

Phase 6: User Interface Development

1. Lightning App Builder

Theory:

Lightning App Builder allows you to create custom applications by dragging and dropping standard/custom components, LWC, Visualforce, etc.

In your project:



- Job_Application_Tracker.app-meta.xml → defines the custom **Lightning App** for this project.

File Edit Selection View Go Run Terminal Help < > Job Application Tracker-main

EXPLORER

JOB-APPLICATION-TRACKER-MAIN

Job Application Tracker-main

force-app/main/default

applications

Job Application Tracker.app-meta.xml

aura

_eslintrc.json

classes

EventReminder.cls

EventReminder.cls-meta.xml

EventReminderScheduler.cls

EventReminderScheduler.cls-meta.xml

EventReminderScheduler.Test.cls

EventReminderScheduler.Test.cls-meta.xml

EventReminderTest.cls

EventReminderTest.cls-meta.xml

EventTest.cls

EventTest.cls-meta.xml

EventTriggerHandler.cls

EventTriggerHandler.cls-meta.xml

JobApplicationCleaningQueueable.cls

JobApplicationCleaningQueueable.cls-meta.xml

JobApplicationCleaningQueueable.Test.cls

JobApplicationCleaningQueueable.Test.cls...

JobApplicationCleaningQueueable.cls...

JobApplicationCleaningScheduler.cls

JobApplicationCleaningScheduler.cls-m...

JobApplicationCleaningSchedulerTest.cls

JobApplicationCleaningSchedulerTest.cls...

JobApplicationContactTest.cls

JobApplicationContactTest.cls-meta.xml

JobApplicationContactTriggerHandler.cls

OUTLINE

TIMELINE

Company/_c_field-meta.xml

Jobbie_REST_APIExternalCredential.meta.xml

Job_Application_Tracker.app-meta.xml

1 <?xml version="1.0" encoding="UTF-8"?>

2 <CustomApplication xmlns="http://soap.sforce.com/2006/04/metadata">

3 <actionOverrides>

4 <actionName>View</actionName>

5 <comment>Action override created by Lightning App Builder during activation.</comment>

6 <content>Job_Application_Record_Page</content>

7 <formFactor>Large</formFactor>

8 <skipRecordTypeSelect>false</skipRecordTypeSelect>

9 <type>Flexipage</type>

10 <pageOrSubjectType>Job_Application_c</pageOrSubjectType>

11 </actionOverrides>

12 <actionOverrides>

13 <actionName>View</actionName>

14 <comment>Action override created by Lightning App Builder during activation.</comment>

15 <content>Job_Application_Record_Page</content>

16 <formFactor>Small</formFactor>

17 <skipRecordTypeSelect>false</skipRecordTypeSelect>

18 <type>Flexipage</type>

19 <pageOrSubjectType>Job_Application_c</pageOrSubjectType>

20 </actionOverrides>

21 <actionOverrides>

22 <actionName>View</actionName>

23 <comment>Action override created by Lightning App Builder during activation.</comment>

24 <content>Job_Application_contact_Record_Page</content>

25 <formFactor>Large</formFactor>

26 <skipRecordTypeSelect>false</skipRecordTypeSelect>

27 <type>Flexipage</type>

28 <pageOrSubjectType>Job_Application_Contact_c</pageOrSubjectType>

29 </actionOverrides>

30 <actionOverrides>

31 <actionName>View</actionName>

32 <comment>Action override created by Lightning App Builder during activation.</comment>

33 <content>Job_Application_Contact_Record_Page</content>

34 <formFactor>Small</formFactor>

35 <skipRecordTypeSelect>false</skipRecordTypeSelect>

36 <type>Flexipage</type>

37 <pageOrSubjectType>Job_Application_Contact_c</pageOrSubjectType>

CHAT

Welcome to Copilot

Let's get started

Add context (#), extensions (@), commands

Build Workspace Show Config

Review AI output carefully before use.

In 1 Col 1 Spaces: 4 UTF-8 LF {} XML Finish Setup

28°C Mostly cloudy

```

<CustomApplication xmlns="http://soap.sforce.com/2006/04/metadata">
    <actionOverrides>
        <recordTypeSelect>true</recordTypeSelect>
        <skipRecordTypeSelect>false</skipRecordTypeSelect>
        <pageOrObjectType>standard-home</pageOrObjectType>
    </actionOverrides>
    <brand>
        <headerColor>#007002</headerColor>
        <shouldOverrideOrgTheme>false</shouldOverrideOrgTheme>
    </brand>
    <description>App for users to track job applications</description>
    <formFactors>small;/formFactors>
    <isNavAutoTempTabsDisabled>false</isNavAutoTempTabsDisabled>
    <isNavPersonalizationDisabled>false</isNavPersonalizationDisabled>
    <isNavAbPersistenceDisabled>false</isNavAbPersistenceDisabled>
    <isOmniPinEnabled>false</isOmniPinEnabled>
    <label>Job Application Tracker</label>
    <navType>standard;/navType>
    <tabs>standard-home</tabs>
    <tabs>Job Application</tabs>
    <tabs>standard-Event</tabs>
    <tabs>standard-Task</tabs>
    <tabs>standard-Account</tabs>
    <tabs>standard-Contact</tabs>
    <tabs>standard-DeletedEvent</tabs>
    <uiType>Lightning</uiType>
    <utilityBar>Job_Application_Tracker_Utilities</utilityBar>
</CustomApplication>

