

Student Name	Date	Number of Hours	Description of Work Done
Russell Han Josef	Jan 18, 2025	1	Planning on the project scope and role distribution.
Russell Han Josef	Jan 20, 2025	1.5	Research on screens to have. Finding screens inspiration. App name and branding.
Russell Han Josef	Jan 22, 2025	2	Research on Heroku Dynos and Postgres workflow
Russell Han Josef	Jan 22, 2025	1	Researched about conducting surveys and user testing.
Russell Han Josef	Jan 22, 2025	1	Created survey questionnaires and shared using Microsoft Forms. -> Github Repo -> Misc -> Form_Exploration App Survey_ Help Us Build Your Dream Travel Planner.pdf
Russell Han Josef	Jan 25, 2025	2	Commit Git for Proposal.pdf. Researched for APIs: OpenAI gpt model to use, Google Places, Google Maps, Eventbrite, OpenWeatherMap. Get all the API Keys needed.
Russell Han Josef	Jan 28, 2025	0.5	Get Free credits from Heroku account using GitHub for Student Developer Pack
Russell Han Josef	Jan 28, 2025	2	Researched on Database Schema. Learned more about PostgreSQL vs Microsoft SQL vs MySQL. Then Firebase NoSQL. Learned about Hybrid Architecture Approach with both PostgreSQL and NoSQL. Exported 2 diagrams. -> Github Repo -> Misc -> WayPoint-SQL-Schema.png -> Github Repo -> Misc -> WayPoint-NoSQL-Schema.png
Russell Han Josef	Jan 28, 2025	1	Created Video to share surveys on Instagram. Link: https://www.instagram.com/reel/DFXOyODRLOI/?igsh=ZnF4OWhqbTk0NnVx
Russell Han Josef	Jan 29, 2025	1	Get more insights from Prof. Priya in how do we approach surveys collected and user testing if user doesn't reside in a same country.
Russell Han Josef	Feb 03, 2025	1.5	Created backend shell. Tested the shell and it's running. Pushed to GitHub repo.
Russell Han Josef	Feb 04, 2025	1	Frontend shell wasn't working on Russell's machine. Troubleshooting. Cocoapods installed but pod wasn't installed successfully. Finding: XCode wasn't install properly. When running xcode-select -p on terminal. It showed other thing than "/Applications/Xcode.app/Contents/Developer" Details: The problem was that the path to the Xcode command-line tools was not correctly set, causing the xcrun command to be unable to locate the iOS SDK (iphoneos). This resulted in the error message: SDK "iphoneos" cannot be located. Code to run: sudo xcode-select -s /Applications/Xcode.app/Contents/Developer
Russell Han Josef	Feb 04, 2025	2.5	I encountered an issue while deploying my FastAPI app to Heroku. The deployment failed with a ModuleNotFoundError for the backend module. After investigating, I realized that the folder containing the app was named Implementation with a capital "I", but the Procfile was referencing it as implementation with a lowercase "i". Instead of renaming the folder, I updated the Procfile to correctly reference the folder name with the capital "I" as it appeared in the project. After this update, the deployment was successful, and the app was properly hosted on Heroku.
Russell Han Josef	Feb 04, 2025	1	Problem: Setting up PostgreSQL was challenging, especially connecting pgAdmin to both my local database and Heroku's remote database. I struggled with authentication issues, missing roles, and ensuring my tables were correctly created in both environments. Solution: I configured my local PostgreSQL by setting the correct roles and connected pgAdmin to Heroku using the provided DATABASE_URL. I ensured the database schema was consistent across both environments and created tables using SQLAlchemy. Explanation: This helped me understand how PostgreSQL differs locally and on Heroku, how to manage database credentials, and how to properly set up pgAdmin for database administration.
Russell Han Josef	Feb 04, 2025	1	Problem: Deploying to Heroku failed due to an incorrectly placed Procfile, missing dependencies in requirements.txt, and misconfigured environment variables like DATABASE_URL and SECRET_KEY. Solution: I moved Procfile to the root directory, updated requirements.txt, and set DATABASE_URL correctly in Heroku's environment variables. Restarting the Heroku dyno applied these fixes. Explanation: This taught me the importance of directory structure and configuration files in deployment and how to debug deployment failures using heroku logs --tail.

