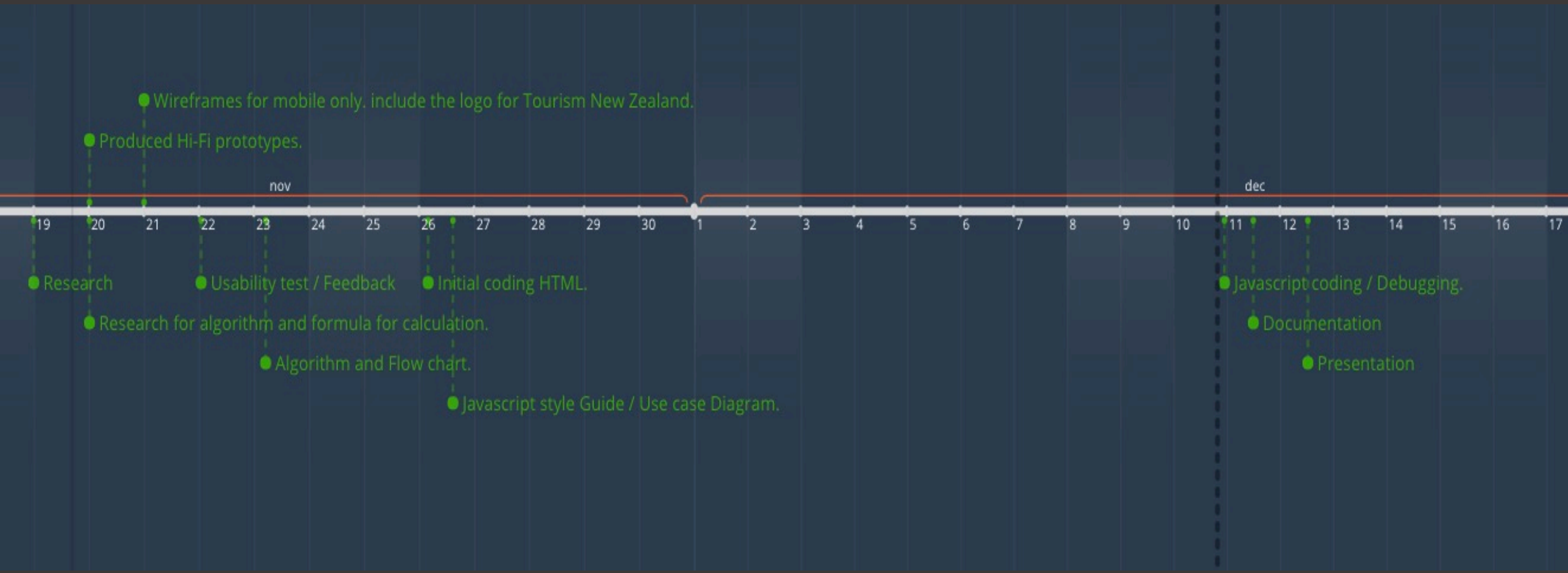
Accommodation SINGLE Web Page

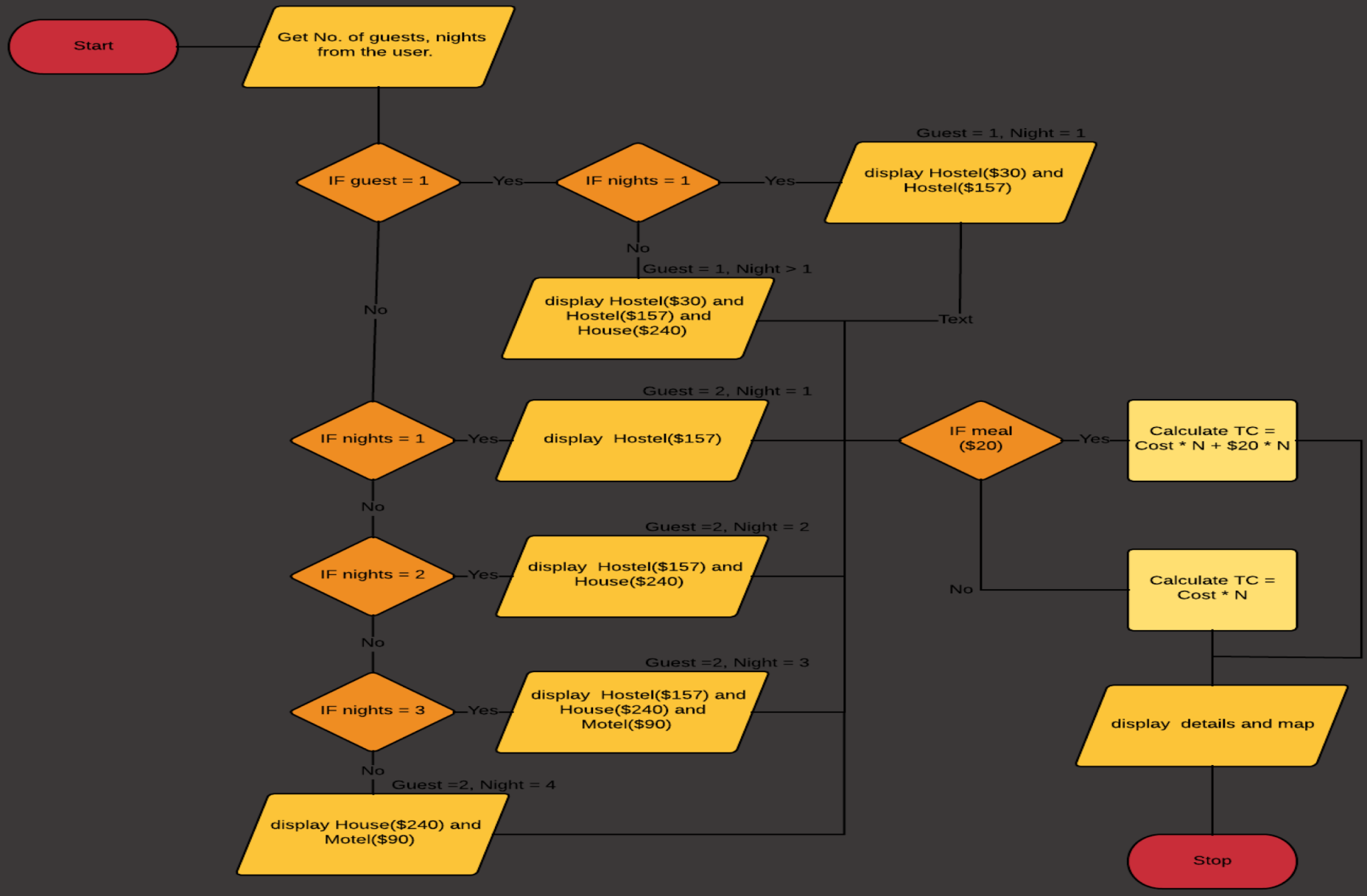
MODULE 2 FOUNDATION CODING SUMMATIVE

Joon Yoon

TIMELINE


Wireframes for mobile only. include the logo for Tourism New Zealand.
Produced Hi-Fi prototypes.
nov
dec
19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Research
Usability test / Feedback
Initial coding HTML.
Javascript coding / Debugging.
Research for algorithm and formula for calculation.
Documentation
Algorithm and Flow chart.
Presentation
Javascript style Guide / Use case Diagram.

FLOWCHART


Start
Get No. of guests, nights from the user.
IF guest = 1
Yes
IF nights = 1
Yes
Guest = 1, Night = 1
display Hostel($30) and Hostel($157)
No
Guest = 1, Night > 1
display Hostel($30) and Hostel($157) and House($240)
Text
No
IF nights = 1
Yes
Guest = 2, Night = 1
display Hostel($157)
IF meal ($20)
Yes
Calculate TC = Cost * N + $20 * N
No
IF nights = 2
Yes
Guest =2, Night = 2
display Hostel($157) and House($240)
No
Calculate TC = Cost * N
No
IF nights = 3
Yes
Guest =2, Night = 3
display Hostel($157) and House($240) and Motel($90)
display details and map
No
Guest =2, Night = 4
display House($240) and Motel($90)
Stop

ALGORITHM

Step 1. User presses the "Reserve your stay" button on the landing page.

Step 2. User inputs the "Check in" date and "Check out" date under Date & Guest tab of the modal.

Step 3. User inputs the How many guests are staying under Date & Guest tab and press next button.

Step 4. If user inputs 1 person for their accommodation, display the option of Hostel $30/night and Hostel $157/night. and proceed to Step 7.

Step 5. If user inputs 2 people for their accommodation, display the option of Hostel $157/night and Motel $90/night. and proceed to Step 7.

Step 6. If user inputs 3 or 4 people for their accommodation, display the option of House $240/night and Motel $90/night. and proceed to Step 7.

