Date	Date	Number of Hours	Description of Work Done
1/17/25 6:00 PM	Jan 17, 2025	1	Meeting. Distribution of work. Choosing team lead. Decide on the app. -> GitHub Repo -> Misc -> Applied Research_ Logo and Name Research.pdf
1/18/25 4:00 PM	Jan 18, 2025	1	Planning on the project scope and role distribution.
1/20/25 8:30 PM	Jan 20, 2025	1.5	Research on screens to have. Finding screens inspiration. App name and branding.
1/22/25 10:00 AM	Jan 22, 2025	2	Research on Heroku Dynos and Postgres workflow
1/22/25 4:30 PM	Jan 22, 2025	1	Researched about conducting surveys and user testing.
1/22/25 5:30 PM	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
1/25/25 8:00 PM	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.
1/27/25 10:00 PM	Jan 27, 2025	0.5	Regroup to discuss about the progress and knowledge sharing: Wireframe, APIs.
1/28/25 3:00 PM	Jan 28, 2025	0.5	Get Free credits from Heroku account using GitHub for Student Developer Pack
1/28/25 3:30 PM	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
1/28/25 7:30 PM	Jan 28, 2025	1	Created Video to share surveys on Instagram. Link: https://www.instagram.com/reel/DFXOyODRLOl/?igsh=ZnF4OWhqbTk0NnVx
1/29/25 9:30 AM	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.
2/3/25 9:30 PM	Feb 03, 2025	1.5	Created backend shell. Tested the shell and it's running. Pushed to GitHub repo.
2/3/25 11:00 PM	Feb 03, 2025	1	Regroup to discuss about Database Schema of PostgreSQL and NoSQL. Update on the frontend shell. Update on the backend shell. Troubleshoot how to run ios on the machine on the first time pulling. Ensure frontend shell can work on both machines. Tasks assignment for the upcoming week.
2/4/25 12:00 AM	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 Solved!
2/4/25 9:30 AM	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 Implentation 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.

2/4/25 1:30 PM	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.
2/4/25 2:30 PM	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.
2/4/25 3:30 PM	Feb 04, 2025	2	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. 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. Explanation: This reinforced the differences between local and production environments, the need for dynamic configurations, and how Heroku manages deployments and environment variables.
2/9/25 7:30 PM	Feb 09, 2025	1	Created travel style quiz. Total 7 questions. Scoring system using point-based system.

2/10/25 8:00 PM	Feb 10, 2025	2	Regroup / Knowledge Sharing session. Russell: - Explained how the backend works for PSQL part. - Explained how to use pqAdmin tool as database management tool. - Explained the workflow with backend: when writing new code for backend, test it on local machine before pushing to Github. - Showed how deployment works with Github - Heroku setup. General: - Discussed the next workflow in tackling MVPs. - Discussed to re-order MVP priority based on survey. Task Assignments:
2/11/25 1:30 PM	Feb 11, 2025	3.5	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. 2. Circular Import Issue Problem: Importing Base from db.py led to circular dependencies. Solution: Moved Base to base.py and updated model imports. 3. Missing email-validator Error Problem: FastAPI required email-validator, despite being in requirements.txt. Solution: Reinstalled dependencies manually on Heroku. 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. 5. Heroku App Not Restarting Properly Problem: Deployment changes weren't reflecting. Solution: Restarted the app and purged Heroku build cache. 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. 7. Database Connection Test Failed on Heroku Problem: Remote database connection wasn't verifying. Solution: Created /test-db endpoint and confirmed it works.

2/11/25 5:00 PM	Feb 11, 2025	0.5	Running db.py to Create Tables on Heroku: Problems & Solutions 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. 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. 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 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'; Final Outcome: Successfully ran db.py on Heroku, ensuring tables were created in the correct schema.
2/12/25 10:00 AM	Feb 12, 2025	1.5	Summary of Fixes & Progress 1. CRUD Implementation for Users Created POST /users → Create User (with password hashing). Created GET /users/{user_id} → Retrieve User by ID. Created PUT /users/{user_id} → Update User (name, email, password). Created DELETE /users/{user_id} → Delete User. 2. Fixed Errors InvalidRequestError → Added ForeignKey("users.id") in quiz_model.py. TypeError: 'password' is an invalid keyword argument for User → Ensured password_hash is used in user_model.py. NameError: name 'user_schema' is not defined → Fixed incorrect import in user_routes.py. zsh: no matches found: passlib[bcrypt] → Installed using pip install "passlib[bcrypt]". 3. Fixed Duplicate URL Path Issue Issue: "/users/users/{user_id}" in FastAPI /docs. Fix: Removed redundant /users prefix from routes in user_routes.py. 4. Tested Locally Verified all CRUD operations using FastAPI /docs. Confirmed correct URL paths after fixing duplication. Next Steps Test CRUD operations for Itineraries, Places, Badges, and Quiz Results. Once confirmed, deploy to Heroku and re-test on live API.

2/12/25 11:30 AM	Feb 12, 2025	1	Log Summary for Places CRUD Implementation
			1. Implemented CRUD for Places
			Created POST /places → Add a new place.
			Created GET /places/{id} → Retrieve a place by ID.
			Created PUT /places/{id} → Update place details.
			Created DELETE /places/{id} → Remove a place.
			2. Fixed Issues
			Fixed timezone inconsistency → Ensured last_updated is stored in UTC.
			Resolved datetime.utcnow() deprecation warning → Used datetime.now(timezone.utc).replace(tzinfo=None).
			Verified timestamps consistency → Matched last_updated with created_at format.
			3. Successfully Tested
			POST /places → Verified place creation with manual data.
			GET /places/{id} → Retrieved created places correctly.
			PUT /places/{id} → Updated place details without timezone mismatch.
			DELETE /places/{id} → Successfully removed places from the database.
			Next Steps
			Implement User Favorites (user_favorite_routes.py).
			Ensure Users ↔ Places relationship works correctly.
			Prepare for Google Places API integration.
2/12/25 3:30 PM	Feb 12, 2025	0.5	Log Summary for User Favorites Implementation
			1. Implemented CRUD for User Favorites
			Created POST /user_favorites → Add a place to favorites.
			Created GET /user_favorites/{user_id} → Retrieve a user's favorite places.
			Created DELETE /user_favorites/{favorite_id} → Remove a favorite place.
			2. Fixed Issues
			Validated user and place existence before adding a favorite.
			Prevented duplicate favorites by checking existing records.
			Ensured added_at timestamp is stored in UTC for consistency.
			3. Successfully Tested
			POST /user_favorites → Added places to favorites successfully.
			GET /user_favorites/{user_id} → Retrieved correct favorites for users.
			DELETE /user_favorites/{favorite_id} → Removed favorites as expected.
			4. Updated main.py
			Included user_favorite_routes in FastAPI router.
			Next Steps
			Implement Badges (badge_routes.py).
			Ensure User ↔ Badges relationship works correctly.