Russell Han Josef	Feb 04, 2025	2	<p>Problem: After fixing deployment, my app still crashed on Heroku (H10 App Crashed) due to SQLAlchemy not recognizing Heroku's DATABASE_URL format and FastAPI failing to bind to the correct port.</p> <p>Solution: I modified db.py to convert postgres:// to postgresql://, ensuring SQLAlchemy could connect. I also updated Procfile to bind FastAPI to Heroku's \$PORT.</p> <p>Explanation: This reinforced the differences between local and production environments, the need for dynamic configurations, and how Heroku manages deployments and environment variables.</p>
Russell Han Josef	Feb 09, 2025	1	Created travel style quiz. Total 7 questions. Scoring system using point-based system.
Russell Han Josef	Feb 11, 2025	3.5	<p>1. Foreign Key Dependency Issues Problem: Models had incorrect import order, causing foreign key errors. Solution: Adjusted import order in __init__.py to ensure dependencies load correctly.</p> <p>2. Circular Import Issue Problem: Importing Base from db.py led to circular dependencies. Solution: Moved Base to base.py and updated model imports.</p> <p>3. Missing email-validator Error Problem: FastAPI required email-validator, despite being in requirements.txt. Solution: Reinstalled dependencies manually on Heroku.</p> <p>4. uvicorn: command not found on Heroku Problem: uvicorn was missing in the runtime environment. Solution: Updated Procfile to use python -m uvicorn and verified installation.</p> <p>5. Heroku App Not Restarting Properly Problem: Deployment changes weren't reflecting. Solution: Restarted the app and purged Heroku build cache.</p> <p>6. App Not Binding to \$PORT Problem: FastAPI wasn't binding correctly to the environment port. Solution: Ensured uvicorn runs with --port=\${PORT} in Procfile.</p> <p>7. Database Connection Test Failed on Heroku Problem: Remote database connection wasn't verifying. Solution: Created /test-db endpoint and confirmed it works.</p>
Russell Han Josef	Feb 11, 2025	0.5	<p>Running db.py to Create Tables on Heroku: Problems & Solutions</p> <p>1. Running db.py on Heroku caused ModuleNotFoundError: No module named 'app' Fix: Used PYTHONPATH=. python app/db/db.py to ensure the correct module path.</p> <p>2. Tables were not appearing in Heroku Postgres after running db.py Fix: Explicitly set Base.metadata.schema = "public" in db.py to ensure tables are placed in the correct schema.</p> <p>3. Needed a way to manually trigger db.py on Heroku Fix: Opened a Heroku shell with heroku run bash -a waypoint-travel, then executed: PYTHONPATH=. python app/db/db.py</p> <p>4. Wanted to verify if tables were created in Heroku Postgres Fix: Used Heroku Postgres CLI to check tables: heroku pg:psql -a waypoint-travel SELECT tablename FROM pg_tables WHERE schemaname = 'public';</p> <p>Final Outcome: Successfully ran db.py on Heroku, ensuring tables were created in the correct schema.</p>

