registration successful!",
            Toast.LENGTH_SHORT).show();

        // Return to login (MainActivity)
        Intent intent = new Intent(RegistrationActivity.this, MainActivity.class);
        startActivity(intent);
        finish();
      }
    });

    // Clear form button
    clearB.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        fullET.setText("");
        usernameET.setText("");
        passwordET.setText("");
        confirmET.setText("");
        niET.setText("");
        genderRG.clearCheck();
        birthSP.setSelection(0);
        Toast.makeText(RegistrationActivity.this,
            "Form cleared!",
            Toast.LENGTH_SHORT).show();
      }
    });

    // Back button
    backB.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        finish(); // Closes this screen
      }
    });
  }
}
```

```
package com.example.cryptostocksloginapp;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.view.View;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class WelcomeScreen extends AppCompatActivity {

    private static final String PREFS_NAME = "CryptoStockPrefs";

    private TextView welcomeMessage;
    private TextView detailsHeader;
    private TextView userDetails;
    private Button logoutButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.welcome);


        initViews();

        // Retrieve username from the intent
        String username = getIntent().getStringExtra("username");
        if (username == null || username.isEmpty()) {
            username = getString(R.string.usernameS); //  For missing username
        }

        // Load and display user details
        loadAndDisplayUserDetails(username);

        // Setup logout button
        setupLogoutButton();
    }

    private void initViews() {
        welcomeMessage = findViewById(R.id.welcomeTV);
        detailsHeader = findViewById(R.id.detailsHeaderTV);
        userDetails = findViewById(R.id.userDetailsTV);
        logoutButton = findViewById(R.id.logoutB);
    }

    private void loadAndDisplayUserDetails(String username) {
        SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);

        // Fetch details from SharedPreferences
        String fullName = prefs.getString(username + "_fN", getString(R.string.full_nameS));
        String gender = prefs.getString(username + "_g", getString(R.string.select_genderS));
        String birthYear = prefs.getString(username + "_y", getString(R.string.birthS));
        String nin = prefs.getString(username + "_nin", getString(R.string.ninS));

        // Calculate age if possible
    String age = calculateAge(birthYear);

    // Update welcome message and details
    welcomeMessage.setText(getString(R.string.welcomeS) + " " + fullName + "!");
    userDetails.setText(getString(R.string.user_details_placeholder, fullName, gender, age,
nin));
  }

  private String calculateAge(String birthYear) {
    int currentYear =
android.icu.util.Calendar.getInstance().get(android.icu.util.Calendar.YEAR);
    try {
      int yearOfBirth = Integer.parseInt(birthYear);
      int age = currentYear - yearOfBirth;
      return age > 0 ? String.valueOf(age) : getString(R.string.invalid_formS);
    } catch (NumberFormatException e) {
      return getString(R.string.invalid_formS);
    }
  }

  private void setupLogoutButton() {
    logoutButton.setText(getString(R.string.logoutS));
    logoutButton.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        Intent intent = new Intent(WelcomeScreen.this, MainActivity.class);
        startActivity(intent);
        finish(); // Close the WelcomeScreen
      }
    });
  }
}
```

Task 6: Athlete Connect Contact

```
package com.example.athleteconnectcontacts;

import android.os.Bundle;
import android.widget.SearchView;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {

  private RecyclerView rV;
```

```
    private AthleteAdapter aA;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);


        rV = findViewById(R.id.recyclerView);
        SearchView searchBar = findViewById(R.id.searchView);

        // Set up RecyclerView
        List<Athlete> athleteData = getAthleteData();
        aA = new AthleteAdapter(this, athleteData);
        rV.setLayoutManager(new LinearLayoutManager(this));
        rV.setAdapter(aA);

        // Implement search functionality
        searchBar.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
            @Override
            public boolean onQueryTextSubmit(String query) {
                return false; // Ignore submit action
            }

