PROJECT REPORT - PACK & GO

GROUP 2: LAUREN COLE, LOUISA RHODES DE GREY, MOIRA CORRADINI. SHEETAL VARSANI, SHIROMINI SATKUNARAJAH

INTRODUCTION

Our Aims & Objectives

Our project aims to allow a user to see information about countries that they want to visit such as language, currency, weather and a map of the area. We want the user to choose and save their own custom bucket list of potential destinations that they would like to visit. Based on their trip requirements the site will generate a recommended packing list depending on the chosen country and weather.

The primary objective was to allow a user to generate a personalised packing list based on their holiday type, and a secondary objective included the user being able to make a "bucket list" of destinations they would like to visit

Roadmap of the Report

This report will go into detail of various aspects of our project, starting with an introduction to the project background, followed by specifications and design details. The implementation process, tools used, and the team's development approach will be shared and the report will further explore the testing and evaluation strategies used through the project. Finally, we will conclude with an overview of the project and potential areas for future improvement. Additionally, you can find figures referenced in the report in the appendix.

BACKGROUND

The purpose of this web app is to help people make a list of things they would need to pack when they go on holiday, so that they don't forget to pack the essentials. We will also allow our users to add 'custom' items to their list.

Our target audience is anyone who is planning a trip abroad and loves being organised, but doesn't have time to make a list. Also many people get travel anxiety, so this app will help with the packing part of travelling for those that need it..

SPECIFICATIONS AND DESIGN

Requirements

Our project's specifications were outlined through a collaborative effort involving both technical and non-technical requirements. The technical requirements encompass a working knowledge of HTML, CSS and JavaScript, and integrating this knowledge into a fully working React app. Team members also needed knowledge in the use of GitHub and how to interact with Git by using the command line (in order to clone the repository, make pull requests and review branches).

The non-technical requirements focused on the principles of Agile product development and management, frequent and clear team communication, and product design (wireframing). These specifications were crucial in guiding the project's development.

Design and Architecture

The main purpose of our project was to help travellers pack for their trip, so we wanted the homepage to be sleek and simple with call to action buttons to help the user navigate through the web app.
We chose a minimal and neutral colour palette, with warm tones. And made sure to have dark colours and light colours for contrast. We made sure to use light coloured text on dark backgrounds and dark coloured text on light backgrounds for accessibility and ease of reading. With browns representing outdoors, and blues being tied to trust and peace, we felt these colours went best for our project.

1. **Primary User Flow:**
   Homepage > Pack List > Pack List Results > Weather Forecast *(Appendix Fig. 1)*.
   The user would start on the homepage, and be able to generate a personalised packing list based on their holiday type on the "pack list" page. After the user has had their packing list, they would also be able to see the weather forecast for the next 7 days at their destination should they wish to; if the user is travelling soon, this information would help them with their packing further.

2. **Bucket List Page:**
   On the Bucket List page, users can search for a destination and add it to their 'bucket list' *(Appendix Fig. 2)*.

3. **Navigation:**
   Users can navigate to the About page, where they can find information about the creators and read testimonials from previous users *(Appendix Fig. 3)*. The About, Bucket List, and FAQs pages, along with the Weather Forecast call-to-action, provide users with extra information and help in trip planning. For further information, users can reach out via the Contact Us page *(Appendix Fig. 4)*.

4. **Login/Sign-up Page:**
   Using the login/sign-up page, users can save their packing list and bucket list *(Appendix Fig. 4)*.

IMPLEMENTATION AND EXECUTION

Development Approach

We completed our SWOT analysis (*see Appendix Fig. 5*) and proposed that each member should present a quick elevator pitch for an idea during a follow-up call. The Travel Bucket List app received the most votes, becoming our chosen project. We then discussed potential features to establish the Project Vision (*see Appendix Fig. 6*). We gathered these ideas and conducted a MoSCow analysis (*see Appendix Fig. 7*), which then formed our project Epics. Due to project time constraints, we decided to work in a single sprint. Instead of creating backlog tickets, we assigned one page per team member to minimise code conflicts. This decision stemmed from varying team member availability, which we tracked using a calendar on the MIRO board (*see Appendix Fig. 8*).