2/12/25 4:00 PM	Feb 12, 2025	1.5	Set Up Google Places API Integration

Chose Google Places API (Old Version) for simpler API key authentication.
Tested API manually using Postman & cURL.
Implemented FastAPI Route for Places Search

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

Implemented latitude/longitude boundary check to block requests outside BC.
Verified restriction by testing New York (Successfully blocked).
Error Handling & Optimizations

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

Verified working results for Vancouver, BC.
Ensured API key security using environment variables (.env, Heroku Config Vars). |
| 2/13/25 10:00 AM | Feb 13, 2025 | 1.5 | Tasks Completed:
- 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.

Next Steps:
- Implement Edit Profile feature.
- Add form validation for Login & Signup.
- Prepare backend integration for authentication and profile updates.
- Enhance UI with better styling. |

| 2/13/25 11:30 AM | Feb 13, 2025 | 0.75 | Tasks Completed:
- Backend Integration for Authentication:
  - 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:
  - 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:
  - 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:
  - 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.

Next Steps:
- 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. |
| 2/13/25 12:45 PM | Feb 13, 2025 | 2.25 | Attempted to rename React Native app from "frontend" to "WayPoint" → Encountered issues, reverted to "frontend"
Updated package.json and app.json to reflect the correct app name → Reverted due to build errors
Checked and updated Xcode Signing & Capabilities → Used free Apple ID for provisioning
Attempted to set correct Bundle Identifier for Firebase setup → Reverted due to build failures
Installed Firebase dependencies (@react-native-firebase/app) → Successfully installed
Fixed CocoaPods issues with modular headers → Modified Podfile and ran pod install --repo-update
Configured Firebase in AppDelegate.swift → Updated to FirebaseApp.configure()
Encountered xcodebuild error code 65 while running iOS build → Attempted multiple fixes
Deleted and reinstalled CocoaPods, node_modules, and Xcode DerivedData → No success
Manually deleted ios/build/ and cleaned Xcode project → Issue persisted
Ran xcodebuild clean and pod install --repo-update → Did not resolve the issue
Tried running the app via Metro Bundler (npx react-native run-ios) → Still failed
Decided to fully reset the project by deleting and reinstalling all dependencies → Still encountering build issues |

2/14/25 2:00 PM	Feb 14, 2025	4	Summary Log: Firebase Realtime Database Setup & Next Steps
			Problems:
			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
			Solutions Attempted:
			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
			What's Next:
			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
			Next attempt: Clean setup from the beginning

2/14/25 7:00 PM	Feb 14, 2025	1.5	Work Log: Firebase Integration in React Native (iOS)
			Problems & Solutions
			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.
			Next Steps
			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.

2/14/25 8:30 PM	Feb 14, 2025	0.5	Work Log: Firebase Auto-Detection Fix in React Native (iOS)
			Problems & Solutions **Firebase Not Initializing Automatically** Firebase was not detecting GoogleService-Info.plist. Manually initialized Firebase in AppDelegate.swift. Confirmed Firebase auto-detection now works. **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. **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/. Next Steps Fetch and display data from Firebase in the app. Secure Firebase database rules based on authentication.

2/17/25 10:30 PM	Feb 17, 2025	1.25	Meeting Notes Summary (February 17, 2025)
			1. Upcoming 1-Week Tasks
			Focus on the next two MVPs:
			Personalized Recommendations (Google Places API integration for recommendations)
			Interactive Map (Google Maps API integration for visualization)
			2. Past Week Progress Updates
			Team members shared knowledge and updates on completed tasks.
			3. Heroku Backend Server Documentation
			Discussion on CRUD operations for backend endpoints
			4. Connecting Simulator to Heroku Server
			Setting up the React Native simulator to interact with the backend hosted on Heroku.
			5. Planning for Video Workflow
			Outlining the video workflow for the mid-term report.
			Deciding on tools and steps for video creation.
			6. Firebase Realtime Database & App Distribution
			Revisiting Firebase Realtime Database setup.
			Setting up Firebase App Distribution for testing.
			7. Google Places API on /search Path
			Integrating Google Places API for search functionality.
			Ensuring that the API can return filtered results based on user preferences.

2/22/25 10:30 PM	Feb 22, 2025	2	Fixes & Improvements in SettingsScreen.js and Backend Initial Issues & Fixes Login Issues (422 Unprocessable Content) 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 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) Issue: navigation.replace('HomeScreen') failed due to missing screen. Fix: Updated App.js to correctly route users to Main after login. Backend Issues & Fixes Travel Style Not Saving (422 Unprocessable Content) 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 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.
2/22/25 2:00 PM	Feb 22, 2025	1	Drafted video recording flow.
2/22/25 3:00 PM	Feb 22, 2025	0.5	Fixed QuizScreen and LoginScreen to ensure first time user is able to take the quiz and save it correctly.