            @Override
            public boolean onQueryTextChange(String newText) {
                aA.filterByName(newText); // Filter on text change
                return true;
            }
        });
    }

    // Load initial athlete data
    private List<Athlete> getAthleteData() {
        List<Athlete> athletes = new ArrayList<>();
        athletes.add(new Athlete(getString(R.string.cristianoS), getString(R.string.ronaldoNS),
getString(R.string.ronaldoES), getString(R.string.ronaldoPS), R.drawable.ronaldo));
        athletes.add(new Athlete(getString(R.string.messiS), getString(R.string.messiNS),
getString(R.string.messiES), getString(R.string.messiPS), R.drawable.messi));
        athletes.add(new Athlete(getString(R.string.neymarS), getString(R.string.neymarNS),
getString(R.string.neymarES), getString(R.string.neymarPS), R.drawable.neymar));
        athletes.add(new Athlete(getString(R.string.giannisS), getString(R.string.giannisNS),
getString(R.string.giannisES), getString(R.string.giannisPS), R.drawable.giannis));
        athletes.add(new Athlete(getString(R.string.rootS), getString(R.string.rootNS),
getString(R.string.rootsES), getString(R.string.rootPS), R.drawable.root));
        athletes.add(new Athlete(getString(R.string.sirajS), getString(R.string.sirajNS),
getString(R.string.sirajES), getString(R.string.sirajPS), R.drawable.siraj));
        athletes.add(new Athlete(getString(R.string.leeS), getString(R.string.leeNS),
getString(R.string.leeES), getString(R.string.leePS), R.drawable.lee));
        athletes.add(new Athlete(getString(R.string.benzemaS), getString(R.string.benzemaNS),
getString(R.string.benzemaES), getString(R.string.benzemaPS), R.drawable.benzema));
        athletes.add(new Athlete(getString(R.string.khabibS), getString(R.string.khabibNS),
getString(R.string.khabibES), getString(R.string.khabibPS), R.drawable.khabib));
        athletes.add(new Athlete(getString(R.string.naseemS), getString(R.string.naseemNS),
getString(R.string.naseemES), getString(R.string.naseemPS), R.drawable.naseem));
        return athletes;
    }
}
```

```
package com.example.athleteconnectcontacts;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

// RecyclerView Adapter to handle Athlete items
public class AthleteAdapter extends RecyclerView.Adapter<AthleteAdapter.AthleteViewHolder>
{

    private List<Athlete> dL; // Displayed list
    private List<Athlete> cL; // Complete list
    private Context aC; // Application context

    public AthleteAdapter(Context context, List<Athlete> athletes) {
        this.aC = context;
        this.dL = new ArrayList<>(athletes); // Start with full data
        this.cL = new ArrayList<>(athletes);
    }

    @Override
    public AthleteViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        // Inflate list item layout
        View itemView = LayoutInflater.from(parent.getContext()).inflate(R.layout.list_item, parent,
false);
        return new AthleteViewHolder(itemView);
    }
```

```
@Override
public void onBindViewHolder(AthleteViewHolder holder, int position) {
    Athlete athlete = dL.get(position);

    // Populate ViewHolder with data
    holder.Fname.setText(athlete.getFName());
    holder.phoneNo.setText(athlete.getPhoneNo());
    holder.emailA.setText(athlete.getEmailA());
    holder.profession.setText(athlete.getProfession());
    holder.photo.setImageResource(athlete.getImageResId());

    // Open dialer when clicked
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent dialIntent = new Intent(Intent.ACTION_DIAL);
            dialIntent.setData(Uri.parse("tel:" + athlete.getPhoneNo()));
            aC.startActivity(dialIntent);
        }
    });
}

@Override
public int getItemCount() {
    return dL.size(); // Total visible items
}