```

2. Record Pages

Theory:

Record Pages customize how records appear in Lightning Experience. They can include fields, related lists, and components.

In your project:

force-app/main/default/flexipages

- Account_Record_Page.flexipage-meta.xml
- Contact_Record_Page.flexipage-meta.xml
- Event_Record_Page.flexipage-meta.xml
- Job_Application_Record_Page.flexipage-meta.xml
- Job_Application_Contact_Record_Page.flexipage-meta.xml

Account_Record_Page.flexipage-meta.xml:

The screenshot shows the Microsoft Copilot extension integrated into the Visual Studio Code interface. The left sidebar displays a file tree with various XML files related to a 'JOB-APPLICATION-TRACKER-MAIN' project. The main editor area shows a large XML document with numerous changes highlighted by blue arrows pointing from the original text to modified or new content. A vertical scrollbar on the right indicates the document is very long. The status bar at the bottom shows 'Line 1, Col 1' and other standard VS Code metrics. On the right side, there's a 'CHAT' panel with a 'Welcome to Copilot' message and a 'Let's get started' button, along with buttons for 'Build Workspace' and 'Show Config'. A 'Rain warning' icon is visible in the bottom-left corner.

Contact_Record_Page.flexipage-meta.xml:

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "JOB-APPLICATION-TRACKER-MAIN".
- Editor:** Displays the XML content of "Contact_Record_Page.flexipage-meta.xml".
- Copilot Panel:** Located on the right, it says "Welcome to Copilot" and "Let's get started". It includes buttons for "Add context (#)", "extensions (@)", "commands", "Build Workspace", and "Show Config".
- Status Bar:** Shows "1n 1, Col 1", "Spaces: 4", "UTF-8", "LF", "XML", "Finish Setup", and system icons.

3. Tabs

Theory:

Custom Tabs provide quick access to custom objects, LWC, or Visualforce pages.

In your project:



- Job_Application__c.tab-meta.xml
- Job_Application_Contact__c.tab-meta.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<CustomTab xmlns="http://soap.sforce.com/2006/04/metadata">
    <customObject>true</customObject>
    <description>A custom tab for the Job Application custom object</description>
    <motif>Custom: Diamond</motif>
</CustomTab>
```

4. Home Page Layouts

Theory:

Home Page Layouts define the main landing page for users, showing dashboards, reports, or custom components.

In your project:



- Job_Application_Tracker_Home_Page.flexipage-meta.xml

5. Utility Bar

Theory:

Utility Bar is a persistent bar at the bottom of Lightning Apps to provide tools like notes, history, or custom components.

In your project:



- Job_Application_Tracker.UtilityBar.flexipage-meta.xml

6. LWC (Lightning Web Components)

Theory:

LWCs are modern, lightweight components built with HTML, JavaScript, and metadata files to run natively on Salesforce.

In your project:



- calculateTakeHome/ → salary/tax calculation UI.
- errorPanel/ → reusable error display component.
- jobSearch/ → search for jobs.
- jobSearchResult/ → display job search results.
- ldsUtils/ → utility functions.

Each contains .html, .js, .js-meta.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightningComponentBundle xmlns="http://soap.sforce.com/2006/04/metadata">
    <apiVersion>59.0</apiVersion>
    <isExposed>true</isExposed>
    <masterLabel>Take Home Salary</masterLabel>
    <targets>
        <target>lightning_AppPage</target>
        <target>lightning_RecordPage</target>
        <target>lightning_UtilityBar</target>
    </targets>
</LightningComponentBundle>
```

