# Practice project: Conference Expense Planner app

In the **Conference Expense Planner** app practice project, you will build the front end of an application that allows the user to price out the expenses related to hosting a conference at a convention center. The lab will guide you through the development of the application elements required for this project.

The **Conference Expense Planner** will have a landing page, a product selection page, and a pop-up summary window. The product selection page allows users to select their rooms, their add-ons, and meals. Based on those selections, it provides pricing. The pop-up window summarizes the expenses in a table based on the user's selections from the product selection page.

We strongly encourage you to complete the practice project prior to starting the final project. It will give you practice developing many of the same elements and functionality you will need to develop in your final project. The practice project lab contains step-by-step instructions for the code required in this application.

Let's discuss the features of the conference event budgeting app in greater detail. Sample screenshots are provided, though feel free to experiment with the code, components, and layouts.
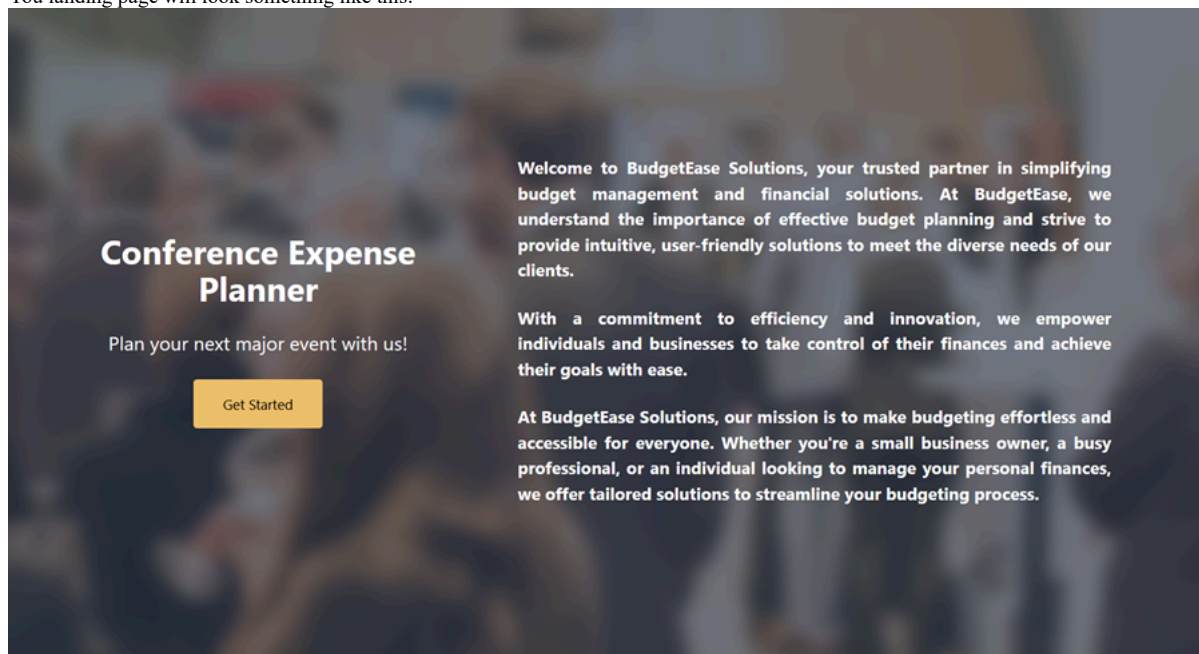
## Landing page

The landing page will have the same elements as the final project's landing page.

You will create a landing page like the image below that engages your customers with your product line and provides entry into the main application. See the screenshot at the end of this section for a visual of what this page might look like.

Features on this page should include:

- A paragraph about the company
- A background image
- Your company name and
- A **Get Started** button linking to the product selection page

You landing page will look something like this:



## Product selection page

After selecting **Get Started**, the application should direct the user to the product selection page. This page allows users to select rooms in the venue, add-ons needed for presentations such as microphones and speakers, and meals they would like catered for the participants. This page will have three sections: room selection, add-ons, and meals. Each section should have its own header and also a page header with navigation to each section.