2/23/25 1:00 PM	Feb 23, 2025	3	Google Maps Integration on iOS Problem: Needed to integrate Google Maps SDK for iOS. Solution: Installed and configured Google Maps SDK (v7.0.0). Problem: API key was stored in Info.plist, causing security issues. Solution: Moved API key to .env and dynamically loaded it in AppDelegate.swift. Problem: "Tried to register two views with the same name AIRMap" error. Solution: Ensured only one instance of react-native-maps to fix duplicate registration. Problem: API key needed to be dynamically passed to Swift. Solution: Updated Podfile to load .env variables and inject API key into the build. Problem: Map was not displaying if the API key was missing. Solution: Added error handling to InteractiveMapScreen.js to show a message when the API key is missing. Problem: Needed to verify Google Maps displayed properly. Solution: Successfully displayed Google Maps centered on Vancouver, BC in the iOS simulator. Next Steps Implement user location tracking. Add custom markers for points of interest. Optimize map rendering and interactions. Test on a real iOS device for stability.
2/23/25 10:00 PM	Feb 23, 2025	2	Added markers with Google Places API.
2/24/25 10:00 AM	Feb 24, 2025	1	Check on Firebase app Distribution. Upon checking, need to enrol in Apple Developer Program. Completed: Register for enrolment under educational institution. Request to waive the fee. What's next: Waiting for reply.
2/24/25 5:00 PM	Feb 24, 2025	1.75	Video Recording for Mid Term Report Checkpoint
2/24/25 9:30 PM	Feb 24, 2025	2.5	Finishing and finalizing midterm report. Preparing documents for submission. Push log to the github.
2/26/25 12:00 PM	Feb 26, 2025	0.25	Midterm Feedback by Priya

3/4/25 1:00 PM	Mar 04, 2025	3	Interactive Recommendations Screen
			Ensured the filter buttons update dynamically based on the selected travel style.
			Adjusted the zoom in, zoom out, and fullscreen buttons, positioning them correctly in the bottom-right corner of the map (35% of the screen).
			Refactored the category filter logic to dynamically pull unique categories based on available places.
			Places Router Updates
			Updated the API request to fetch places dynamically based on the selected travel style.
			Extracted unique categories from the API response to dynamically populate the category filter.
			Ensured error handling for cases where a travel style is missing or places cannot be retrieved.
			Next Steps
			Fix the extra white space issue in the scrollable category filter
			➡️Investigate margin/padding issues in filterScrollContainer and listContainer
			Implement API caching for places data
			➡️Cache the API response to prevent unnecessary requests when the same location and travel style are selected again
3/4/25 4:00 PM	Mar 04, 2025	2	Updated user_routes.py
			Removed quiz_results dependency.
			Now stores travel_style_id directly in the users table.
			Defaults travel_style_id = 4 (Undefined) on signup.
			Allows updating travel_style_id after quiz completion.
			Created Stack Navigator (AppNavigator.js)
			Wrapped BottomNavigation inside a Stack.Navigator.
			Added QuizScreen as a separate screen for navigation.
			Ensures navigation.navigate('QuizScreen') works from HomeScreen.
			Updated App.js
			Now uses AppNavigator instead of BottomNavigation.
			Ensures proper screen navigation for non-tab screens.
			Updated HomeScreen.js
			Fetches travel_style_id from AsyncStorage & Firebase.
			Shows "Take Quiz" banner if travel_style_id = 4.
			Logs user activity in Firebase when viewing the home screen.
			Updated LoginScreen.js & SignupScreen.js
			Integrated Firebase Realtime Database.
			Stores user details in Firebase on signup.
			Retrieves and syncs user data on login.
			🛠️ Next Steps:
			Update QuizScreen.js
			Ensure quiz tracking and progress-saving work correctly.
			Properly update travel_style_id in Firebase & Backend.

3/4/25 10:30 PM	Mar 04, 2025	2	Update QuizScreen Added Firebase to track Quiz Progression
3/5/25 10:00 AM	Mar 05, 2025	1	Updated Buttons for Zoom and Fullscreen Updated icon for list to load from Places API fixed the list spacing (scrollFilterContainer)
3/6/25 10:30 AM	Mar 06, 2025	2	Fix device node modules issue. Unable to launch iOS device simulator. Checked the existing Bundler, Ruby, and CocoaPods versions. Verified the project's Gemfile.lock for required dependencies. Removed old Bundler 1.17.2 references from .bundle and vendor/bundle. Updated Bundler to the latest version using gem update bundler. Ran rbenv rehash to ensure the system recognized the new Bundler version. Installed missing gems bigdecimal and logger, which were removed from Ruby 3.4.2 defaults. Removed and reinstalled CocoaPods using gem uninstall cocoapods and gem install cocoapods -v 1.15.2. Updated the project dependencies using bundle install and bundle update --bundler. Verified the installed CocoaPods version with pod --version. Ran bundle exec pod install in the ios directory to set up iOS dependencies. Started Metro Bundler without issues using npx react-native start --reset-cache. Successfully ran the app using npx react-native run-ios.
3/6/25 12:30 PM	Mar 06, 2025	3.5	🔷 Backend Changes (FastAPI & PostgreSQL) Migrated user_id to UUID Format Replaced Integer user IDs with UUID for security, scalability, and consistency across PostgreSQL, Firebase, and AsyncStorage. Added Email Availability Check Before Signup Implemented /users/check_email/ API to prevent duplicate signups. Integrated email validation in SignupScreen.js. Created itinerary_models.py for Collaborative Planning Consolidated Itinerary, ItineraryDay, ItineraryMember, and Activity models. Ensured UUID support across all itinerary models. Added API Endpoint to Fetch Travel Style Details Created /travel_styles/{travel_style_id} API to return travel style name & description. Used in ProfileScreen.js to enhance UX.

3/6/25 12:30 PM	Mar 06, 2025		🔷 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.
3/6/25 4:45 PM	Mar 06, 2025	1.5	Fix Homescreen and added some details. More descriptions to be added….
3/7/25 10:00 AM	Mar 07, 2025	1.5	Itinerary Updated Router Added Itinerary List Screen.
3/7/25 11:30 PM	Mar 07, 2025	2	Itinerary: Added Itinerary Details Screen. Added Swipeable feature to delete.
3/7/25 2:30 PM	Mar 07, 2025	1.5	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
3/7/25 4:00 PM	Mar 07, 2025	1.5	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
3/10/25 4:00 PM	Mar 10, 2025	1	
3/10/25 4:00 PM	Mar 10, 2025	0.6	Simone shared about the Navigation progress (done) and Chatbot. Russell shared about 50% progression of the Itinerary.
3/10/25 10:00 PM	Mar 10, 2025	1	Done

| 3/13/25 2:30 PM | Mar 13, 2025 | 3 | Backend:
Fixed POST /itineraries/ by ensuring created_by matches an existing user ID.
Implemented GET /itineraries/{itinerary_id} to return full itinerary details.
Added POST /itineraries/{itinerary_id}/days/ to allow adding days to an itinerary.
Added POST /itineraries/{itinerary_id}/days/{day_id}/activities/ to support adding activities to a day.
Implemented DELETE /itineraries/{itinerary_id} to allow itinerary deletion.
Implemented DELETE /itineraries/{itinerary_id}/days/{day_id} to allow itinerary day deletion.
Implemented PUT /itineraries/{itinerary_id} to allow editing itinerary details.