Job Application Tracker Main

```

File Edit Selection View Go Run Terminal Help < > Job-Application-Tracker-main
EXPLORER Job_Application_c.tab-meta.xml calculateTakeHome.html calculateTakeHome.js-meta.xml errorPanel.test.js ...
JOB-APPLICATION-TRACKER-M... Job-Application-Tracker-main force-app\main\default layouts
  Event-Event Layout.layout-meta.xml
  Job_c-Job Layout.layout-meta.xml
  Job_Application_c-Job Application Lay...
  Job_Application_Contact_c-Job Appli...
  Task-Task Layout.layout-meta.xml
  lwc
    calculateTakeHome
      _tests_
        calculateTakeHome.html
        calculateTakeHome.js
        calculateTakeHome.js-meta.xml
    errorPanel
      _tests_
        errorPanel.test.js
      templates
        errorPanelLhtml
        errorPaneljs
        errorPaneljs-meta.xml
    jobSearch
      _tests_
        jobSearch.test.js
        jobSearch.html
        jobSearch.js
        jobSearch.js-meta.xml
    jobSearchResult
      _tests_
        # jobSearchResult.css
  < OUTLINE > TIMELINE
  0 0 0 Air: Satisfactory Tomorrow
  Search
  Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Finish Setup
  CHAT + ⚙️ ... | ☰
  Welcome to Copilot
  Let's get started
  Add context (#), extensions (@), commands
  Build Workspace Show Config
  Review AI output carefully before use.
  18:37 25-09-2025

```

Job Application Tracker Main

```

File Edit Selection View Go Run Terminal Help < > Job-Application-Tracker-main
EXPLORER netा.xml calculateTakeHome.html calculateTakeHome.js-meta.xml errorPanel.test.js jobSearch.test.js ...
JOB-APPLICATION-TRACKER-M... Job-Application-Tracker-main force-app\main\default layouts
  Event-Event Layout.layout-meta.xml
  Job_c-Job Layout.layout-meta.xml
  Job_Application_c-Job Application Lay...
  Job_Application_Contact_c-Job Appli...
  Task-Task Layout.layout-meta.xml
  lwc
    calculateTakeHome
      _tests_
        calculateTakeHome.html
        calculateTakeHome.js
        calculateTakeHome.js-meta.xml
    errorPanel
      _tests_
        errorPanel.test.js
      templates
        errorPanelLhtml
        errorPaneljs
        errorPaneljs-meta.xml
    jobSearch
      _tests_
        jobSearch.test.js
        jobSearch.html
        jobSearch.js
        jobSearch.js-meta.xml
    jobSearchResult
      _tests_
        # jobSearchResult.css
  < OUTLINE > TIMELINE
  0 0 0 Air: Satisfactory Tomorrow
  Search
  Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript Finish Setup
  CHAT + ⚙️ ... | ☰
  Welcome to Copilot
  Let's get started
  Add context (#), extensions (@), commands
  Build Workspace Show Config
  Review AI output carefully before use.
  18:37 25-09-2025

```

7. Apex with LWC

Theory:

LWCs can interact with Apex classes for server-side logic using `@AuraEnabled` methods.

In your project:

- `JobApplicationController.cls` → called by LWCs for job application handling.
- `JobController.cls` → interacts with job objects.
- `TaxCalculation.cls` → used by `calculateTakeHome` LWC.

8. Events in LWC

Theory:

Events are used for communication between LWCs (child → parent, or sibling → via pub-sub). They're fired in JS (new `CustomEvent`) and handled in parent components.

In your project:

- `jobSearch.js` (fires events when user searches).
- `jobSearchResult.js` (listens and displays search results).

9. Wire Adapters

Theory:

Wire adapters fetch Salesforce data declaratively into LWCs using the `@wire` decorator. Example: `@wire(getRecord, { recordId: ..., fields: [...] })`.

In your project:

- ldsUtils.js → contains wire service utilities for data access.
- jobSearch.js → may use wire adapters for fetching job data.

10. Imperative Apex Calls

Theory:

Instead of reactive wire service, you can call Apex imperatively (on button click, form submit).

Syntax:

```
import getJobs from '@salesforce/apex/JobController.getJobs';  
getJobs({ param1: 'value' }).then(...).catch(...);
```

In your project:

- jobSearch.js → calls JobController imperatively to fetch job search results.
- calculateTakeHome.js → calls TaxCalculation.cls for salary calculations.

Summary of Phase 6 Files

- **Lightning App** → applications/Job_Application_Tracker.app-meta.xml
- **Record Pages** → flexipages/*Record_Page.flexipage-meta.xml
- **Tabs** → tabs/Job_Application__c.tab-meta.xml,
tabs/Job_Application_Contact__c.tab-meta.xml
- **Home Page** → flexipages/Job_Application_Tracker_Home_Page.flexipage-meta.xml
- **Utility Bar** → flexipages/Job_Application_Tracker_UtilityBar.flexipage-meta.xml
- **LWCs** → lwc/* (jobSearch, jobSearchResult, calculateTakeHome, etc.)
- **Apex with LWC** → JobApplicationController.cls, JobController.cls,
TaxCalculation.cls
- **Events in LWC** → jobSearch.js, jobSearchResult.js
- **Wire Adapters** → ldsUtils.js, jobSearch.js

- **Imperative Apex Calls** → jobSearch.js, calculateTakeHome.js
- **Navigation Service** → jobSearchResult.js