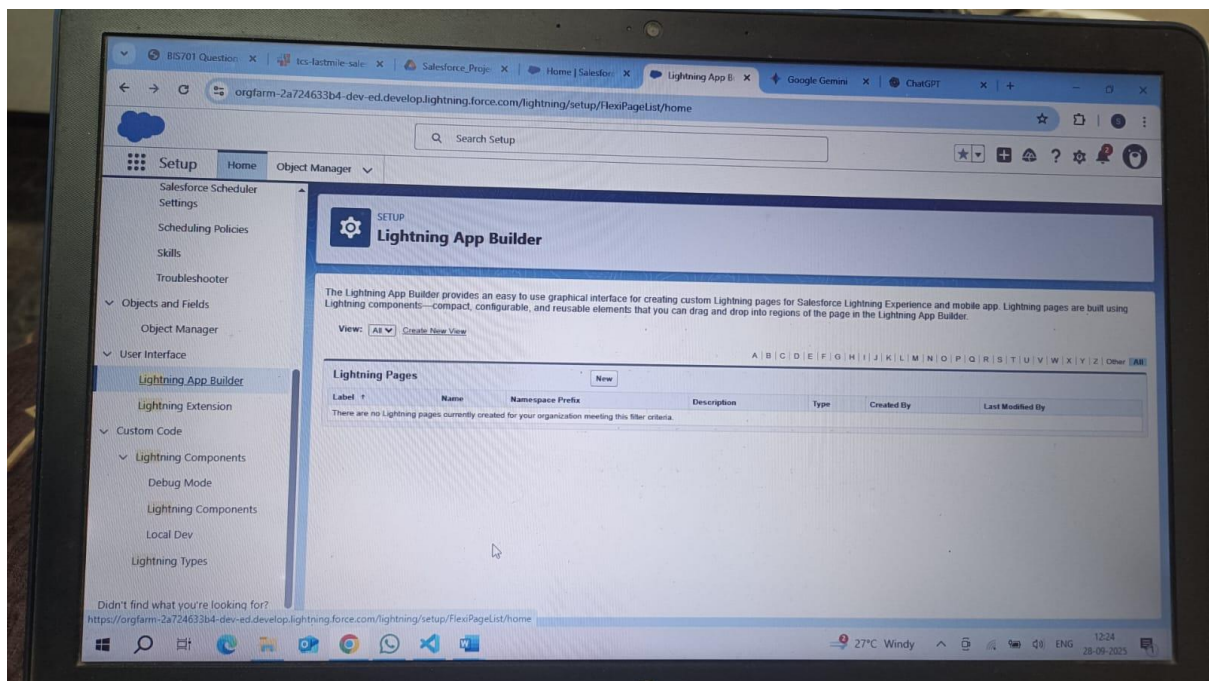


# Phase 6: User Interface Development – AI Career Coach with Salesforce Integration

## 1. Lightning App Builder

- Use Lightning App Builder to create a **custom Salesforce app** for the AI Career Coach.
- **Components to include:**
  - AI Chatbot component (LWC) for student interaction.
  - Student profile summary (record component showing fields like courses enrolled, progress, skills).
  - Recommended courses and career paths based on AI analysis.

