Guest360 CRM Project – Phase 6: User Interface Development

This document explains Phase 6 (User Interface Development) of the Guest360 CRM Project. It includes detailed click-by-click steps and screenshot placeholders for each UI configuration.

Step 1: Guest360 Page – Loyalty & Sentiment Data

Path: **Setup → Lightning App Builder → New → App Page**

Steps:
1. Setup → Lightning App Builder → New → App Page → Next → Name: Guest360.
2. Drag Tabs Component onto the canvas.
3. Create tabs: Details, Loyalty, Sentiment.
4. Loyalty Tab: Add Rich Text or Report Chart to display loyalty tiers (Gold, Silver, Platinum).
5. Sentiment Tab: Add Report Chart to display sentiment data (positive, neutral, negative).
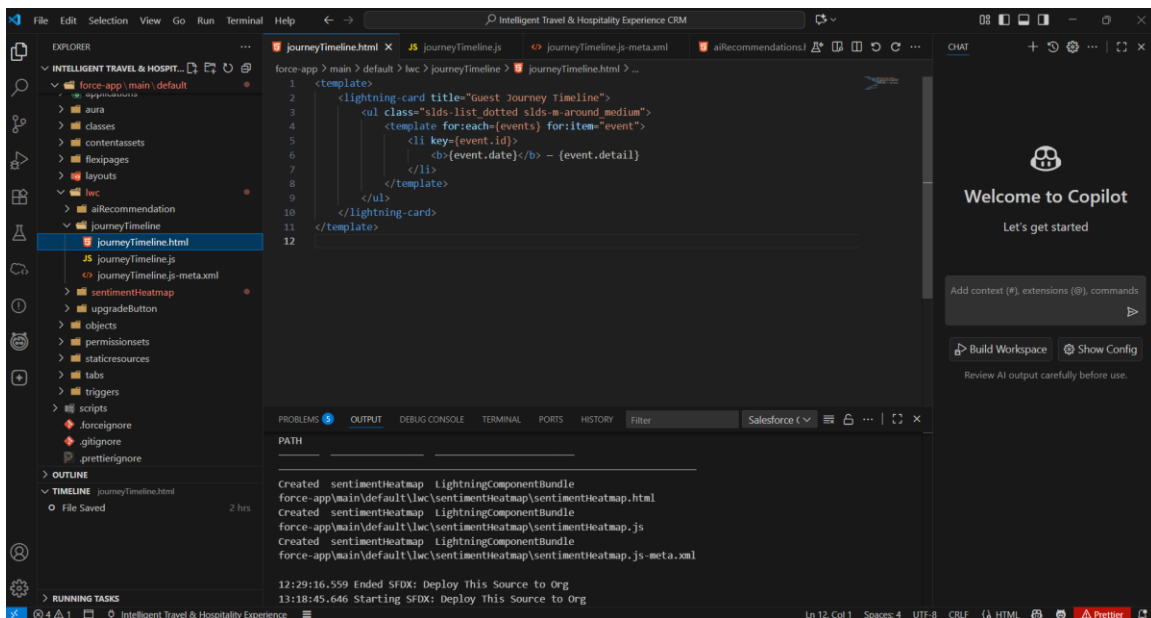6. Save → Activate.
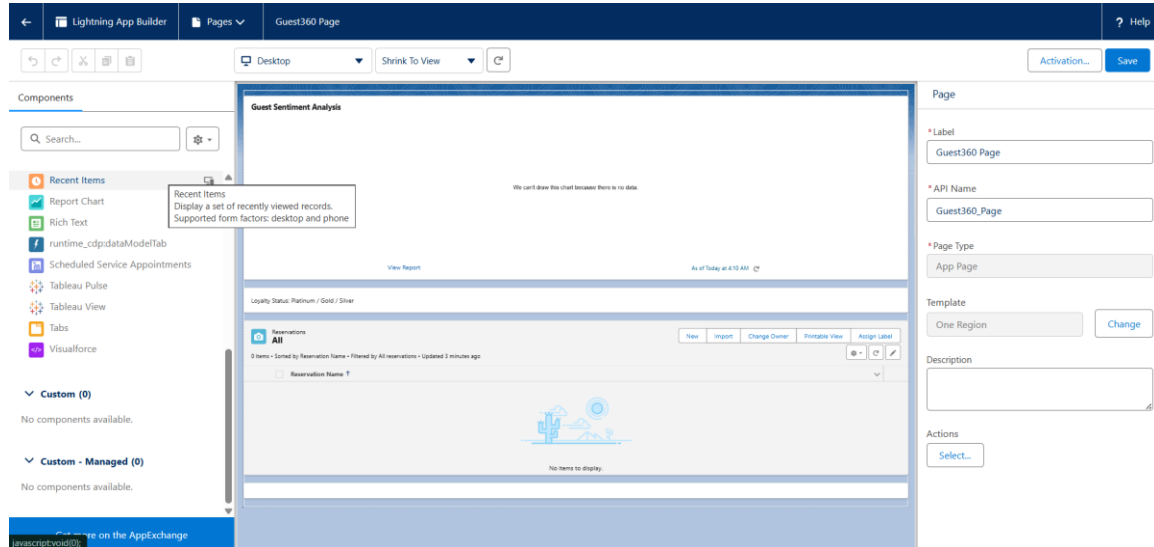
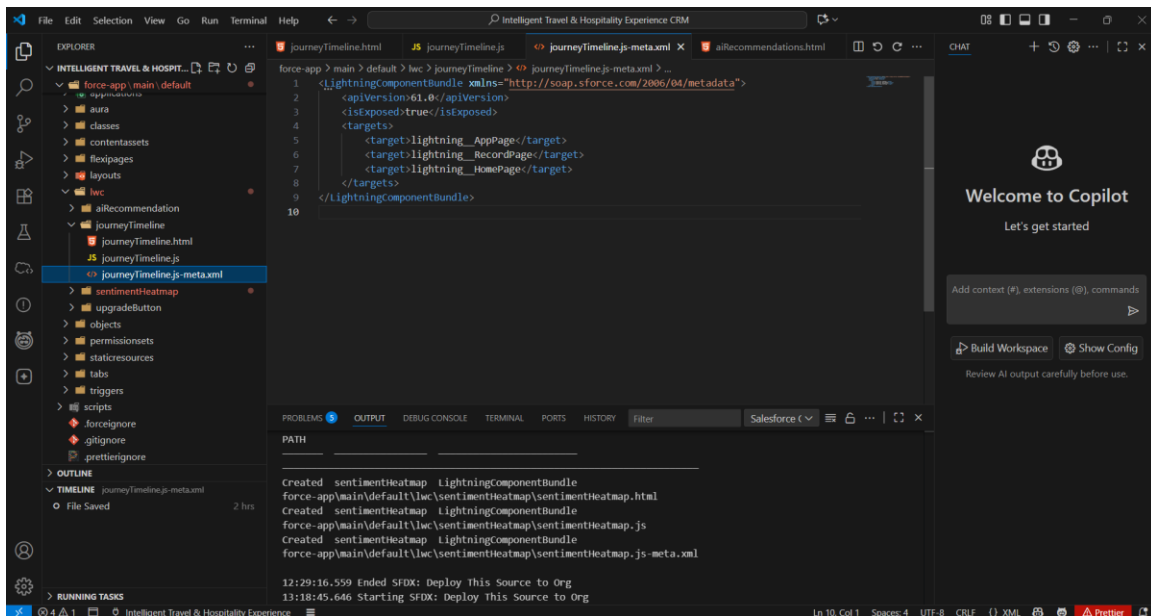## Step 2: LWC Component – Journey Timeline, AI Recommendations, Sentiment Heatmap.