Frontend:
Implemented Itinerary List Screen to fetch itineraries from PostgreSQL and display them.
Improved Itinerary List Item UI for better readability.
Implemented Itinerary Form Screen to allow creating and editing itineraries.
Integrated react-native-calendars for selecting start and end dates in the itinerary form.
Ensured that upon itinerary creation, the user is navigated to ItineraryDetailScreen with the new itinerary ID.
Implemented Itinerary Detail Screen to display itinerary details and handle deletion.
Added Itinerary Day Screen to list all activities for a given itinerary day.
Implemented react-native-date-picker for activity time selection.
Converted activity list in ItineraryDayScreen to be swipeable, with a placeholder delete action.
Fixed swipeable delete button height issue to ensure alignment with activity cards.
Improved UI consistency across itinerary-related screens.

Next steps:
Implement actual deletion for activities in ItineraryDayScreen.
Allow users to edit itinerary days and activities instead of just deleting them.
Integrate Firebase Realtime Database for tracking changes in itineraries. |
| 3/13/25 10:00 PM | Mar 13, 2025 | 2 | Added an Invite Collaborators button in the Itinerary Detail Screen
Placed the button in the bottom fixed button container
Designed the InviteCollaboratorsScreen with inline styling
Created a search bar for filtering users
Displayed a list of users with an Invite button
Added a pending invites section to track invited users
Used dummy data for now, preparing for Firebase integration
Fixed Text strings must be inside a Text component warning
Wrapped the trash icon inside a Text component
Checked and ensured all text elements are properly inside Text components |

3/14/25 10:30 PM	Mar 14, 2025	1.5	Updated InviteCollaboratorsScreen.js to store invitations under /invitations/invitee/{userId}. Tracked pending invites under /live_itineraries/{itineraryId}/pendingInvites/. Ensured invited users are displayed in InviteCollaboratorsScreen.js. Updated ItineraryListScreen.js to show pending invitations before shared itineraries. Fetched invitations from /invitations/invitee/{userId}. Added "Accept" and "Decline" buttons with placeholder alerts.
3/15/25 10:00 AM	Mar 15, 2025	3	Fixed fetchItineraryDetails to update days properly Ensured the invite button only shows if the user is the itinerary owner Updated fetchSharedItineraries to fetch owner details from FastAPI Modified renderItineraryItem to display owner name and email Prevented users from inviting themselves in InviteCollaboratorsScreen Ensured adding a new day updates the UI correctly
3/15/25 5:00 PM	Mar 15, 2025	1.5	Improved UI/UX for Day: Implemented PUT endpoint for editing itinerary days Added left swipe action to reveal the Edit button on day cards Updated the modal to pre-fill day details for editing Improved date display format to "Sat, Mar 15" using locale options Handled timezone issues by parsing date strings into local Date objects Fixed button rendering logic to show "Remove" for collaborators and "Delete" for the owner Ensured isCollaborator updates correctly before rendering buttons Updated fetchItineraryDetails to properly check if the user is a collaborator Confirmed navigation back to ItineraryListScreen after removal Verified Firebase updates when a user removes themselves from an itinerary
3/15/25 5:30 PM	Mar 15, 2025	1	ItineraryDayScreen: Fix the delete activity height to match the card. ItineraryListScreen: Show Last Updated for better UX. Itinerary Models, Routes, Schema: Add last_updated_by column to the itineraries table

3/15/25 9:00 PM	Mar 15, 2025	2	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 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 Prepared to modify the FastAPI response to return UUIDs as strings
3/16/25 2:50 PM	Mar 16, 2025	3	InviteCollaboratorsScreen: Added function for remove collaborators as the owner. 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
3/16/25 2:50 PM	Mar 16, 2025	3	Firebase initialization is performed both natively and in JavaScript Source of API configuration is embedded in native files (google-services.json, GoogleService-Info.plist) Native initialization in AppDelegate.swift remains necessary for iOS support Implementation of Firebase Storage for images has been challenging and remains unresolved
3/16/25 10:30 PM	Mar 16, 2025	2	Re-try Firebase again - Part 1 Firebase Storage module initialization not clear within the React Native context Difficulty integrating native Firebase configurations with JavaScript code Uncertainty in setting up correct file paths and handling file uploads Potential permission issues when accessing device storage for images

3/17/25 12:00 PM	Mar 17, 2025	5	Re-try Firebase again - Part 2 Problem: Build errors due to warnings treated as errors and non-modular includes in Firebase modules Adjusted the Podfile's post_install hook to remove -Werror and disable pedantic warnings Allowed non-modular includes for RNFB targets by setting CLANG_ALLOW_NON_MODULAR_INCLUDES_IN_FRAMEWORK_MODULES Removed a stray reference ("x") causing a syntax error in the Podfile Cleaned Derived Data, deintegrated, updated and reinstalled CocoaPods Converted Objective-C import syntax to Swift import (import Firebase) for Swift usage Conclusion: 10 Hours spent for Firebase Storage service and still not working. Going for another alternative: AWS S3
3/17/25 9:00 PM	Mar 17, 2025	4	Attempting AWS S3 for image storage. Successful implementation for AWS S3. Image upload feature with react-native-image-picker is not working due to iOS limitation. Alternative: react-native-image-crop-picker Updated the S3 upload endpoint to generate a presigned URL and update the itinerary record's extra_data with the new image URL Modified the extra_data field in the Itinerary model to use MutableDict so in-place JSON updates are detected Adjusted the React Native ItineraryDetailScreen to fetch the image URL from extra_data and display it using an ImageBackground Troubleshot duplicate image rendering by removing an extra Image component Provided an alternative endpoint approach to stream the image from S3 if needed
3/19/25 9:30 AM	Mar 19, 2025	2.5	Fixed alignment issues for the "Add" button in PlacesModal.js Added spacing between text input and "Add" button in PlacesModal.js Made notes section scrollable while keeping the title fixed Allowed notes panel to expand dynamically but stop at a max height Integrated Firebase Realtime Database for storing itinerary notes Replaced AsyncStorage with Firebase for notes persistence Synced notes preview in ItineraryDetailScreen.js with Firebase updates Adjusted modal layout in NotesModal.js to provide more writing space Ensured "Tap to add notes" appears when notes are empty Fixed "Places to Visit" container height to be flexible Ensured "Places to Visit" list appears below the title Allowed places panel to grow dynamically instead of having a fixed height Adjusted Firebase integration for NotesModal.js by correctly passing itineraryId Fixed inconsistencies between notesPanel and placesPanel heights Ensured preview text in notes appears below the title Wrapped notes preview in a ScrollView to enable proper scrolling Adjusted modal size in NotesModal.js to allow more space for text input Ensured notes persist and update in real-time across devices Implemented better text wrapping and positioning for notes preview Updated useEffect hooks in both NotesModal.js and ItineraryDetailScreen.js for better state handling