// Filtering functionality
public void filterByName(String searchQuery) {
    dL.clear();
    if (searchQuery.isEmpty()) {
        dL.addAll(cL);
    } else {
        for (Athlete athlete : cL) {
            if (athlete.getFName().toLowerCase().contains(searchQuery.toLowerCase())) {
                dL.add(athlete);
            }
        }
    }
    notifyDataSetChanged(); // Refresh RecyclerView
}

// ViewHolder class for managing item views
public static class AthleteViewHolder extends RecyclerView.ViewHolder {
    TextView Fname, phoneNo, emailA, profession;
    ImageView photo;
```

```
        public AthleteViewHolder(View itemView) {
            super(itemView);

            // Link layout elements to ViewHolder variables
            Fname = itemView.findViewById(R.id.nameTV);
            phoneNo = itemView.findViewById(R.id.phoneTV);
            emailA = itemView.findViewById(R.id.emailTV);
            profession = itemView.findViewById(R.id.professionTV);
            photo = itemView.findViewById(R.id.athleteIV);
        }
    }
}
```

```
package com.example.athleteconnectcontacts;

// All athlete infomation
public class Athlete {

    private String name;
    private String contactNumber;
    private String emailId;
    private String profession;
    private int imageId;

    public Athlete(String name, String contactNumber, String emailId, String profession, int
imageId) {
        this.name = name;
        this.contactNumber = contactNumber;
        this.emailId = emailId;
        this.profession = profession;
        this.imageId = imageId;
    }
//Week 8
    public String getFName() {
        return name;
    }

    public String getPhoneNo() {
        return contactNumber;
    }

    public String getEmailA() {
        return emailId;
    }

    public String getProfession() {
        return profession;
    }
```

```
  public int getImageResId() {
    return imageId;
  }
}
```

Task 7 : TravelNow

```
package com.example.travelnowapp;

import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.example.travelnowapp.R;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
  private RecyclerView recyclerView;
  private List<City> cityList;
  private CityAdapter cityAdapter;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    initializeViews();
    setupRecyclerView();
    fetchAndParseJsonData();
  }

  private void initializeViews() {
    recyclerView = findViewById(R.id.recyclerView);
  }

  // Set up RecyclerView and Adapter
  private void setupRecyclerView() {
    cityList = new ArrayList<>();
    cityAdapter = new CityAdapter(this, cityList);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    recyclerView.setAdapter(cityAdapter);
  }
```

```
    // Fetch and Parse JSON Data
    private void fetchAndParseJsonData() {
        GetJsonData fetchData = new GetJsonData();
        fetchData.setOnFetchCompleteListener(new GetJsonData.OnFetchCompleteListener() {
            @Override
            public void onFetchComplete(String json) {
                parseJsonData(json);
            }
        });
        fetchData.execute();
    }

    // Parse JSON Data into City objects
    private void parseJsonData(String json) {
        try {
            if (json != null) {
                JSONArray jsonArray = new JSONArray(json);
                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject obj = jsonArray.getJSONObject(i);
                    City city = createCityFromJson(obj);
                    cityList.add(city);
                }
                cityAdapter.notifyDataSetChanged();
            }
        } catch (Exception e) {
            Log.e("MainActivity", "Error parsing JSON data", e);
        }
    }

    // Create a City object from JSON
    private City createCityFromJson(JSONObject obj) throws Exception {
        return new City(
                obj.getString("city"),
                obj.getString("country"),
                obj.getString("airline"),
                obj.getString("price"),
                obj.getString("image_url")
        );
    }
}
```

```
package com.example.travelnowapp;

import android.os.AsyncTask;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
```

```
public class GetJsonData extends AsyncTask<Void, Void, String> {
  private static final String JSON_URL = "https://api.jsonserve.com/A-FOpK";
  private OnFetchCompleteListener listener;

  // Interface for Fetch Complete Listener
  public interface OnFetchCompleteListener {
    void onFetchComplete(String json);
  }

  public void setOnFetchCompleteListener(OnFetchCompleteListener listener) {
    this.listener = listener;
  }