We had 2 stand up calls a week for the most part to discuss what we were working on, if we had any blockers and or any suggested improvements. This, along with frequent communication in Slack meant that we did not have any large dependency delays and we were able to review and merge branches in GitHub effectively. The work was also tracked on the Sprint Board (*Appendix Fig. 9*) and we referred back to, and amended, our design wireframe as we progressed.

Team Member Roles

Lauren
- Transcribed group meetings
- Built the About and FAQ page of the web app
- Attempted creation of a backend in order to implement a login

Louisa
- Started the MIRO board for us to track the project from ideation to live; this included reference material on the SDLC, MoSCoW analysis and the sprint board
- Did a call on agile ceremonies for the group
- Built the Packing List page where so the user can obtain their packing list, and made it customisable by giving trip options
- Used customer AI generated images for the buttons on this page

Moira
- Built the MapSearch part where the user can explore countries and add them to My bucket list
- Create MapComponent to fetch and display an interactive city maps using OpenStreetMap API
- Create CountryDetails for more country info
- Build the Home Page
- Put in a custom AI-generated logo
- Develop a user authentication with Node.js Express by creating secure login and signup API endpoint
- Utilised Axios for making HTTP requests from the frontend to the backend API

Sheetal
- Recorded group meetings and uploading recordings to shared folder
- Designed initial wireframe with input and feedback from team
- Front-end work on Header, Footer, Testimonial-Carousel components
- Built initial front-end Sign-Up/Login Page
- Built Contact Us Page

Shiromini
- Found location and weather APIs we could implement in our app
- Built the page to show current-weather and weather-forecast in chosen location using the APIs

Tools and Libraries

We made the development process smoother for our project by using different tools and libraries. This helped us work together better and be more efficient. We used important version control systems and project management tools in our toolkit. We also added some libraries to make our functions better.

Front-end Development:
For the front-end, we employed HTML, CSS, and JavaScript, with React as our primary framework and React Router for seamless navigation. Bootstrap served as our CSS framework, and Animate.css added animations to our pages. React Slick, a slider component, was implemented and customized to meet our specific design and functional requirements for our Testimonial Carousel.

Pack List Page:
On the Pack List page, we used JSON to display packing items, ensuring easier maintenance and flexibility for a dynamic list. Material UI was implemented for its accordion component, and Dall-E 2 was utilized for image generation.

Mapping Features:
The project's mapping features were powered by Mapbox, complemented by Geocoding with Nominatim (OpenStreetMap) to translate place names into precise map locations.

Weather Forecast Features:
For the weather related functionalities, npm packages such as 'react-select-async-paginate' were integrated for querying APIs and presenting options as a dropdown. Additionally, 'React-accessible-accordion' was used to create an accordion menu with expandable and collapsible sections.

Sign-up and Login Functionality:
For sign-up and login functionality, Express acted as the backend framework,complemented by Prisma for database management. MySQL Workbench handled database operations. Bcrypt ensured secure password hashing, Axios facilitated API interactions, and Dotenv handled sensitive information like API URLs and database credentials.

Testing and Organization:
React testing library and Jest were employed for comprehensive testing. Babel made JavaScript compatible across all browsers, while Webpack organized the site. CORS (Cross-Origin Resource Sharing) ensured smooth communication between the client and server, and bodyParser served as a middleware tool for extracting data from incoming requests. The useState hook managed component states, and useEffect handled tasks such as fetching data from an API.

Agile Development

The iterative approach came through very naturally in this project as we started with the design informing the look and planned for group members to build individual pages. As the pages were built we were then able to combine elements on the pages e.g. the testimonials on the About page and the 7 day forecast on the packing list. The code for these pages was branched, developed, tested and deployed, all team members reviewed the code throughout the build and feedback and suggestions were given in Slack and on Teams.