LWC Component – Journey Timeline

Path: **Setup → Developer Console → New Lightning Web Component**

Steps:
1. Create LWC named journeyTimeline.
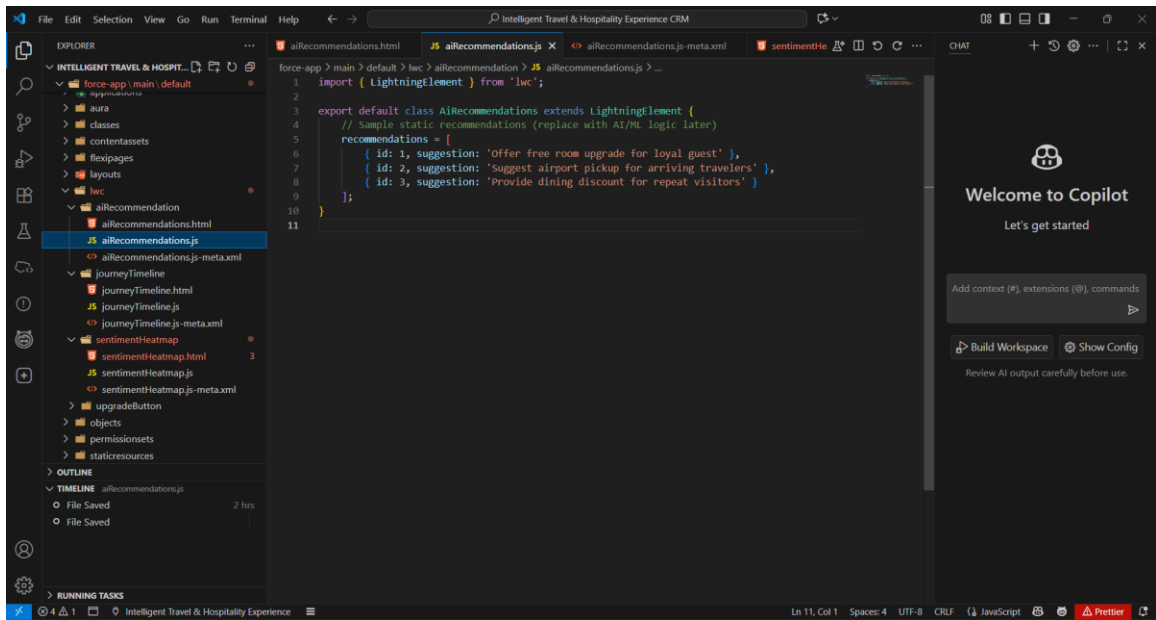