3/19/25 12:00 PM	Mar 19, 2025	0.1	Fixed "Remove" button not showing for collaborators Properly set isCollaborator when fetching collaborators Ensured collaborators list updates correctly in Firebase
3/19/25 2:00 PM	Mar 19, 2025	0.5	Replaced AsyncStorage with Firebase for storing the places list Ensured places sync in real-time across devices Fixed itineraryId not being passed to PlacesModal Updated ItineraryDetailScreen to fetch places from Firebase Ensured PlacesModal saves and retrieves places from Firebase Removed all AsyncStorage references for places list
3/19/25 2:30 PM	Mar 19, 2025	1.25	Identified estimated_cost missing in API response Fixed ActivitySchema to include estimated_cost Ensured estimated_cost is explicitly cast as float in get_itinerary Verified API response using Swagger UI Updated fetchItineraryDetails to sum total estimated cost of all days Preserved sortActivitiesByTime while calculating total cost Updated UI to display the dynamic total estimated cost in ItineraryDetailScreen
3/19/25 3:45 PM	Mar 19, 2025	0.75	Added a horizontal scroll view for the collaborators list Updated the collaborators section to allow left to right scrolling Ensured the scroll view does not show horizontal scroll indicators Nodified styles to align items in a row and add spacing Tested the UI to verify smooth horizontal scrolling
3/19/25 4:30 PM	Mar 19, 2025	0.5	added a modal for entering and managing other costs allowed selection of cost type and subtype from a predefined list enabled users to input item name and amount displayed total other costs in the other costs panel listed saved other costs inside the modal added functionality to remove saved costs
3/19/25 8:30 PM	Mar 19, 2025	0.5	Updated CFBundleDisplayName in Info.plist Renamed project in Xcode under Identity and Type Updated Bundle Identifier in Xcode (if needed) Cleared cache and rebuilt the project Replaced app icons in Images.xcassets/AppIcon.appiconset/ Ensured new icons appear in Xcode Cleaned and rebuilt the project
3/19/25 9:00 PM	Mar 19, 2025	0.25	Identified duplicate weather API calls in HomeScreen.js Found API calls triggered in both handleLocationGranted and useEffect Removed one API call to prevent redundant requests Ensured weather data is fetched only once when location updates

3/19/25 9:15 PM	Mar 19, 2025	0.25	Added a centered "No Itineraries" message when the list is empty Updated Personal Itineraries and Shared Itineraries to display the empty state Replaced the Add Itinerary button with a FontAwesome "+" icon Styled the floating "+" button to be minimalist and positioned at the bottom-right
3/19/25 9:30 PM	Mar 19, 2025	1.75	Feature Update: UI Enhancements & Date Picker Fixes Implemented Google Places Autocomplete with a modal for destination selection Ensured modal expands fully and remains visible when clicked Added a clear ("✕") button to easily remove the destination input Adjusted destination input size to match other fields (90% input, 10% clear button) Standardized font size (18px) and left padding across all input fields Fixed date picker behavior to ensure the first click highlights the start date in blue Ensured date range selection properly marks start, end, and in-between dates Updated calendar logic to automatically close after selecting an end date Improved UI consistency for input spacing, margin, and alignment
3/20/25 10:15 PM	Mar 20, 2025	0.5	Created reusable FeatureCarousel component Added horizontal FlatList with paging and snapping Used image backgrounds for each feature card Applied resizeMode="cover" to handle varying image sizes Made layout responsive to SafeAreaView using useWindowDimensions() Added dark overlay for text readability Implemented pagination dots under the carousel Made dots tappable to scroll to specific cards
3/20/25 10:45 PM	Mar 20, 2025	0.5	Added a feature carousel with image backgrounds and pagination Implemented a tappable pagination dot system for smooth navigation Created a "Start Your Journey" banner with animated button effects Conditionally displayed the banner based on quiz completion state Always displayed the logo in the top left corner Adjusted SafeAreaView and header styles to reduce extra space Fine-tuned padding and height for better logo positioning
3/20/25 11:15 PM	Mar 20, 2025	0.75	Replaced separate letter buttons with full-width answer buttons Ensured the entire button is tappable for better UX Added bounce animation when selecting an answer Displayed checkmark on the right side of the selected answer Used useRef for animatedScales to maintain hook order Moved useAnimatedStyle outside JSX to prevent render errors Fixed "Rendered Fewer Hooks Than Expected" issue