Russell Han Josef	Feb 12, 2025	1.5	<p>Summary of Fixes & Progress</p> <p>1. CRUD Implementation for Users</p> <p>Created POST /users → Create User (with password hashing).</p> <p>Created GET /users/{user_id} → Retrieve User by ID.</p> <p>Created PUT /users/{user_id} → Update User (name, email, password).</p> <p>Created DELETE /users/{user_id} → Delete User.</p> <p>2. Fixed Errors</p> <p>InvalidRequestError → Added ForeignKey("users.id") in quiz_model.py.</p> <p>TypeError: 'password' is an invalid keyword argument for User → Ensured password_hash is used in user_model.py.</p> <p>NameError: name 'user_schema' is not defined → Fixed incorrect import in user_routes.py.</p> <p>zsh: no matches found: passlib[bcrypt] → Installed using pip install "passlib[bcrypt]".</p> <p>3. Fixed Duplicate URL Path Issue</p> <p>Issue: "/users/users/{user_id}" in FastAPI /docs.</p> <p>Fix: Removed redundant /users prefix from routes in user_routes.py.</p> <p>4. Tested Locally</p> <p>Verified all CRUD operations using FastAPI /docs.</p> <p>Confirmed correct URL paths after fixing duplication.</p> <p>Next Steps</p> <p>Test CRUD operations for Itineraries, Places, Badges, and Quiz Results.</p> <p>Once confirmed, deploy to Heroku and re-test on live API.</p>
Russell Han Josef	Feb 12, 2025	1	<p>Log Summary for Places CRUD Implementation</p> <p>1. Implemented CRUD for Places</p> <p>Created POST /places → Add a new place.</p> <p>Created GET /places/{id} → Retrieve a place by ID.</p> <p>Created PUT /places/{id} → Update place details.</p> <p>Created DELETE /places/{id} → Remove a place.</p> <p>2. Fixed Issues</p> <p>Fixed timezone inconsistency → Ensured last_updated is stored in UTC.</p> <p>Resolved datetime.utcnow() deprecation warning → Used datetime.now(timezone.utc).replace(tzinfo=None).</p> <p>Verified timestamps consistency → Matched last_updated with created_at format.</p> <p>3. Successfully Tested</p> <p>POST /places → Verified place creation with manual data.</p> <p>GET /places/{id} → Retrieved created places correctly.</p> <p>PUT /places/{id} → Updated place details without timezone mismatch.</p> <p>DELETE /places/{id} → Successfully removed places from the database.</p> <p>Next Steps</p> <p>Implement User Favorites (user_favorite_routes.py).</p> <p>Ensure Users ↔ Places relationship works correctly.</p> <p>Prepare for Google Places API integration.</p>
Russell Han Josef	Feb 12, 2025	0.5	<p>Log Summary for User Favorites Implementation</p> <p>1. Implemented CRUD for User Favorites</p> <p>Created POST /user_favorites → Add a place to favorites.</p> <p>Created GET /user_favorites/{user_id} → Retrieve a user's favorite places.</p> <p>Created DELETE /user_favorites/{favorite_id} → Remove a favorite place.</p> <p>2. Fixed Issues</p> <p>Validated user and place existence before adding a favorite.</p> <p>Prevented duplicate favorites by checking existing records.</p> <p>Ensured added_at timestamp is stored in UTC for consistency.</p> <p>3. Successfully Tested</p> <p>POST /user_favorites → Added places to favorites successfully.</p> <p>GET /user_favorites/{user_id} → Retrieved correct favorites for users.</p> <p>DELETE /user_favorites/{favorite_id} → Removed favorites as expected.</p> <p>4. Updated main.py</p> <p>Included user_favorite_routes in FastAPI router.</p> <p>Next Steps</p> <p>Implement Badges (badge_routes.py).</p> <p>Ensure User ↔ Badges relationship works correctly.</p>

Russell Han Josef	Feb 12, 2025	1.5	<p>Set Up Google Places API Integration</p> <p>Chose Google Places API (Old Version) for simpler API key authentication. Tested API manually using Postman & cURL. Implemented FastAPI Route for Places Search</p> <p>Created /places/search endpoint to fetch nearby places. Integrated Google Places API (maps.googleapis.com). Cached results in PostgreSQL to reduce API calls. Restricted API to British Columbia (BC), Canada</p> <p>Implemented latitude/longitude boundary check to block requests outside BC. Verified restriction by testing New York (Successfully blocked). Error Handling & Optimizations</p> <p>Improved handling for invalid locations and API failures. Implemented database caching to avoid redundant API requests. Added X-Goog-FieldMask to optimize API responses. Tested & Debugged API Responses</p> <p>Verified working results for Vancouver, BC. Ensured API key security using environment variables (.env, Heroku Config Vars).</p>
Russell Han Josef	Feb 13, 2025	1.5	<p>Tasks Completed:</p> <ul style="list-style-type: none"> - Configured React Navigation with StackNavigator & BottomTabNavigator. - Created Login & Signup screens with placeholder values. - Implemented Profile screen with user details, Edit Profile (future), and Log Out. - Built Settings screen with Travel Style, Notifications, Language, and Account Management. - Used SafeAreaView & ScrollView to fix UI layout issues. - Replaced deprecated Picker with @react-native-picker/picker. - Ensured dynamic spacing to prevent overlap with iPhone Dynamic Island. <p>Next Steps:</p> <ul style="list-style-type: none"> - Implement Edit Profile feature. - Add form validation for Login & Signup. - Prepare backend integration for authentication and profile updates. - Enhance UI with better styling.
Russell Han Josef	Feb 13, 2025	0.75	<p>Tasks Completed:</p> <ul style="list-style-type: none"> - Backend Integration for Authentication: <ul style="list-style-type: none"> - Reviewed backend schemas, models, and routes. - Confirmed API endpoint for user registration (POST /users/). - Identified and fixed login API endpoint (POST /users/auth/login). - Updated SignupScreen.js: <ul style="list-style-type: none"> - Connected to backend (POST /users/) for user registration. - Handled form submission, API request, and error handling. - Added navigation to Login screen upon successful signup. - Updated LoginScreen.js: <ul style="list-style-type: none"> - Integrated POST /users/auth/login using query parameters. - Ensured login request matches the correct FastAPI route. - Redirects users to Main app upon successful login. - Displays alerts for errors and invalid credentials. - Backend API Testing & Debugging: <ul style="list-style-type: none"> - Successfully tested user registration and login via FastAPI. - Ensured POST /users/auth/login worked with query parameters. - Verified API response handling in React Native app. <p>Next Steps:</p> <ul style="list-style-type: none"> - Implement persistent authentication (store session/token). eg. JWT - Add form validation for signup & login fields. - Enhance UI styling & error messages for better user experience. - Implement Edit Profile feature in ProfileScreen.js.