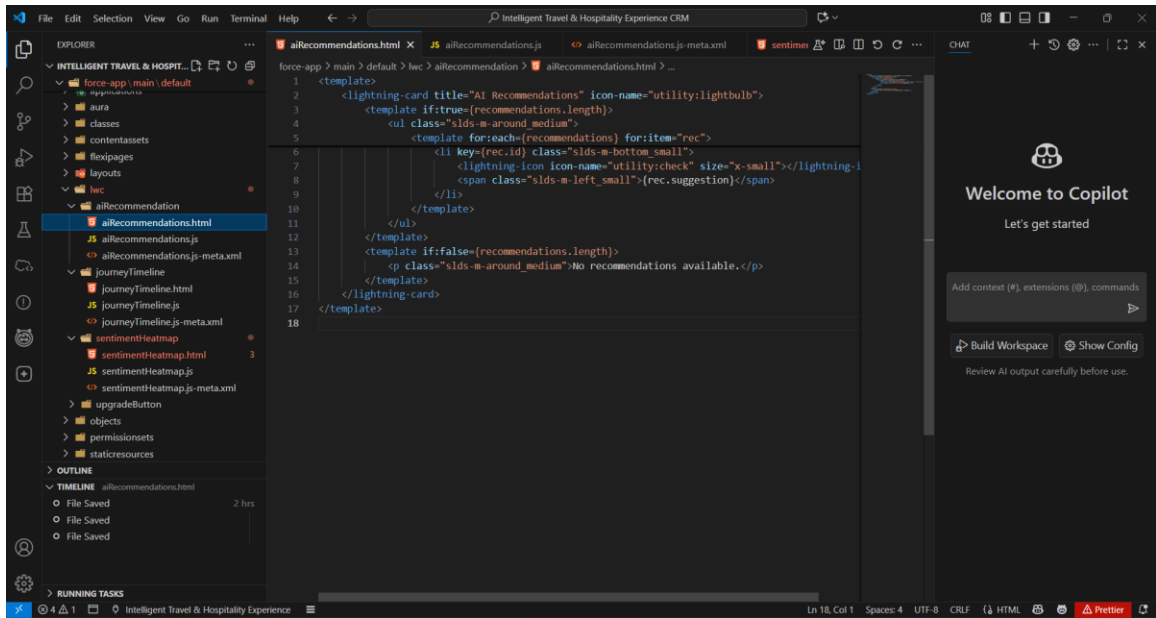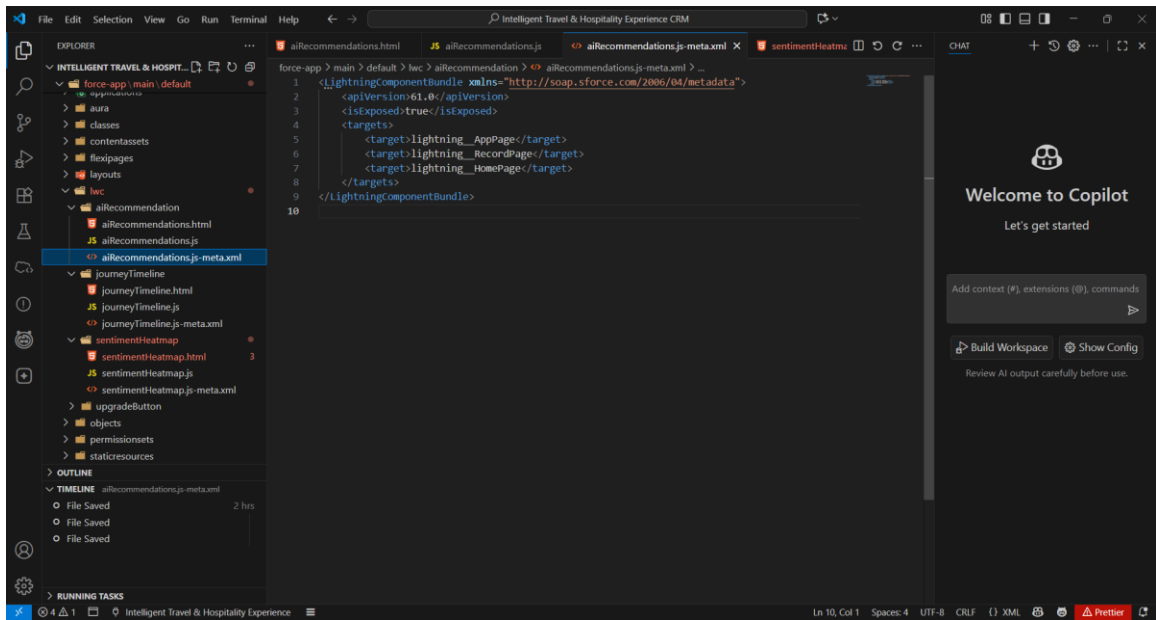2. Deploy component and add it to Guest360 Page.
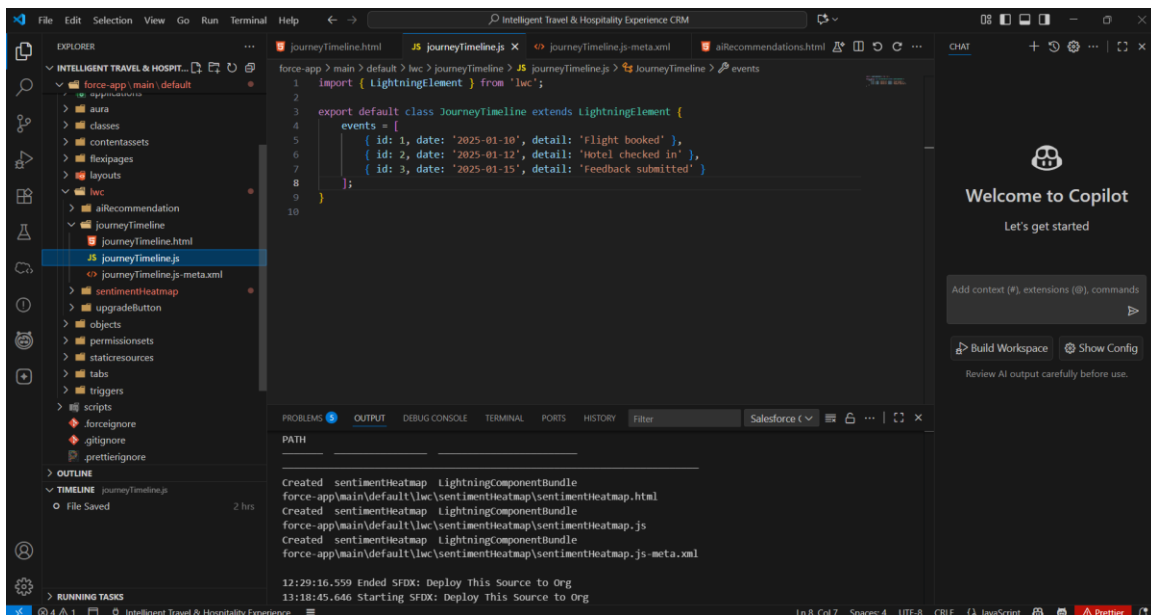
LWC Component – AI Recommendations