Step 7. Out to filtered two options, user chooses one of the options and add breakfast.

Step 8. Total cost including breakfast is calculated and displayed in the in the Total modal.

PARTIALS AND VARIABLES

/* Variables */

```
$dark-blue: #292F3A;
$blue: #30364C;
$teal: #73CDD9;
$red-orange: #F0573D;
$white: #FFFFFF;
$black: #000000;
$pure-red: #e60000;
```

/* FONTS */

```
$font: 'Exo 2', sans-serif;
```

/* Partials */

```
_buttons.scss
_maps.scss
_modal.scss
_result.scss
_variables.scss
_main.scss
```

JAVASCRIPT STYLE GUIDE

VARIABLES

- Use camelCase for identifier names (variables and functions). - All names start with a letter.

Example:

```
firstName = "Name";
```

SPACE AND OPERATORS

- Always put spaces around operators (= + - * / , and after commas.

Examples:

```
a = b + c ; ( a > b ) && ( a > c) ;
var values = [ "John" , "Sam" , "Sarah" ];
```

CODE INDENTATION

- Always use 2 spaces for indentation of code blocks.
  (NOTE: Do not use tabs for indentation. Different editors interpret tabs differently).

Example:

```
function sayHelloName() {
} document.write ( 'Hello Name' );
}
```

FILE NAMES

- Use lower case file names.

GENERAL RULES FOR COMPLEX (COMPOUND) STATEMENTS

- Put the opening curly braces at the end of the first line.
- Use one space before the opening bracket.
- Put the closing bracket on a new line, without leading spaces.
- Do not end a complex statement with a semicolon.

Conditional example:

```
if ( name = John ) {
document.write ( "Welcome back John" ); } else {
} document.write ( "Please enter your correct name." );
```

Function example:

```
function sayHelloName() {
} document.write ( "Hello Name" );
```

Loops example:

```
for (i = 0; i < 5; i++) {
X = i;
}
```

STATEMENT RULES (SIMPLE)

- Always end a simple statement with a semicolon.

Example:

```
var values = { firstName: "John" ,
}; lastName: "Smith" ,
};
```

OBJECT RULES (SIMPLE)

- Short objects can be written compressed, on one line, using spaces only between properties separated by commas.

Example:

```
var person = { firstName: "John" ,
}; lastName: "Smith" ,
};
```

GENERAL RULES FOR OBJECT DEFINITION

- Place the opening bracket on the same line as the object name.
- Use colon plus one space between each property and its value.
- Use quotes around string values, not around numeric values.
- Do not add a comma after the last property-value pair.
- Place the closing bracket on a new line, without leading spaces.
- Always end an object definition with a semicolon.
- When many properties are involved, span across multiple lines with one property-value pair on each line followed by a comma.

Example:

```
var person = { firstName: "John" , lastName: "Smith"
, age: 50 ,
eyeColor: "blue"
};
```

Short objects can be written compressed, on one line, using spaces only between properties, like this example:

```
var person = {firstName: "John" , lastName: "Smith" ,
age: 50 , eyeColor: "blue" };
```

LINE LENGTH

- For readability, avoid lines longer than 80 characters.
- If a Javascript statement does not fit on one line, the best place to break it, is after an operator or a comma.

Example:

```
var person = {
firstName: "John" ,
age: 48 ,
phoneNumber: 021 234 567, gender: "male"
};
```

JAVASCRIPT STYLE GUIDE

NAMING CONVENTIONS

- Variable and function names are written in camelCase.
- Do not use hyphens as they can be mistaken for subtracting attempts.
- Do not start names with a $sign as it conflicts with many Javascript library names.

Example:

```
var person = { firstName: "John" , lastName: "Smith"
, age: 50 ,
eyeColor: "blue"
};
```

LOADING JAVASCRIPT IN HTML

- Use simple syntax for loading external scripts (the type attribute is not necessary).

Example (including folder path):

```
<script src="js/script.js"</script>
```

COMMENTS

- Use // for single line comments.
- Use /* for multiple line comments.

DECLARATION

- Declare all variables, arrays and objects in the beginning of statement blocks. - Initialize in the beginning to avoid errors.
- Declare loop variables before the use in loops for faster loading effects.