The suggested layout is displayed in an annotated screenshot for each section and an image of the whole page.
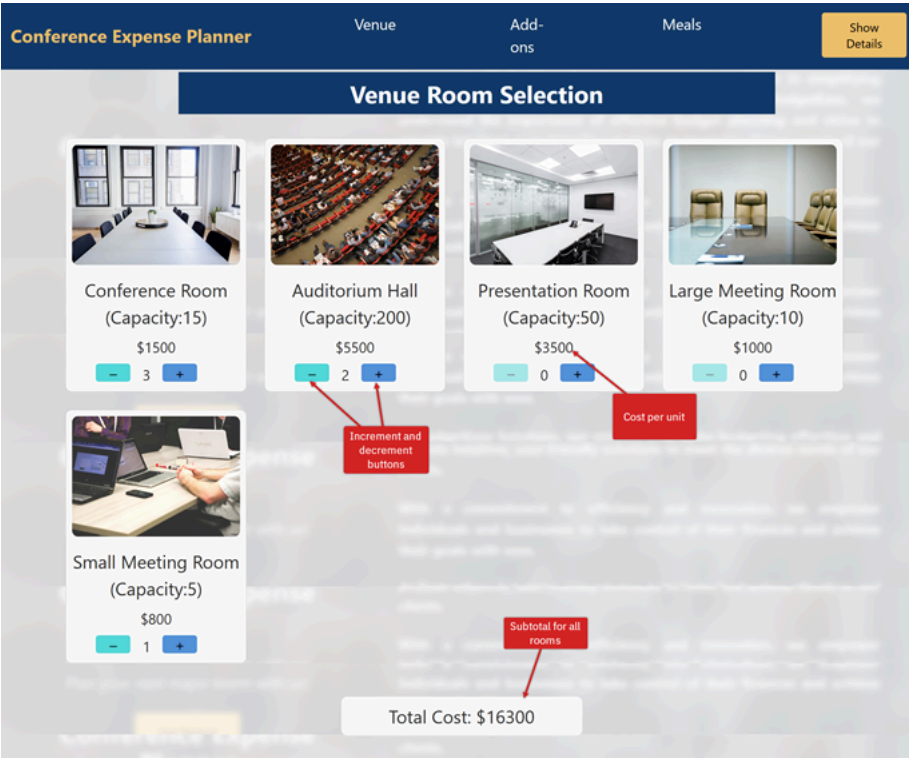
### Room selection and navigation bar

Users select from 5 room types:

- Auditorium hall – capacity 200, $5500 ea
- Conference room – capacity 15, $3500 ea
- Presentation room – capacity 50, presentation room, $700 ea
- Large meeting room – capacity 10, $900 ea
- Small meeting room – small meeting room, capacity 5, $1100 ea

You will provide an increment and decrement button for each room type so the user can request the number of each room type they need.

The product selection page should also have a header with a navigation bar. The navigation bar should display a **Show Details** button, which opens a pop-up window displaying data related to the user's selections.
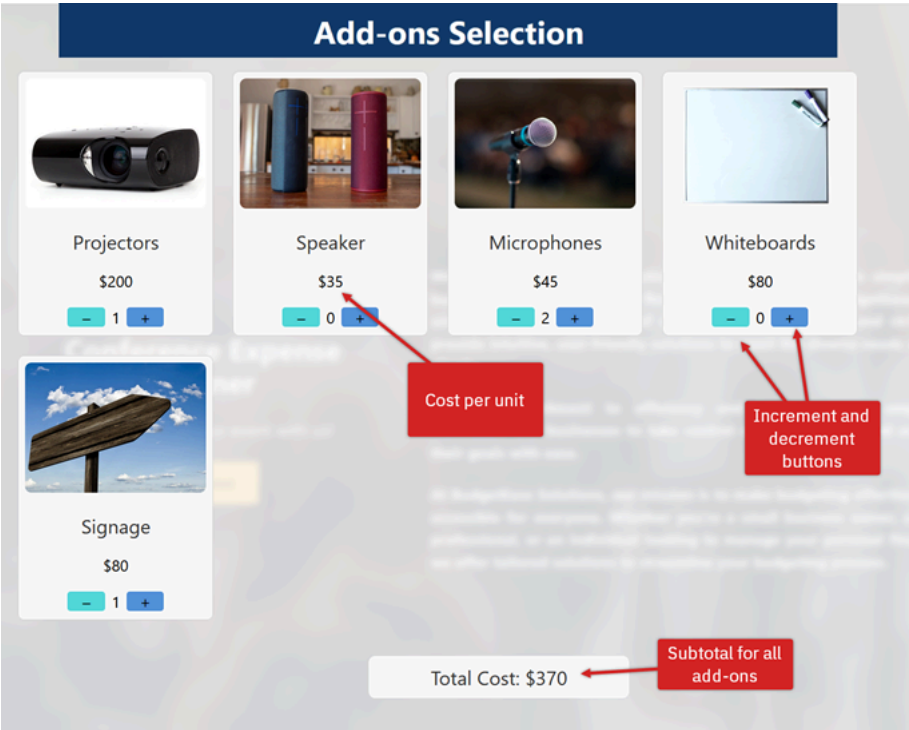
**Add-ons selection**