Path: **Setup → Developer Console → New Lightning Web Component**

Steps:

1. Create LWC named aiRecommendations.

2. Display recommendations in cards or list view with CTA buttons

3. Deploy component and add it to Guest360 Page.

Screenshot 1 — aiRecommendations.html

```html
<template>
    <lightning-card title="AI Recommendations" icon-name="utility:lightbulb">
        <template if:true={recommendations.length}>
            <ul class="slds-m-around_medium">
                <template for:each={recommendations} for:item="rec">
                    <li key={rec.id} class="slds-m-bottom_small">
                        <lightning-icon icon-name="utility:check" size="x-small"></lightning-i
                        <span class="slds-m-left_small">{rec.suggestion}</span>
                    </li>
                </template>
            </ul>
        </template>
        <template if:false={recommendations.length}>
            <p class="slds-m-around_medium">No recommendations available.</p>
        </template>
    </lightning-card>
</template>
```


Screenshot 2 — aiRecommendations.js

```javascript
import { LightningElement } from 'lwc';

export default class AiRecommendations extends LightningElement {
    // Sample static recommendations (replace with AI/ML logic later)
    recommendations = [
        { id: 1, suggestion: 'Offer free room upgrade for loyal guest' },
        { id: 2, suggestion: 'Suggest airport pickup for arriving travelers' },
        { id: 3, suggestion: 'Provide dining discount for repeat visitors' }
    ];
}
```
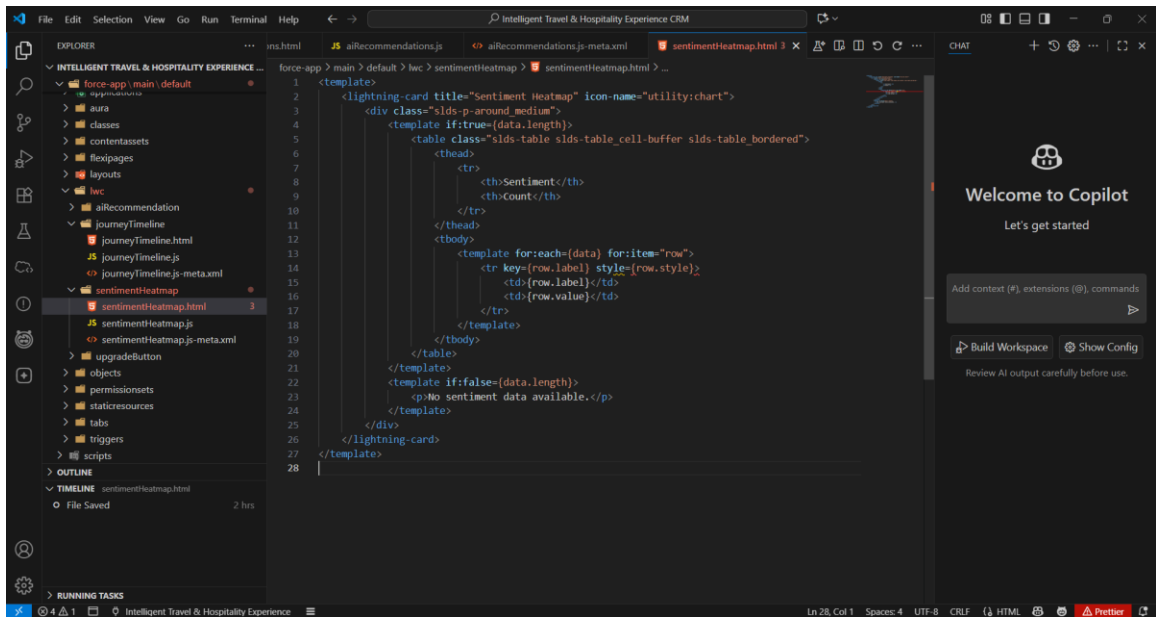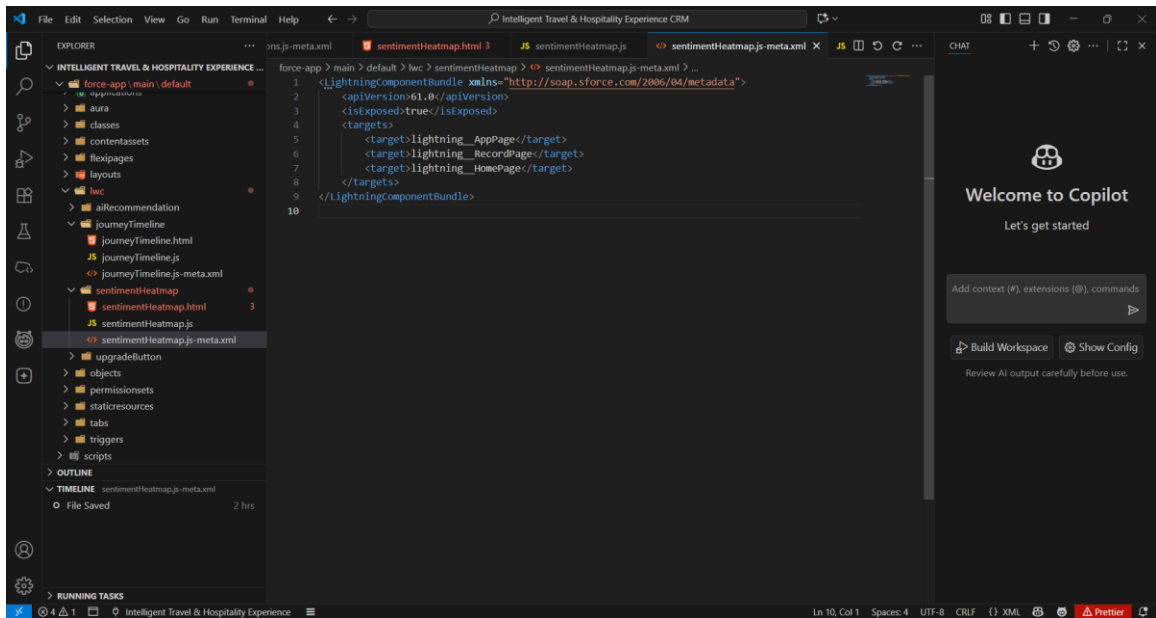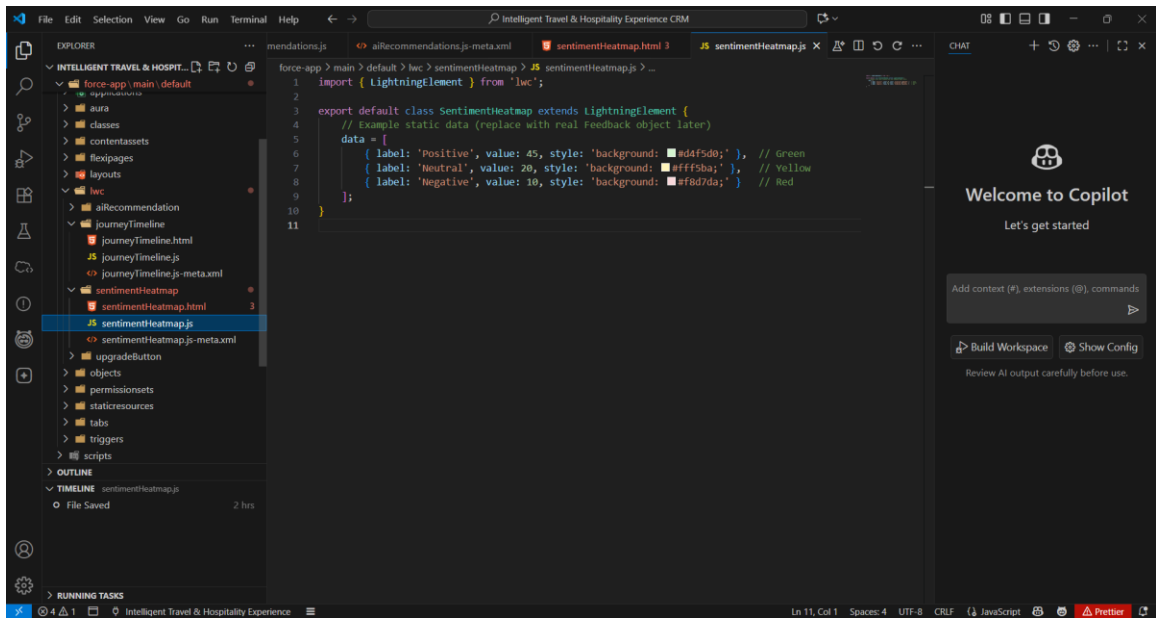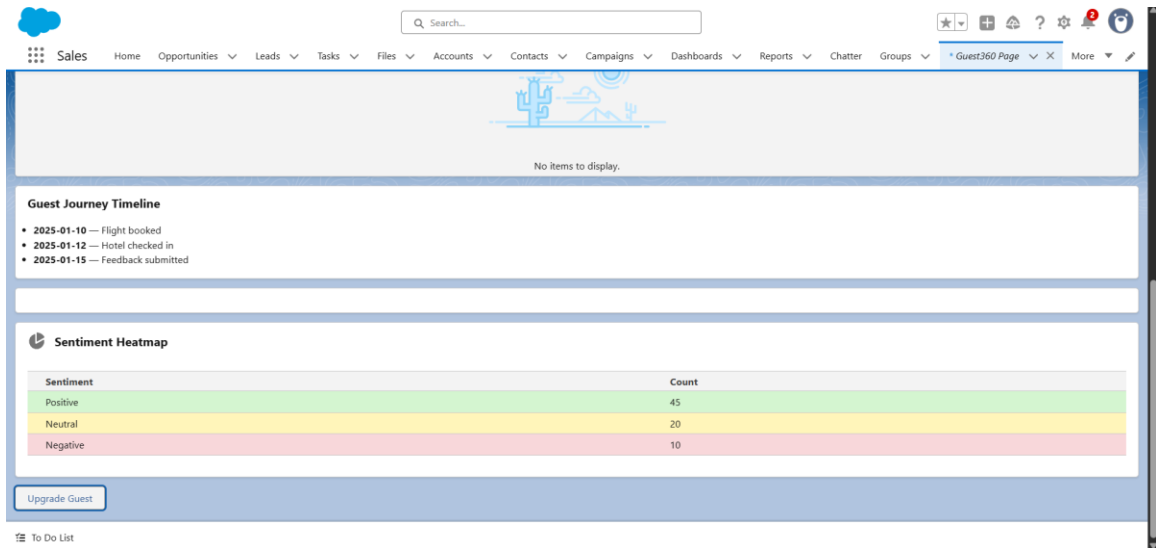
LWC Component – Sentiment Heatmap