Russell Han Josef	Feb 13, 2025	2.25	<p>Attempted to rename React Native app from "frontend" to "WayPoint" → Encountered issues, reverted to "frontend"</p> <p>Updated package.json and app.json to reflect the correct app name → Reverted due to build errors</p> <p>Checked and updated Xcode Signing & Capabilities → Used free Apple ID for provisioning</p> <p>Attempted to set correct Bundle Identifier for Firebase setup → Reverted due to build failures</p> <p>Installed Firebase dependencies (@react-native-firebase/app) → Successfully installed</p> <p>Fixed CocoaPods issues with modular headers → Modified Podfile and ran pod install --repo-update</p> <p>Configured Firebase in AppDelegate.swift → Updated to FirebaseApp.configure()</p> <p>Encountered xcodebuild error code 65 while running iOS build → Attempted multiple fixes</p> <p>Deleted and reinstalled CocoaPods, node_modules, and Xcode DerivedData → No success</p> <p>Manually deleted ios/build/ and cleaned Xcode project → Issue persisted</p> <p>Ran xcodebuild clean and pod install --repo-update → Did not resolve the issue</p> <p>Tried running the app via Metro Bundler (npx react-native run-ios) → Still failed</p> <p>Decided to fully reset the project by deleting and reinstalling all dependencies → Still encountering build issues</p>
Russell Han Josef	Feb 14, 2025	4	<p>Summary Log: Firebase Realtime Database Setup & Next Steps</p> <p>Problems:</p> <ul style="list-style-type: none"> Multiple React-Core dependencies causing conflicts React-RCTAppDelegate not linking correctly FirebaseAuth/FirebaseAuth-Swift.h file not found (even though not needed) Xcode build error: "unable to initiate PIF transfer session" ReactCommon module redefinition error CocoaPods installation issues <p>Solutions Attempted:</p> <ul style="list-style-type: none"> Refactored Podfile to use use_modular_headers! and fixed React-Core conflicts Updated AppDelegate.swift with FirebaseApp.configure() Removed and reinstalled dependencies (node_modules, Pods, Podfile.lock) Cleared Xcode cache (DerivedData, xcodebuild clean) Ensured only needed Firebase modules were installed <p>What's Next:</p> <ul style="list-style-type: none"> Start fresh to ensure a clean build Get React Native running first before adding Firebase Verify Podfile with default settings, then add Firebase Test a basic build (npx react-native run-ios) before integrating Firebase features Implement Firebase Realtime Database CRUD to confirm it works <p>Next attempt: Clean setup from the beginning</p>
Russell Han Josef	Feb 14, 2025	1.5	<p>Work Log: Firebase Integration in React Native (iOS)</p> <p>Problems & Solutions</p> <ul style="list-style-type: none"> Firebase Not Initializing (No Firebase App '[DEFAULT]' has been created) Firebase was not auto-detecting GoogleService-Info.plist. Manually initialized Firebase in firebase.js. Firebase connected successfully using manual config. Missing or Invalid FirebaseOptions property 'apiKey' Error Firebase could not find apiKey from GoogleService-Info.plist. Verified plist format and corrected key names. Ensured plist was inside Implementation/frontend/ios/. Linked plist in Xcode under Build Phases → Copy Bundle Resources. Still using manual config; plist auto-detection needs verification. Firebase Data Not Appearing in Realtime Database Firebase connection worked, but no data appeared. Updated Firebase database rules to allow reads/writes. Created a test function in LoginScreen.js to write data. Confirmed successful data write to Firebase Console. <p>Next Steps</p> <ul style="list-style-type: none"> Remove manual Firebase config and verify plist auto-detection. Fetch and display a list of data from Firebase. Secure Firebase Database rules based on authentication.