3/21/25 1:45 PM	Mar 21, 2025	0.25	Wrapped entire content inside ScrollView to allow vertical scrolling Moved the header outside ScrollView to keep it fixed at the top Applied position absolute to header to ensure it stays visible Added marginTop to ScrollView to prevent overlap with the fixed header Used zIndex and elevation to keep header above other content Added shadow for better visual separation
3/21/25 8:30 PM	Mar 21, 2025	0.5	Added isLoadingResult state to control when loading shows Updated determineTravelStyle to show loading before result Used Progress.CircleSnail as the animated loading spinner Added a delay using setTimeout before showing the result screen Created loadingContainer and loadingText styles for clean layout Installed react-native-svg to fix CircleSnail "Unimplemented" error
3/21/25 8:30 PM	Mar 21, 2025	0.25	Added Confetti Animation When Result Appears Added Fade-In Animation for the Result Card Using Reanimated Updated Result Card Layout for Better Visual Hierarchy Increased Emoji Size and Spacing Separated Result Text into Multiple Lines for Clarity Updated Result Card Background Color to Brand Navy #1E3A8A Adjusted Text Colors for Contrast on Dark Background Improved Card Elevation and Shadow for Depth
3/21/25 8:45 PM	Mar 21, 2025	0.5	Replaced animatedScales with a 2D animatedScalesRef using useRef Created animatedStylesRef outside render loop for safe hook usage Removed all useSharedValue and useAnimatedStyle from inside loops and conditionals Ensured only the selected answer animates when tapped Fixed runtime errors caused by missing or misplaced hook declarations Validated that quiz screen runs cleanly without crashing
3/21/25 9:15 PM	Mar 21, 2025	0.25	Replaced bottom navigation emoji icons with FontAwesome icons Updated CustomBottomNavigation to render FontAwesome icons with dynamic color Adjusted navIcon style for better alignment with FontAwesome Replaced More menu emoji items with FontAwesome icons Used sign-in-alt icon for Check In to avoid duplicating map icon Ensured layout and spacing remain consistent in both menus
3/21/25 9:30 PM	Mar 21, 2025	1.25	Created a modal component to display AI-generated packing tips Used OpenAI endpoint to generate suggestions based on weather and city Added tap interaction to the left side of the weather widget Handled modal visibility and loading states properly Fixed missing weather location name with fallback city logic Made sure the modal opens only when data is valid Confirmed that clicking the widget now triggers a smart packing tip

3/23/25 7:00 PM	Mar 23, 2025	3	Add AWS S3 bucket and created users/{userId}/profile.jpg folder structure Create presigned URL FastAPI route for profile photo upload Upload image from React Native using Image Crop Picker and XMLHttpRequest to S3 Save uploaded photo URL to Firebase Realtime Database Add image display logic in ProfileScreen with fallback to AsyncStorage Auto-refresh profile image using Firebase onValue listener Append ?ts=Date.now() to bust image cache after upload Show spinner (ActivityIndicator) while uploading image Display "Tap to change photo" hint below profile picture Handle image picker cancel without showing error Fix AWS S3 bucket permissions to allow public image access Ensure image loads with proper Firebase and S3 structure Prevent multiple uploads by disabling button while uploading
3/23/25 10:00 PM	Mar 23, 2025	1	added floating draggable button to HomeScreen button opens Chatbot screen when pressed error: upon clicking, the button stucks at the top left.
3/23/25 11:00 PM	Mar 23, 2025	1	Replaced PanResponder with Gesture.Pan from Gesture Handler Used useSharedValue and useAnimatedStyle for dynamic positioning Applied clamp to restrict dragging within screen bounds Set initial position to bottom right using screen dimensions Used Gesture.Simultaneous to handle both drag and tap Wrapped navigation inside runOnJS to avoid UI thread crash Ensured floating button is rendered after all other content Applied absolute positioning and high zIndex to stay above modals Removed TouchableOpacity and used View to avoid gesture conflicts
3/24/25 12:00 AM	Mar 24, 2025	0.8	Separated TouchableOpacity from direct string content Wrapped Tap to change photo inside its own <Text> component Moved image and text inside a parent <View> for layout safety Ensured no raw strings are rendered outside <Text> tags Prevented potential crash from unexpected text in TouchableOpacity block
3/24/25 10:45 AM	Mar 24, 2025	0.5	Added input validation for email and password Displayed specific inline error messages instead of alerts Applied red border to inputs with errors Added password visibility toggle using FontAwesome icons Fixed padding/margin to align email and password fields Ensured errors persist visually after failed login attempts Removed double borders on password input field Used consistent styles between email and password fields Cleaned up state management for error updates on input change

3/24/25 12:00 PM	Mar 24, 2025	2.5	Reported unexpected charges for March billing Clarified that the project was for academic purposes and part of a student assignment Provided project details and confirmed usage intentions Google Cloud Support reviewed billing and usage activity Support escalated the case for further review Awaiting final resolution or credit approval from Google Cloud team
3/24/25 3:00 PM	Mar 24, 2025	2.5	Work on Places API (New) and update logic Updated to use latest Google Places API (v3) with includedTypes Saved API results to PostgreSQL with proper caching Added /cached router to fetch nearby places from DB Applied travel style filtering to cached results Refactored FlatList and MapView to share same filtered list Displayed placeholder image for each item Capitalized travel style and category labels for better UI Ensured category formatting handles underscore and multiple words Synced map marker press to scroll FlatList Synced FlatList item press to zoom and open map marker callout Fixed unique key warning in FlatList by adding index in keyExtractor
3/24/25 10:00 PM	Mar 24, 2025	1.5	Walk through the workaround for new API route. We are not pulling API using Places API all the time anymore since we put a limitation. Introduced places/search/cached as the route we are using for pulling data from our own database.
3/25/25 9:00 PM	Mar 25, 2025	0.5	Converted cached_data values to valid JSON strings (double-quoted) Ensured cached_data column is importable as JSONB into PostgreSQL Matched CSV columns with database table structure Filled empty rating values with 0.0 Saved final version of CSV for Heroku import via psql using \copy command
3/25/25 10:30 PM	Mar 25, 2025	1.5	Replaced GooglePlacesAutocomplete with custom modal using Places API v3 Created WeatherSearchModal component with city suggestions and selection Implemented debounce on city input to limit API requests Fetched place details including coordinates using placeId Saved selected city to AsyncStorage for persistence Called weather API using selected city coordinates Rendered modal using React Native Modal with 90 percent height Integrated modal into HomeScreen with proper state control Fixed hook usage and debounce-related bugs Verified and polished modal appearance and functionality