Path: **Setup → Developer Console → New Lightning Web Component**

Steps:
1. Create LWC named sentimentHeatmap.
2. Use chart library (e.g., chart.js) to display heatmap by date, location, or interaction.
3. Color-code sentiment values for clarity (positive = green, neutral = yellow, negative = red).
4. Deploy component and add it to Guest360 Page.

Screenshot 1 — sentimentHeatmap.js

```javascript
import { LightningElement } from 'lwc';

export default class SentimentHeatmap extends LightningElement {
    // Example static data (replace with real Feedback object later)
    data = [
        { label: 'Positive', value: 45, style: 'background: #d4f5d0;' }, // Green
        { label: 'Neutral', value: 20, style: 'background: #fff5ba;' },  // Yellow
        { label: 'Negative', value: 10, style: 'background: #f8d7da;' }  // Red
    ];
}
```


Screenshot 2 — sentimentHeatmap.js-meta.xml

```xml
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>61.0</apiVersion>
    <isExposed>true</isExposed>
    <targets>
        <target>lightning__AppPage</target>
        <target>lightning__RecordPage</target>
        <target>lightning__HomePage</target>
    </targets>
</LightningComponentBundle>
```

Step 5: Navigation Service – One-Click Upgrade

Path: **Setup → Lightning App Builder → Guest360 Page**

Create Direct Button for Navigation Service.

That is a upgradeButton.

(The above screenshot is the proof for this step as well.)

Code for upgrade button :-

upgradeButton.html : <template>

   <lightning-button label="Upgrade Guest" onclick={handleClick}></lightning-button>

</template>

upgradeButton.js : import { LightningElement, api } from 'lwc';

import { FlowNavigationNextEvent } from 'lightning/flowSupport';

import { NavigationMixin } from 'lightning/navigation';

export default class UpgradeButton extends NavigationMixin(LightningElement) {

  @api recordId;

  handleClick() {

    // Navigate to Flow with guest ID

    this[NavigationMixin.Navigate]({

      type: 'standard__flow',

      attributes: {

        flowName: 'GuestUpgradeFlow'

      },

      state: {

        GuestId: this.recordId

      }

    });

  }

}

upgradeButton.js-meta.xml : <LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">

```xml
<apiVersion>61.0</apiVersion>

<isExposed>true</isExposed>

<targets>

    <target>lightning__RecordPage</target>

    <target>lightning__AppPage</target>

</targets>

<targetConfigs>

    <targetConfig targets="lightning__RecordPage">

        <property name="recordId" type="String" description="Current Guest Record ID" />

    </targetConfig>

</targetConfigs>

</LightningComponentBundle>
```

Conclusion

Phase 6 delivers a fully functional Guest360 Page with interactive LWCs for journey tracking, AI recommendations, and sentiment visualization. The one-click upgrade button simplifies loyalty management and enhances user experience.