Russell Han Josef	Feb 14, 2025	0.5	<p>Work Log: Firebase Auto-Detection Fix in React Native (iOS)</p> <p>Problems & Solutions</p> <p>Firebase Not Initializing Automatically Firebase was not detecting GoogleService-Info.plist. Manually initialized Firebase in AppDelegate.swift. Confirmed Firebase auto-detection now works.</p> <p>Missing Firebase Setup in AppDelegate.swift React Native Firebase requires Firebase to be initialized in AppDelegate.swift. Added FirebaseApp.configure() inside didFinishLaunchingWithOptions. Restarted the app and confirmed successful Firebase initialization.</p> <p>Plist File Not Being Read by Xcode GoogleService-Info.plist was not linked in Build Phases → Copy Bundle Resources. Manually added the plist file in Xcode. Verified correct plist location in Implementation/frontend/ios/.</p> <p>Next Steps Fetch and display data from Firebase in the app. Secure Firebase database rules based on authentication.</p>
Russell Han Josef	Feb 22, 2025	2	<p>Fixes & Improvements in SettingsScreen.js and Backend</p> <p>Initial Issues & Fixes</p> <p>Login Issues (422 Unprocessable Content)</p> <p>Issue: FastAPI rejected login requests due to incorrect request body formatting. Fix: Ensured email and password were correctly passed in the axios.post request in LoginScreen.js. User Data Not Persisting After Login</p> <p>Issue: Logged-in user details were not being saved for profile and settings. Fix: Stored user data in AsyncStorage after a successful login. Navigating to Home Screen After Login (REPLACE Error)</p> <p>Issue: navigation.replace('HomeScreen') failed due to missing screen. Fix: Updated App.js to correctly route users to Main after login.</p> <p>Backend Issues & Fixes</p> <p>Travel Style Not Saving (422 Unprocessable Content)</p> <p>Issue: FastAPI expected user_id in the request body for PUT requests. Fix: Created QuizResultUpdate schema to accept only travel_style. Modified PUT /quiz_results/user/{user_id} to update travel style correctly. Fetching Travel Style for User Settings</p> <p>Issue: Travel style was not being retrieved from PostgreSQL. Fix: Created GET /quiz_results/user/{user_id} to fetch travel style. Updated SettingsScreen.js to call this API and store the result in AsyncStorage.</p>
Russell Han Josef	Feb 22, 2025	1	<p>Drafted video recording flow.</p>
Russell Han Josef	Feb 22, 2025	0.5	<p>Fixed QuizScreen and LoginScreen to ensure first time user is able to take the quiz and save it correctly.</p>