Implementation Challenges

Pack List Functionality:
The initial design of our web app aimed to provide users with the ability to use a checklist, allowing them to remove unwanted items. However, formatting issues arose when implementing the accordion component, preventing the exclusion of individual items while leaving out entire sections. Despite exploring various options, such as radio buttons and checkboxes, none were found to maintain accessibility without skewing the containers.

Integration of External APIs:
Finding the right API for location and weather was a bit of a challenge. At first, we wanted to use a weather API to predict conditions at the user's destination during their travel dates. But predicting weather accurately beyond a week turned out to be tough. So, we had to stick with a 7-day limit. We thought about using historical weather data from last year to give a rough idea of the weather during their trip, but the API required a paid subscription for historical data so that wasn't feasible.

Additional API Challenges:
As well as the weather-related challenges, we encountered hurdles in finding an API that could provide information on the capital, currencies, languages, and timezone of a given location. Additionally, creating a secure login system posed difficulties, requiring a seamless connection between the frontend and backend while effectively managing user authentication and security.

TESTING AND EVALUATION

Testing Strategy & Breakdown

Our comprehensive testing strategy covered various aspects, employing both automated and manual approaches. Functional testing rigorously assessed individual functions to ensure they aligned with specifications.User testing involved employing usability studies and collecting feedback from our target users.

Pack-List Testing:

Approach:
- Used React Testing Library
- Used 'getByAltText' to confirm the presence of image buttons
- Used 'waitFor' to simulate user actions (e.g. clicking the button to reveal the accordion)

Outcome:
- 3 tests were run, all passing
- Verified the Pack List page displays 4 image buttons to user
- Confirmed that clicking a button opens the accordion and tests a section's existence
- Ensured that clicking on "Snow" brings up the relevant rendered content

Home Component Testing:

Approach:
- Used React, MemoryRouter, and Home components from the React library
- Used 'render' to render the component and 'screen' to interact with elements
- Defined test cases using 'test' and selected elements using query methods like 'getByText', 'getByRole', and 'getByAltText'

Outcome:
- The testing strategy details components, utilities, and methods used for element selection during testing.

MapComponent Component Testing:

Approach:
- Evaluated that the MapComponent renders without errors
- Used React Testing Library's 'render' function to render the MapComponent
- Selected a country name such as France to check the component's behavior
- Used screen query methods such as 'getByTextId' to verify elements

Outcome:
- Confirmed that the MapComponent renders without crashing.

MapSearch Testing:

Approach:
- Used 'render' and 'screen' from React Testing Library
- Imported the MapSearch component for testing
- Defined a test case using 'test' to check if the search input works
- Used 'screen.getByPlaceHolderText' to select the search input and verified its presence in the HTML document

Outcome:
- Confirmed that the search input in the MapSearch component renders as expected.

Testimonial Carousel & "Back to Top" Testing:

Approach:
- Utilized React Testing Library to test the Testimonial Carousel and Back to Top button
- Checked the rendering of the carousel component
- Confirmed the rendering of the tagline
- used 'fireEvent' to simulate clicking "Back to Top" and trigger a smooth scroll to the top

Outcome:
- Validated the expected behavior of the Testimonial Carousel and Back to Top functionality.

This diverse testing approach ensured that our web application underwent thorough evaluation, addressing various components and functionalities across the project. See screenshot showing our tests passed (*see Appendix Fig. 10*)

CONCLUSION

We were able to create a functional web app using React that closely resembles our original ideas and wireframe. Using Git allowed all team members to work on separate branches at any given time without disrupting other team members' work.

In the future, we would like Pack & Go to have full Android app and iOS app functionality and support. We would like users to be able to create packing lists for more than just themselves for example if they are travelling as a group.

We would also add a login feature right at the start of the website, to ensure users have immediate access to personalized features.