Example:

```
var person = { firstName: "John" , lastName : "Smith"
, id: 55555 ,
fullName : function() {
 return this.firstName + " " + this.lastName;
}
};
```

QUOTES

- Use double quotes for string values

Example:

```
var person = {
} firstName: "John" ,
```

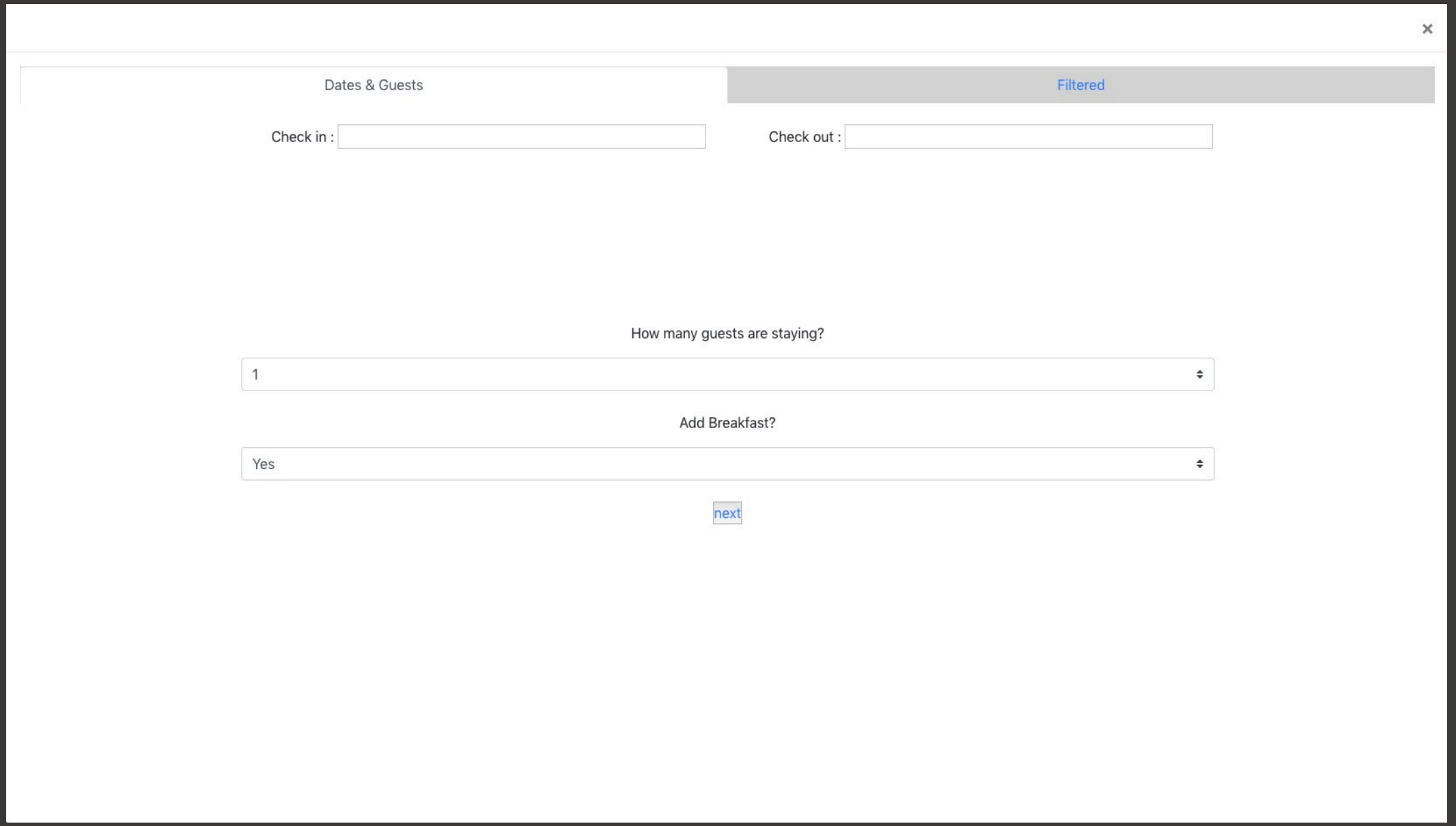
- Use single quotes for apostrophes .

Example:

```
var person = {
} firstName: "James'" ,
}
```

SCREENSHOTS

SCREENSHOTS



Booking for

187 Dollars

1 Days >

1 Guest(s) >

with breakfast ($30)

Ref#4747883

O'Reily Avenue, Te Aro, Wellington

(04) 473 7697

Start Over | Confirm