Russell Han Josef	Feb 23, 2025	3	<p>Google Maps Integration on iOS</p> <p>Problem: Needed to integrate Google Maps SDK for iOS.</p> <p>Solution: Installed and configured Google Maps SDK (v7.0.0).</p> <p>Problem: API key was stored in Info.plist, causing security issues.</p> <p>Solution: Moved API key to .env and dynamically loaded it in AppDelegate.swift.</p> <p>Problem: "Tried to register two views with the same name AIRMap" error.</p> <p>Solution: Ensured only one instance of react-native-maps to fix duplicate registration.</p> <p>Problem: API key needed to be dynamically passed to Swift.</p> <p>Solution: Updated Podfile to load .env variables and inject API key into the build.</p> <p>Problem: Map was not displaying if the API key was missing.</p> <p>Solution: Added error handling to InteractiveMapScreen.js to show a message when the API key is missing.</p> <p>Problem: Needed to verify Google Maps displayed properly.</p> <p>Solution: Successfully displayed Google Maps centered on Vancouver, BC in the iOS simulator.</p> <p>Next Steps</p> <p>Implement user location tracking.</p> <p>Add custom markers for points of interest.</p> <p>Optimize map rendering and interactions.</p> <p>Test on a real iOS device for stability.</p>
Russell Han Josef	Feb 23, 2025	2	<p>Added markers with Google Places API.</p>
Russell Han Josef	Feb 24, 2025	1	<p>Check on Firebase app Distribution.</p> <p>Upon checking, need to enrol in Apple Developer Program.</p> <p>Completed: Register for enrolment under educational institution. Request to waive the fee.</p> <p>What's next: Waiting for reply.</p>
Russell Han Josef	Mar 04, 2025	3	<p>Interactive Recommendations Screen</p> <p>Ensured the filter buttons update dynamically based on the selected travel style.</p> <p>Adjusted the zoom in, zoom out, and fullscreen buttons, positioning them correctly in the bottom-right corner of the map (35% of the screen).</p> <p>Refactored the category filter logic to dynamically pull unique categories based on available places.</p> <p>Places Router Updates</p> <p>Updated the API request to fetch places dynamically based on the selected travel style.</p> <p>Extracted unique categories from the API response to dynamically populate the category filter.</p> <p>Ensured error handling for cases where a travel style is missing or places cannot be retrieved.</p> <p>Next Steps</p> <p>Fix the extra white space issue in the scrollable category filter</p> <p>➡ Investigate margin/padding issues in filterScrollContainer and listContainer</p> <p>Implement API caching for places data</p> <p>➡ Cache the API response to prevent unnecessary requests when the same location and travel style are selected again</p>

Russell Han Josef	Mar 04, 2025		<p>Updated user_routes.py</p> <p>Removed quiz_results dependency.</p> <p>Now stores travel_style_id directly in the users table.</p> <p>Defaults travel_style_id = 4 (Undefined) on signup.</p> <p>Allows updating travel_style_id after quiz completion.</p> <p>Created Stack Navigator (AppNavigator.js)</p> <p>Wrapped BottomNavigation inside a Stack.Navigator.</p> <p>Added QuizScreen as a separate screen for navigation.</p> <p>Ensures navigation.navigate('QuizScreen') works from HomeScreen.</p> <p>Updated App.js</p> <p>Now uses AppNavigator instead of BottomNavigation.</p> <p>Ensures proper screen navigation for non-tab screens.</p>
		2	<p>Updated HomeScreen.js</p> <p>Fetches travel_style_id from AsyncStorage & Firebase.</p> <p>Shows "Take Quiz" banner if travel_style_id = 4.</p> <p>Logs user activity in Firebase when viewing the home screen.</p> <p>Updated LoginScreen.js & SignupScreen.js</p> <p>Integrated Firebase Realtime Database.</p> <p>Stores user details in Firebase on signup.</p> <p>Retrieves and syncs user data on login.</p> <p>✂ Next Steps:</p> <p>Update QuizScreen.js</p> <p>Ensure quiz tracking and progress-saving work correctly.</p> <p>Properly update travel_style_id in Firebase & Backend.</p>
Russell Han Josef	Mar 04, 2025	2	<p>Update QuizScreen</p> <p>Added Firebase to track Quiz Progression</p> <p>Fix device node modules issue. Unable to launch iOS device simulator.</p>
		2	<p>Checked the existing Bundler, Ruby, and CocoaPods versions.</p> <p>Verified the project's Gemfile.lock for required dependencies.</p> <p>Removed old Bundler 1.17.2 references from .bundle and vendor/bundle.</p> <p>Updated Bundler to the latest version using gem update bundler.</p> <p>Ran rbenv rehash to ensure the system recognized the new Bundler version.</p> <p>Installed missing gems bigdecimal and logger, which were removed from Ruby 3.4.2 defaults.</p> <p>Removed and reinstalled CocoaPods using gem uninstall cocoapods and gem install cocoapods -v 1.15.2.</p> <p>Updated the project dependencies using bundle install and bundle update --bundler.</p> <p>Verified the installed CocoaPods version with pod --version.</p> <p>Ran bundle exec pod install in the ios directory to set up iOS dependencies.</p> <p>Started Metro Bundler without issues using npx react-native start --reset-cache.</p> <p>Successfully ran the app using npx react-native run-ios.</p>
Russell Han Josef	Mar 06, 2025		<p>Backend Changes (FastAPI & PostgreSQL)</p> <p>Migrated user_id to UUID Format</p> <p>Replaced Integer user IDs with UUID for security, scalability, and consistency across PostgreSQL, Firebase, and AsyncStorage.</p> <p>Added Email Availability Check Before Signup</p>
		3.5	<p>Implemented /users/check_email/ API to prevent duplicate signups.</p> <p>Integrated email validation in SignupScreen.js.</p> <p>Created itinerary_models.py for Collaborative Planning</p> <p>Consolidated Itinerary, ItineraryDay, ItineraryMember, and Activity models.</p> <p>Ensured UUID support across all itinerary models.</p> <p>Added API Endpoint to Fetch Travel Style Details</p>
Russell Han Josef	Mar 06, 2025		<p>Created /travel_styles/{travel_style_id} API to return travel style name & description.</p> <p>Used in ProfileScreen.js to enhance UX.</p>