We are also contemplating adding a smart feature that suggests travel destinations using AI.The system will suggest countries to users based on their historical searches and preferences.

Having the Slack group chat and regular group meetings proved invaluable and we were all encouraged by one another to ask for help when we needed it. In instances where someone was having trouble whether it be with some code, a framework or GitHub, support was always offered by all team members.

APPENDIX

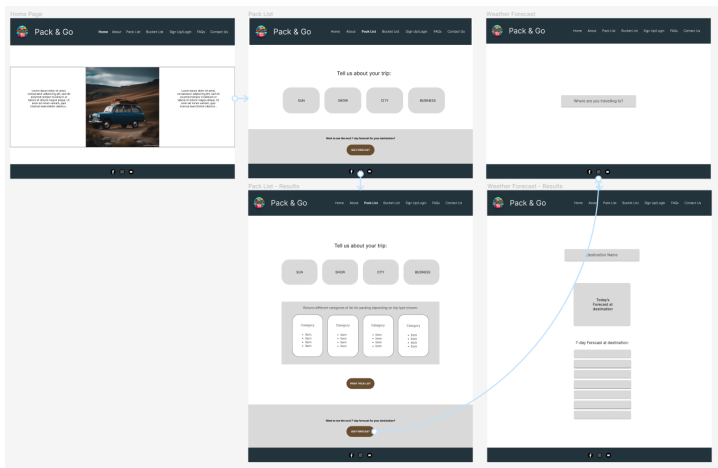*Fig 1.* The primary user flow is: Home Page > Pack List > Pack List Results > Weather Forecast:
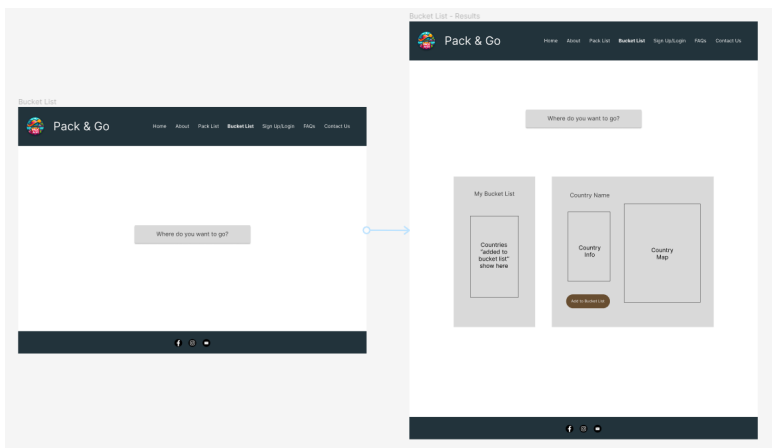

Fig 2. Bucket List Page:

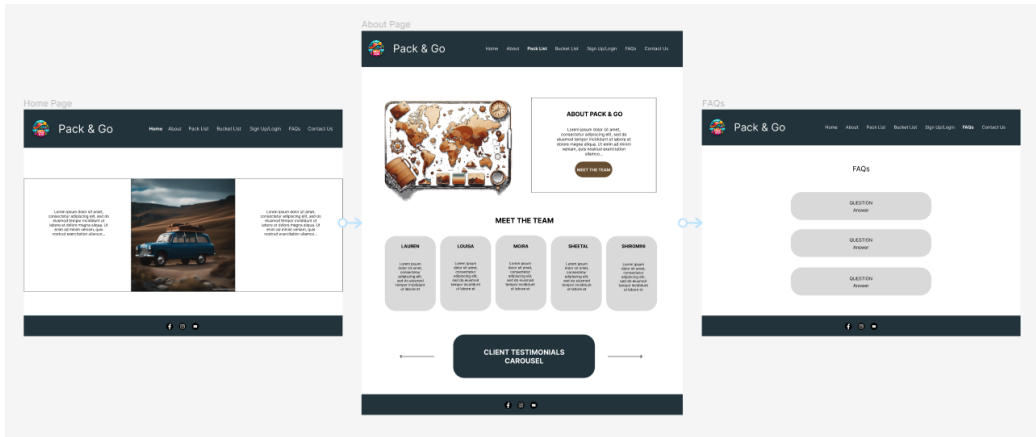
Fig 3. Homepage, About page, FAQs page:

Fig 4. Login, Sign Up, and Contact Pages

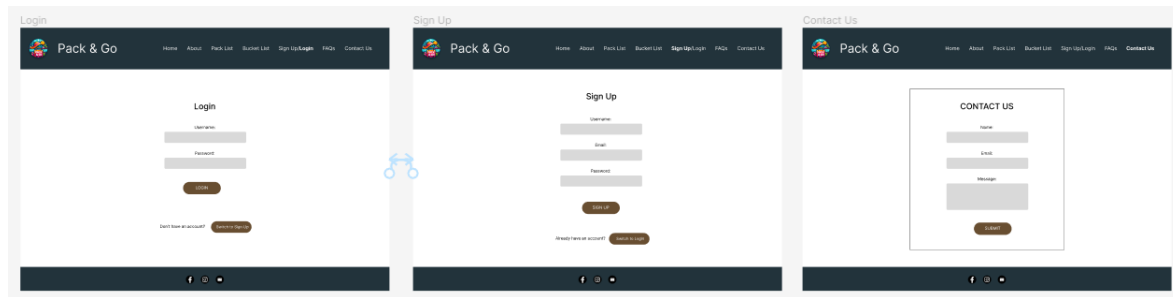*Fig 5.* SWOT Analysis

Fig 6. Project Vision

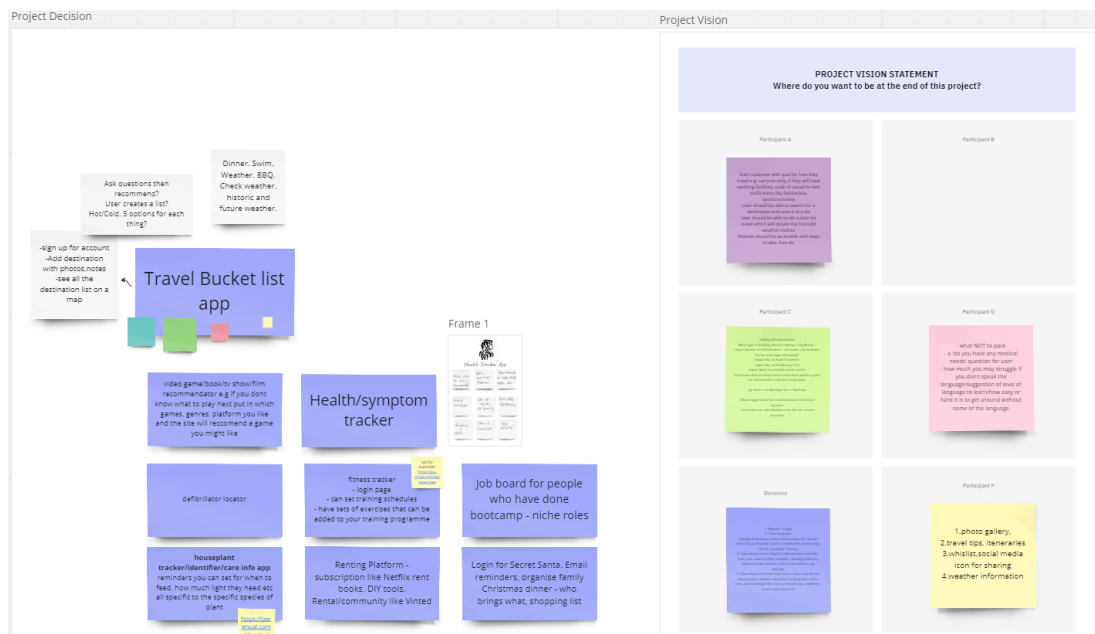*Fig 7.* MoSCow Analysis

MoSCoW Analysis

IDEA	MUST	SHOULD	COULD	WONT
User Login with Password		SS		
User Only 1 user per account (solo traveller)	SS			
User: Allow to make list of multiple family members/friends		SS		
Registration: quiz for how they travel e.g. carry-on only, if they will have washing facilities, scale of casual to new outfit every day fashionista. Outcome will dictate some of generated packing list		SS		
Bucket List Destination - allow user to set up x number of destinations - search for and to list	SS			
Holiday Type - options for weekend break, 2 week beach holiday, activity tour, business/digital nomad			SS	
Length of trip	SS			
ability to add items we don't have in database		SS		
Socials: user can upload photo				