3/26/25 10:00 AM	Mar 26, 2025	0.5	Added custom modal to search destinations using Google Places API Updated modal to return both city and country Formatted selected destination to store as "City, Country" string Updated itinerary form to display and submit destination string properly Fixed render error from object being passed to Text Ensured compatibility with PostgreSQL text field requirement
3/26/25 3:00 PM	Mar 26, 2025	1	Added editable profile fields: username, bio, location, languages, and favorite destinations Implemented edit/save toggle with pencil and save icons in header Synced profile data with Firebase Realtime Database and AsyncStorage Fixed missing pencil icon by confirming header button setup in navigation Resolved dependency issue with objectWithoutProperties by cleaning and reinstalling node modules Verified compatibility of React Navigation and MaskedView packages Profile screen now switches between view and edit mode without error
3/26/25 4:00 PM	Mar 26, 2025	1.5	Added editable fields for username bio location languages favorite destinations dream destination travel app instagram Added editable dropdowns for packing style travel companion budget range planning habit trip role Connected all profile fields to Firebase Realtime Database with real-time updates and fallback from AsyncStorage Enabled S3 profile image upload with presigned URL from backend and automatic Firebase update Cached profile image locally for fallback display Displayed uploaded image with cache-busting timestamp Added loading spinner while uploading profile image Implemented "Edit Profile" mode toggle with "Save Profile" and FontAwesome "X" button Used useFocusEffect to sync data from Firebase and local storage Styled header to match navy blue color all the way to the notch Removed header shadow line using headerStyle with elevation and shadow properties Retained custom header buttons using useLayoutEffect when editing profile
3/26/25 10:00 PM	Mar 26, 2025	2	**Fix Destination Parsing for Itinerary Edit** Updated fetchItineraryDetails to parse destination string from PostgreSQL into city and country object Removed unnecessary setLocation call Confirmed destination modal sets destination object correctly Ensured destination is stored as a string when submitting the form Cleaned up state usage to avoid undefined display bugs **Update Login and Signup Screens with Background Image** Added ImageBackground to Login screen Styled Login screen with background image and overlay card Added SafeAreaView for proper layout on iOS Centered login form vertically when active Prepared styling for a smooth visual transition

3/28/25 9:00 PM	Mar 28, 2025	1	Updated /recent backend route to return full itinerary data including extra_data.image_url Fetched 3 most recent itineraries after login and saved them to AsyncStorage Loaded recent itineraries from AsyncStorage on HomeScreen mount Replaced horizontal ScrollView with FlatList for smoother trip card scrolling Matched FlatList styling to trip card design with snap behavior Fixed vertical gap between "Trip Plans" title and trip cards by adjusting styles
3/28/25 10:00 PM	Mar 28, 2025	0.5	Refactored ProfileScreen layout and logic for clarity and modularity Implemented Firebase data fetching using onValue and fallback with AsyncStorage Added calculateCompletion to compute profile completeness Resolved bug where calculateCompletion was undefined Displayed profile completion percentage with a styled progress bar Improved UX by turning progress bar green when completion reaches 100%
3/28/25 10:30 PM	Mar 28, 2025	0.5	Created AddToItineraryModal.js for selecting an itinerary. Fetched and combined personal and shared itineraries with type field. Sorted itineraries by updated_at. Added icon to distinguish between personal and shared itineraries. Moved icon from left to right of each itinerary item. Integrated modal into InteractiveRecommendations.js. Passed selected place and added it to Firebase under places. Prevented duplicate place entries and showed alerts for success or duplicate.
3/28/25 11:00 PM	Mar 28, 2025	2	Refactored AchievementsScreen to horizontally center achievement grid Adjusted columnWrapperStyle and gridItem width/margin for proper alignment Improved modal layout and progress bar width responsiveness Fixed AchievementsScreen container not being horizontally centered Resolved flicker of LocationPermissions by adding locationPermissionChecked Deferred rendering of HomeScreen until location check completes Ensured smoother transition from LocationPermissions to HomeScreen Moved itinerary fetching logic into useFocusEffect for consistency Re-enabled and integrated deleted useEffect logic for recent itinerary cards Fetched itinerary list from FastAPI using authenticated user ID Loaded recent itineraries from AsyncStorage fallback Updated recentTrips state to ensure trip cards render in Trip Plans section Cleaned up setUserId, onboardingComplete, and travel_style_id logic Improved overall HomeScreen onboarding and weather experience
3/29/25 12:00 PM	Mar 29, 2025	1.25	Adding Toggle mode for the Onboarding Checklist
3/29/25 1:15 PM	Mar 29, 2025	0.25	Added updateRecentTripsInStorage function to ItineraryDetailScreen Called updateRecentTripsInStorage after successful image upload in uploadImage Synced logic with ItineraryFormScreen to keep AsyncStorage updated with recent trips

3/29/25 1:30 PM	Mar 29, 2025	0.1	Added GestureHandlerRootView to wrap the entire app in App.js Identified crash in InteractiveRecommendations caused by gesture conflict with Modal Confirmed that AddToItineraryModal used FlatList and TouchableOpacity inside Modal Verified MapCheckInScreen worked fine because it did not use Modal Explained that wrapping GestureHandlerRootView globally prevents gesture-related crashes Confirmed nested GestureHandlerRootView is safe and does not cause issues
3/29/25 1:40 PM	Mar 29, 2025	2	Removed FlatList inside ScrollView to eliminate VirtualizedList warning Replaced FlatList with manually rendered list using .map() and BottomSheetScrollView Used cardRefs and scrollViewRef for smooth manual snapping to cards Refactored onPress logic so tapping a marker scrolls to and highlights the correct card Refactored onPress logic so tapping a card animates to the marker and shows callout Introduced activeIndex state to highlight active card with custom style Integrated @gorhom/bottom-sheet for swipeable modern layout with snapping positions Made travel style filter bar transparent to create a floating button appearance Preserved category filter inside bottom sheet just above the card list Preserved modal functionality to add a place to itinerary Ensured all components work on both local and Heroku environments Code is now cleaner, more performant, and designed for smoother mobile interaction
3/29/25 3:40 PM	Mar 29, 2025	0.1	Add Firebase profile image fetch in LoginScreen Save profile image to AsyncStorage after login Reuse saved image in MoreMenu for display Ensure profile image updates on ProfileScreen also update AsyncStorage Keep image synced and consistent between login, profile, and menu
3/29/25 3:50 PM	Mar 29, 2025	0.25	Passed itineraryId as prop to OtherCostsModal Fixed Firebase saving and deletion for other costs Improved modal styling with pill-shaped buttons and padding Styled cost type and subtype chips with horizontal scroll Enhanced text inputs with consistent padding and font size Updated saved cost list with cleaner layout and spacing Replaced "Remove" text with FontAwesome trash icon for deletion
3/29/25 4:45 PM	Mar 29, 2025	0.25	Added checklistRefreshTrigger state in HomeScreen Updated Firebase writes for weather_changed and packing_tip_viewed to increment the trigger Passed refreshTrigger prop to OnboardingChecklist Updated OnboardingChecklist to refetch checklist when refreshTrigger change
3/29/25 5:00 PM	Mar 29, 2025	0.25	Updated ItineraryFormScreen to use KeyboardAvoidingView, ScrollView, and TouchableWithoutFeedback for proper keyboard behavior Wrapped entire screen with keyboard-safe layout to prevent input fields from being blocked Refactored DestinationSearchModal to expand fully with flex: 1 Removed gray overlay issue by adjusting modalBox and backdrop styles Verified modal behaves correctly when keyboard is active and now appears clean without dimming artifact