◊ Frontend Changes (React Native)
Ensured user_id is Always Stored as UUID String

Updated SignupScreen.js, LoginScreen.js, and HomeScreen.js to store user_id as a UUID string in AsyncStorage.
Implemented Travel Style Update Across All Data Sources

SettingsScreen.js now updates travel_style_id in AsyncStorage, PostgreSQL, and Firebase for real-time sync.
Fixed Dropdown Issue in SettingsScreen.js

Replaced onChangeValue with onSelectItem to prevent alerts from triggering when expanding the dropdown.
Updated HomeScreen.js to Use Firebase Instead of PostgreSQL

Fetches travel_style_id from Firebase + AsyncStorage instead of making API calls, improving performance.
Enhanced ProfileScreen.js to Display Travel Style Name & Description

Uses travel_style_id from AsyncStorage to fetch travel style details from PostgreSQL.
Shows meaningful travel style descriptions instead of just an ID.

Fix Homescreen and added some details.
More descriptions to be added....

Itinerary
Updated Router

Added Itinerary List Screen.
Itinerary:

Added Itinerary Details Screen.
Added Swipeable feature to delete.

Itinerary:
Added new feature to re order the day.

Using npm install react-native-draggable-flatlist
Due to this feature, I have to update the model, schema, and router.
Add "Add Day" button at the bottom of Swipable Flat List

Itinerary:
Added ItineraryDayScreen to show all the activities within a day.
Added Function for time sorting.
Using Model to add itinerary.

Implement useFocusEffect after navigating back from ItineraryDay to ItineraryList to refresh the activities

Simone shared about the Navigation progress (done) and Chatbot.
Russell shared about 50% progression of the Itinerary.