  @Override
  protected String doInBackground(Void... voids) {
    StringBuilder result = new StringBuilder();
    try {
      URL url = new URL(JSON_URL);
      HttpURLConnection connection = (HttpURLConnection) url.openConnection();

      BufferedReader reader = new BufferedReader(
          new InputStreamReader(connection.getInputStream())
      );

      String line;
      while ((line = reader.readLine()) != null) {
        result.append(line);
      }
      reader.close();
    } catch (Exception e) {
      e.printStackTrace();
    }
    return result.toString();
  }

  @Override
  protected void onPostExecute(String json) {
    if (listener != null) {
      listener.onFetchComplete(json);
    }
  }
}
```

```
package com.example.travelnowapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.travelnowapp.R;
import com.google.android.material.datepicker.MaterialDatePicker;
import com.google.android.material.datepicker.MaterialPickerOnPositiveButtonClickListener;

import androidx.core.util.Pair;

public class DatePickerActivity extends AppCompatActivity {

    private TextView travelDatesTV;
    private String datesSelectPS = "";
    private String cityPS;
    private String countryPS;
    private String airlinePS;
    private String pricePS;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_date_picker);

        // get indent data
        Intent intent = getIntent();
        cityPS = intent.getStringExtra("cityName");
        countryPS = intent.getStringExtra("country");
        airlinePS = intent.getStringExtra("airline");
        pricePS = intent.getStringExtra("price");


        travelDatesTV = findViewById(R.id.select_travel_dates);

        // Set default placeholder
        setDefaultDateRange();

        // Set onClickListener for travel dates selection
        travelDatesTV.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showDatePicker();
            }
        });
    }

    private void setDefaultDateRange() {
        datesSelectPS = getString(R.string.placeholder_date_range);
        travelDatesTV.setText(datesSelectPS);
    }

    private void showDatePicker() {
        // Build the Material Date Picker
        MaterialDatePicker.Builder<Pair<Long, Long>> builder =
MaterialDatePicker.Builder.dateRangePicker();
        builder.setTitleText(R.string.date_range_picker_title);
        MaterialDatePicker<Pair<Long, Long>> materialDatePicker = builder.build();

        // Show the Date Picker
        materialDatePicker.show(getSupportFragmentManager(), "DATE_PICKER");

        // Handle positive selection
        materialDatePicker.addOnPositiveButtonClickListener(new
MaterialPickerOnPositiveButtonClickListener<Pair<Long, Long>>() {
            @Override
            public void onPositiveButtonClick(Pair<Long, Long> selection) {
                datesSelectPS = materialDatePicker.getHeaderText();
                travelDatesTV.setText(datesSelectPS);
                showConfirmationDialog();
            }
        });

        // Handle cancellation
        materialDatePicker.addOnNegativeButtonClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(DatePickerActivity.this, R.string.date_range_no_selection,
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void showConfirmationDialog() {
        String tripDetails = getString(R.string.trip_details_format, cityPS, countryPS, airlinePS,
pricePS, datesSelectPS);

        new android.app.AlertDialog.Builder(this)
                .setTitle(R.string.trip_reserved_successfully)
                .setMessage(tripDetails)
                .setPositiveButton(R.string.proceed_to_checkout, new
android.content.DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(android.content.DialogInterface dialog, int which) {
                        navigateToCheckout(tripDetails);
                    }
                })
            .setNegativeButton(R.string.continue_browsing, null)
            .show();
    }

    private void navigateToCheckout(String tripDetails) {
        Intent intent = new Intent(DatePickerActivity.this, CheckoutActivity.class);
        intent.putExtra("tripDetails", tripDetails);
        startActivity(intent);
    }
}
```

```
package com.example.travelnowapp;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;


import java.util.List;

public class CityAdapter extends RecyclerView.Adapter<CityAdapter.CityViewHolder> {
    private List<City> cityList;
    private Context context;