Socials: Social media icon for sharing		SS		
Socials: Holiday Recommender				SS
Accessible website with adjustable font etc		SS		
Photo gallery,			SS	
User can set date of trip (used for weather api info)	SS			
Medical requirements		SS		
- how much you may struggle if you don't speak the language/suggestion of level of language to learn/how easy or hard it is to get around without some of the language			SS	
travel tips, iteneraries			SS	
Weather Info	SS			
App will generate things to take based on weather (info from external API), activities, country (currency, religious requirements, cultural acceptance e.g. tipping)		SS		
App will generate warning such as place can be too busy, book in advance because it is busy time of the year, can have high tide, storm, volcano, war, epidemic, travel restrictions etc			SS	

Fig 8. Calendar on Miro Board
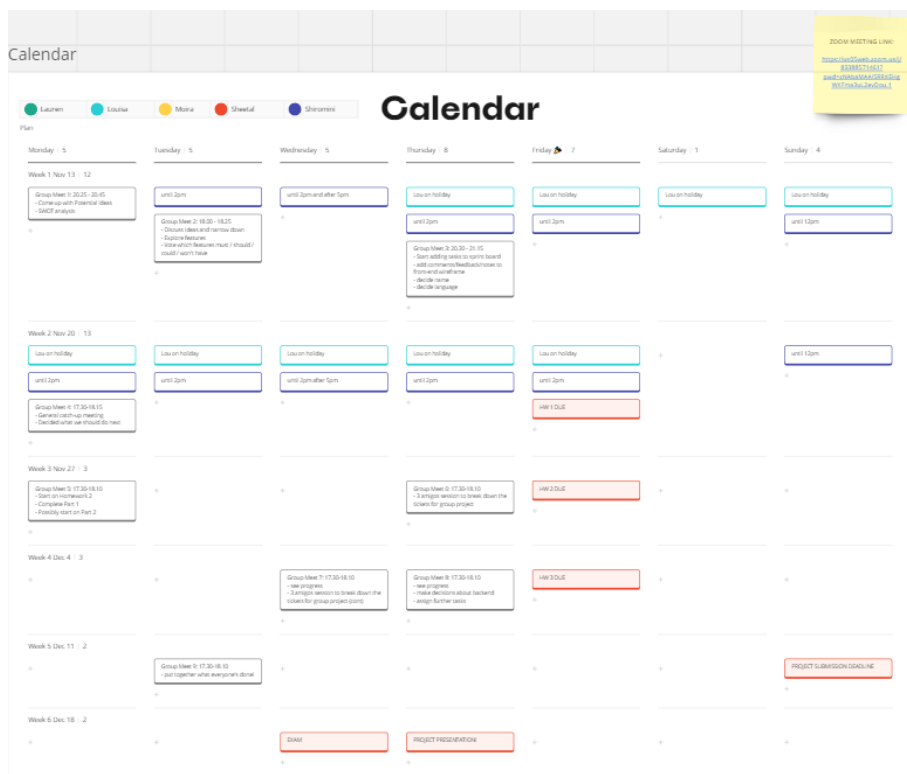
Fig 9. Sprint Board

Fig 10. Testing Results