Done

		<p>Backend:</p> <p>Fixed POST /itineraries/ by ensuring created_by matches an existing user ID.</p> <p>Implemented GET /itineraries/{itinerary_id} to return full itinerary details.</p> <p>Added POST /itineraries/{itinerary_id}/days/ to allow adding days to an itinerary.</p> <p>Added POST /itineraries/{itinerary_id}/days/{day_id}/activities/ to support adding activities to a day.</p> <p>Implemented DELETE /itineraries/{itinerary_id} to allow itinerary deletion.</p> <p>Implemented DELETE /itineraries/{itinerary_id}/days/{day_id} to allow itinerary day deletion.</p> <p>Implemented PUT /itineraries/{itinerary_id} to allow editing itinerary details.</p> <p>Frontend:</p> <p>Implemented Itinerary List Screen to fetch itineraries from PostgreSQL and display them.</p> <p>Improved Itinerary List Item UI for better readability.</p>
	3	<p>Implemented Itinerary Form Screen to allow creating and editing itineraries.</p> <p>Integrated react-native-calendars for selecting start and end dates in the itinerary form.</p> <p>Ensured that upon itinerary creation, the user is navigated to ItineraryDetailScreen with the new itinerary ID.</p> <p>Implemented Itinerary Detail Screen to display itinerary details and handle deletion.</p> <p>Added Itinerary Day Screen to list all activities for a given itinerary day.</p> <p>Implemented react-native-date-picker for activity time selection.</p> <p>Converted activity list in ItineraryDayScreen to be swipeable, with a placeholder delete action.</p> <p>Fixed swipeable delete button height issue to ensure alignment with activity cards.</p> <p>Improved UI consistency across itinerary-related screens.</p>
Russell Han Josef	Mar 13, 2025	<p>Next steps:</p> <p>Implement actual deletion for activities in ItineraryDayScreen.</p> <p>Allow users to edit itinerary days and activities instead of just deleting them.</p> <p>Integrate Firebase Realtime Database for tracking changes in itineraries.</p> <p>Added an Invite Collaborators button in the Itinerary Detail Screen</p> <p>Placed the button in the bottom fixed button container</p> <p>Designed the InviteCollaboratorsScreen with inline styling</p> <p>Created a search bar for filtering users</p>
	2	<p>Displayed a list of users with an Invite button</p> <p>Added a pending invites section to track invited users</p> <p>Used dummy data for now, preparing for Firebase integration</p> <p>Fixed Text strings must be inside a Text component warning</p> <p>Wrapped the trash icon inside a Text component</p>
Russell Han Josef	Mar 13, 2025	<p>Checked and ensured all text elements are properly inside Text components</p> <p>Updated InviteCollaboratorsScreen.js to store invitations under /invitations/invitee/{userId}.</p> <p>Tracked pending invites under /live_itineraries/{itineraryId}/pendingInvites/.</p>
	1.5	<p>Ensured invited users are displayed in InviteCollaboratorsScreen.js.</p> <p>Updated ItineraryListScreen.js to show pending invitations before shared itineraries.</p> <p>Fetches invitations from /invitations/invitee/{userId}.</p>
Russell Han Josef	Mar 14, 2025	<p>Added "Accept" and "Decline" buttons with placeholder alerts.</p> <p>Fixed fetchItineraryDetails to update days properly</p> <p>Ensured the invite button only shows if the user is the itinerary owner</p>
	3	<p>Updated fetchSharedItineraries to fetch owner details from FastAPI</p> <p>Modified renderItineraryItem to display owner name and email</p> <p>Prevented users from inviting themselves in InviteCollaboratorsScreen</p>
Russell Han Josef	Mar 15, 2025	<p>Ensured adding a new day updates the UI correctly</p> <p>Improved UI/UX for Day:</p> <p>Implemented PUT endpoint for editing itinerary days</p> <p>Added left swipe action to reveal the Edit button on day cards</p> <p>Updated the modal to pre-fill day details for editing</p> <p>Improved date display format to "Sat, Mar 15" using locale options</p>
	1.5	<p>Handled timezone issues by parsing date strings into local Date objects</p>
		<p>Fixed button rendering logic to show "Remove" for collaborators and "Delete" for the owner</p> <p>Ensured isCollaborator updates correctly before rendering buttons</p> <p>Updated fetchItineraryDetails to properly check if the user is a collaborator</p> <p>Confirmed navigation back to ItineraryListScreen after removal</p>
Russell Han Josef	Mar 15, 2025	<p>Verified Firebase updates when a user removes themselves from an itinerary</p>

		<p>ItineraryDayScreen: Fix the delete activity height to match the card.</p> <p>ItineraryListScreen: 1 Show Last Updated for better UX.</p>
Russell Han Josef	Mar 15, 2025	<p>Itinerary Models, Routes, Schema: Add last_updated_by column to the itineraries table Fix Activity Update API - Convert UUIDs to Strings in Response Encountered an issue where updating an activity was failing Checked the frontend code for issues in the request payload Verified that the correct X-User-Id header was being sent</p> <p>2 Ensured the itineraryId, dayId, and activityId were correctly passed Reviewed the FastAPI update activity route for possible issues Debugged the request and response to find inconsistencies Identified that UUID fields in the response were causing validation errors</p>
Russell Han Josef	Mar 15, 2025	<p>Prepared to modify the FastAPI response to return UUIDs as strings</p> <p>InviteCollaboratorsScreen: Added function for remove collaborators as the owner.</p>
		<p>3 Added a dedicated edit icon on the day card to decouple editing from the swipe gesture Integrated a collaborators list in the Overview tab using a Firebase listener Improved the Overview UI with a styled card and clear section for collaborators Ensured add day, edit day, delete day, and drag-and-drop reorder functionalities remain intact Refactored and organized code for clarity and maintainability</p>
Russell Han Josef	Mar 16, 2025	