Users can select their audio/visual equipment in the add-ons section:

- Speakers – $35 ea
- Microphones – $45 ea
- Whiteboards – $80 ea
- Projectors – $200 ea
- Signage – $80 ea

You will provide an increment and decrement button for each type of equipment so the user can request the number of each type of AV equipment they need.



**Meals selection**

Users can select the following options for meals in the meals section:

- Breakfast – $50 per person
- Lunch – $60 per person
- High tea – $25 per person
- Dinner – $70 per person

You will provide a text entry box where the user can enter the number of people for these meals.

# Meals Selection

**Number of People:**  30

☐ Breakfast
$50

☑ High Tea
$25

☐ Lunch
$65

☑ Dinner
$70

Total Cost: $2850

Checkbox and price per person

Subtotal for all meals

**Sample product selection page (all sections):**

**Conference Expense Planner**     Venue     Add-ons     Meals     Show Details

## Venue Room Selection



Conference Room
(Capacity:15)
$1500
− 3 +



Auditorium Hall
(Capacity:200)
$5500
− 2 +



Presentation Room
(Capacity:50)
$3500
− 0 +



Large Meeting Room
(Capacity:10)
$1000
− 0 +



Small Meeting Room
(Capacity:5)
$800
− 1 +

Total Cost: $16300

## Add-ons Selection



Projectors
$200
− 4 +



Speaker
$35
− 4 +



Microphones
$45
− 2 +



Whiteboards
$80
− 1 +



Signage
$80
− 0 +

Total Cost: $1110

## Meals Selection

**Number of People:** 50

☐ Breakfast     ☑ High Tea
   $50             $25

☐ Lunch        ☑ Dinner
   $65             $70

Total Cost: $4750

### Show details pop-up

The users will need to see the details of their selections. For this element, they can access this information through a **Show Details** button in the header. When the user selects the button, a window pops up displaying a four-column table. Each row of the table contains the selection type, its unit cost, the quantity indicated, and the subtotal for the quantity of that item type. It will also display the total cost for all items.

TOTAL COST FOR THE EVENT

# $22160

| Name | Unit Cost | Quantity | Total Cost |
|------|-----------|----------|------------|
| Conference Room (Capacity:15) | $1500 | 3 | $4500 |
| Auditorium Hall (Capacity:200) | $5500 | 2 | $11000 |
| Small Meeting Room (Capacity:5) | $800 | 1 | $800 |
| Projectors | $200 | 4 | $800 |
| Speaker | $35 | 4 | $140 |
| Microphones | $45 | 2 | $90 |
| Whiteboards | $80 | 1 | $80 |
| High Tea | $25 | For 50 people | $1250 |
| Dinner | $70 | For 50 people | $3500 |

## Final project similarities

The final project has many elements in common with the practice project. For your final project, you will create a shopping cart application for purchasing household plants.

Similar functionalities between the two projects include:

- Landing page
- Navigation bar
- Array of product items and associated data
- Array of selected item
- Use of the `map()` function to iterate through an array
- Subtotal and total cost calculations
- Dynamic pricing displays based on user selections
- Increment and decrement buttons for product quantity selection
- Use of Redux and the Redux toolkit for managing application states with 'slices'
- Function components for presenting interactive UI elements