    // Constructor
    public CityAdapter(Context context, List<City> cityList) {
        this.context = context;
        this.cityList = cityList;
    }

    @NonNull
    @Override
    public CityViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.city_item, parent, false);
        return new CityViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull CityViewHolder holder, int position) {
        final City city = cityList.get(position);
        // Bind city details to the views
        holder.tvCityName.setText(city.getAreaPS());
        holder.tvCityDetails.setText(String.format("%s - %s (%s)", city.getCountryPS(),
city.getFlyPS(), city.getMoneyPS()));

        // Load image using Glide
        Glide.with(context)
                .load(city.getImageUrlPS())
                .placeholder(R.drawable.placeholder_image)
                .into(holder.ivCityImage);

        // Handle the "Book Now" button click
        holder.btnBookNow.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Pass city details to DatePickerActivity
                Intent intent = new Intent(context, DatePickerActivity.class);
                intent.putExtra("cityName", city.getAreaPS());
                intent.putExtra("country", city.getCountryPS());
                intent.putExtra("airline", city.getFlyPS());
                intent.putExtra("price", city.getMoneyPS());
                context.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() {
        return cityList.size();
    }

    // ViewHolder class
    static class CityViewHolder extends RecyclerView.ViewHolder {
        TextView tvCityName, tvCityDetails;
        ImageView ivCityImage;
        Button btnBookNow;

        public CityViewHolder(@NonNull View itemView) {
            super(itemView);

            // Initialize views
            tvCityName = itemView.findViewById(R.id.tvCityName);
            tvCityDetails = itemView.findViewById(R.id.tvCityDetails);
            ivCityImage = itemView.findViewById(R.id.ivCityImage);
            btnBookNow = itemView.findViewById(R.id.btnBookNow);
        }
```

```
  }
}
```

```
package com.example.travelnowapp;

public class City {
  private String areaPS;
  private String CountryPS;
  private String flyPS;
  private String moneyPS;
  private String imageUrlPS;

  public City(String cityName, String country, String airline, String price, String imageUrl) {
    this.areaPS = cityName;
    this.CountryPS = country;
    this.flyPS = airline;
    this.moneyPS = price;
    this.imageUrlPS = imageUrl;
  }

  public String getAreaPS() {
    return areaPS;
  }

  public String getCountryPS() {
    return CountryPS;
  }

  public String getFlyPS() {
    return flyPS;
  }

  public String getMoneyPS() {
    return moneyPS;
  }

  public String getImageUrlPS() {
    return imageUrlPS;
  }
}
```

```
package com.example.travelnowapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class CheckoutActivity extends AppCompatActivity {

    private TextView detailsTripPTV;
    private Button backPB;
    private Button stopPB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_checkout);

        initializeViews();
        displayTripDetails();
        setupButtonListeners();
    }

    private void initializeViews() {
        detailsTripPTV = findViewById(R.id.tvTripDetails);
        backPB = findViewById(R.id.btnGoBack);
        stopPB = findViewById(R.id.stopAppB);
    }

    private void displayTripDetails() {
        Intent show = getIntent();
        String tripDetails = show.getStringExtra("tripDetails");

        if (tripDetails != null) {
            detailsTripPTV.setText(tripDetails);
        } else {
            detailsTripPTV.setText(getString(R.string.no_booking_details));
        }
    }

    private void setupButtonListeners() {
        backPB.setOnClickListener(createGoBackClickListener());
        stopPB.setOnClickListener(createCloseAppClickListener());
    }

    private View.OnClickListener createGoBackClickListener() {
        return new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent link = new Intent(CheckoutActivity.this, MainActivity.class);
                startActivity(link);
                finish();
            }
    };
  }

  private View.OnClickListener createCloseAppClickListener() {
    return new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        finishAffinity(); // Closes the app
      }
    };
  }
}
```