3/29/25 5:15 PM	Mar 29, 2025	0.25	Fixed issue where plus button in callout was not clickable Changed Callout to use tooltip={true} Wrapped plus icon inside CalloutSubview Ensured styling keeps plus button on the right side Retained custom design with name, category, and plus icon layout
3/29/25 5:30 PM	Mar 29, 2025	0.5	Set selectedDate to itinerary's start date when adding a new day Used getNextAvailableDate() to ensure consistent default value Improved flow so users don't have to manually pick a starting date Reduced friction when creating the first itinerary day
3/30/25 1:30 PM	Mar 30, 2025	0.25	Wrapped place name text in a flex container Added placeTextWrapper style for layout control Enabled text wrapping with flexWrap: 'wrap' Added optional numberOfLines and ellipsizeMode for truncation Ensured buttons stay aligned with long text
3/30/25 1:45 PM	Mar 30, 2025	0.25	AddActivityModal extracted into its own file under components Refactored ItineraryDayScreen to use AddActivityModal Made display time default to current time when opening modal Preserved time picker selection and form behavior across both create and edit modes
3/30/25 2:00 PM	Mar 30, 2025	2	Added new feature to let users tap a place and assign it directly to a day Closed PlacesModal when a place is tapped If there are days, opened DaySelectionModal after short delay Once a day is selected, prefilled activity data and opened AddActivityModal Initialized time with current time and set place name as default activity If no days exist, switched to Days tab and opened Add Day modal automatically Pre-filled new day with next available date and cleared title Updated tab index using setIndex to show the Days screen Handled state transitions to avoid modal overlap Improved UX by streamlining the flow from selecting a place to planning it in a day
3/30/25 4:00 PM	Mar 30, 2025	0.25	Fixed crash in ItineraryDayScreen by removing strict UUID length check Ensured dayId presence check is simple and non-blocking for API call
3/30/25 4:15 PM	Mar 30, 2025	0.25	Retrieved profile image URL from AsyncStorage using profileImageUri key Stored S3 HTTPS image URL during profile setup Updated ChatbotScreen to load user avatar from S3 if available Added fallback to default image if no URL or invalid Added optional image prefetching for debug validation Ensured profile image renders only if it starts with http

3/30/25 4:30 PM	Mar 30, 2025	0.25	Updated card position to bottom of the screen with keyboard support Set card width to full screen Added tagline "Your Journey, Your Way." above the card Added main title "EXPLORE" and "THE WORLD" in uppercase above tagline Styled login button with full width and pill shape Matched signup button style to login button style for consistency
3/30/25 4:45 PM	Mar 30, 2025	0.25	Added AuthLoadingScreen to check AsyncStorage for user session Implemented logo zoom and fade animation using Animated API Adjusted logo scale for calm and smooth animation with easing Matched initial logo size to avoid starting too small Introduced 0.5s delay before animation for breathing room Updated transition to Main or Login screen after animation Investigated iOS white screen and found solution but did not apply Swift update
3/30/25 5:00 PM	Mar 30, 2025	0.25	Fixed crash caused by accessing user.id when user was null Added null check for user before making DELETE request Improved error message to prompt re-login if user not found Ensured day deletion only proceeds when user is properly loaded
3/30/25 5:15 PM	Mar 30, 2025	0.25	Fixed overlapping issue by adjusting top margin of filter bar Used useSafeAreaInsets to dynamically set spacing from top Replaced SafeAreaView with View and applied inline safe area style Ensured filter buttons stay clear of status bar and clock Added optional horizontal scroll for better layout on smaller screens
3/30/25 5:30 PM	Mar 30, 2025	0.25	Added isSheetExpanded state using useState Set isSheetExpanded to true when BottomSheet index is 2 using onChange Wrapped Travel Style Filter with conditional !isSheetExpanded Updated BottomSheet snapPoints to include 85 percent max height Preserved map interaction and filter logic during transition
3/30/25 9:30 PM	Mar 30, 2025	0.25	Wrapped DestinationSearchModal with React Native Modal Updated backdrop style to include semi-transparent background Set modalBox height to full screen using flex: 1 Ensured keyboard behavior using KeyboardAvoidingView Matched layout and behavior with WeatherSearchModal
3/30/25 9:45 PM	Mar 30, 2025	0.25	Wrapped modal in KeyboardAvoidingView for better keyboard handling Added ScrollView to allow scrolling when keyboard is open Implemented TouchableWithoutFeedback to dismiss keyboard on background tap Adjusted styles to support new layout behavior

3/30/25 10:00 PM	Mar 30, 2025	0.25	Updated cost type and subtype selectors with section labels Wrapped modal content with KeyboardAvoidingView for better keyboard handling Added ScrollView to allow scrolling when keyboard is open Improved overall user experience for input and selection in modal
3/30/25 10:15 PM	Mar 30, 2025	0.5	Moved parseToSortableTime function to the top of the file Updated DayCard to sort item.activities by time using parseToSortableTime Ensured consistent activity order matching ItineraryDayScreen Preserved original UI layout and logic in DayCard
3/30/25 10:45 PM	Mar 30, 2025	0.25	Fixed crash by adding optional chaining to itinerary date. Added budget, notes, and places panels to OverviewTab. Created and moved styles to ItineraryDetailScreenStyle.js. Updated OverviewTab to support external panels and style structure. Verified that screen renders correctly with tab view and modals.
3/30/25 11:00 PM	Mar 30, 2025	0.25	Modified addPlace function to immediately update Firebase after adding a place Removed the need for pressing the Save button Updated UI logic to optionally remove the Save button and keep only the Close button