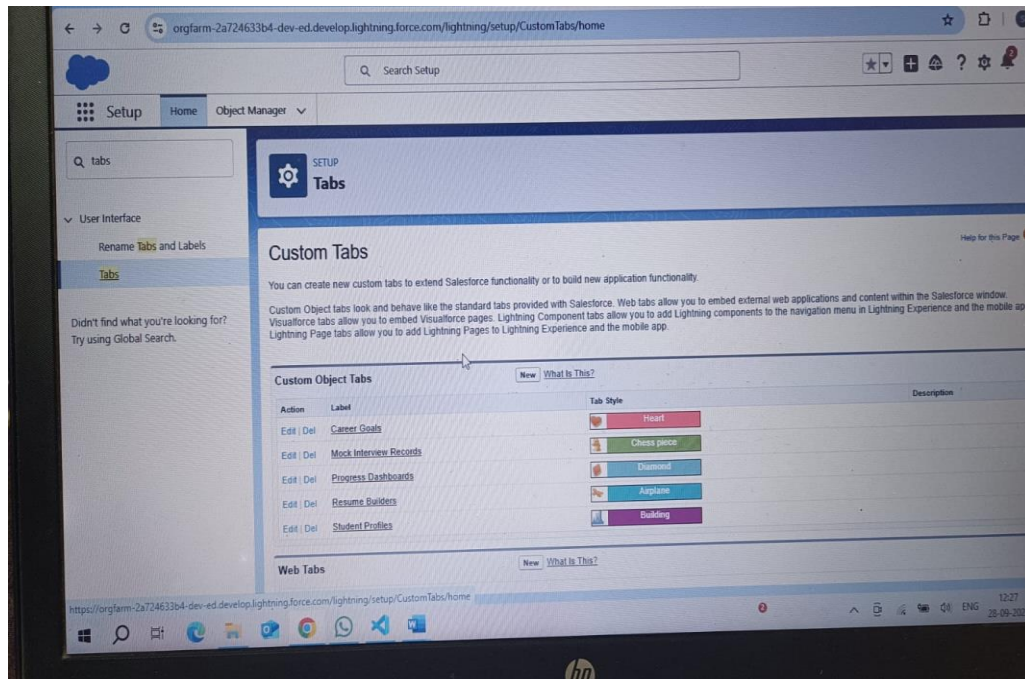


## 2. Record Pages

- Customize **record pages** for key objects:
  - **Student:** Display AI recommendations, progress charts, and interaction history.
  - **Course:** Show course details, enrollment status, and AI suggestions for next modules.
  - **Assessment:** Display quiz scores, feedback, and improvement suggestions.

### 3. Tabs

- Create tabs in the Salesforce app for easier navigation:
  - **Home:** AI Dashboard (summary of activities, alerts).
  - **Courses:** List of available courses.
  - **Career Guidance:** AI-generated recommendations.
  - **Assessments:** Quizzes and mock tests.
  - **Chatbot:** AI interaction component.



---

### 4. Home Page Layouts

- Design a **home page** for logged-in users:
  - **Top panel:** Student progress summary and notifications.
  - **Middle panel:** Recommended courses or skills to learn.
  - **Bottom panel:** AI suggestions and recent activity logs.

---

### 5. Utility Bar

- Add a **utility bar** in the app for quick access:
  - AI Chatbot (always available for queries).
  - Notifications (upcoming assessments, deadlines).
  - Help & Support (FAQs, contact).

---

## 6. Lightning Web Components (LWC)

- Key LWCs for your project:
    - **AIChatbotLWC**: Real-time conversation interface.
    - **ProgressTrackerLWC**: Visual charts of student progress.
    - **CourseRecommendationLWC**: Dynamic suggestions based on AI model.
    - **AssessmentResultLWC**: Display results and recommendations.
- 

## 7. Apex with LWC

- Use Apex controllers to fetch and update Salesforce data for LWCs:
  - Retrieve student profile, completed courses, and assessment data.
  - Send AI recommendation results to Salesforce records.
  - Update progress after quizzes automatically.

### Example:

```
@AuraEnabled(cacheable=true)
public static List<Course__c>
getRecommendedCourses(Id studentId) {
    // Logic to fetch recommended courses based on AI
    model results
}
```

---

## 8. Events in LWC

- **Custom events** to communicate between components:
    - `courseSelected` → Updates the progress tracker.
    - `assessmentCompleted` → Triggers AI recommendation update.
    - `chatMessageSent` → Logs interaction in Salesforce.
- 

## 9. Wire Adapters

- Use `@wire` to fetch Salesforce data reactively:

```
import { LightningElement, wire } from 'lwc';
import getStudentProfile from
 '@salesforce/apex/StudentController.getStudentProfile
';

export default class StudentProfile extends
 LightningElement {
    @wire(getStudentProfile, { studentId: '$recordId'
    })
    student;
}
```

- Example usage: Dynamically show student info or AI recommendations as soon as data changes.

---

## 10. Imperative Apex Calls

- Use for actions triggered by user interaction (not automatic updates):

```
import getRecommendations from
 '@salesforce/apex/AIController.getRecommendations';

handleGetRecommendations() {
    getRecommendations({ studentId: this.studentId })
        .then(result => { this.recommendations =
result; })
        .catch(error => { console.error(error); });
}
```

- Example usage: Fetch AI recommendations when the user clicks a button.

---

## 11. Navigation Service

- Programmatically navigate between pages:

```
import { NavigationMixin } from
 'lightning/navigation';

export default class NavigateToCourse extends
 NavigationMixin(LightningElement) {
    navigateToCoursePage(courseId) {
```

```
    this[NavigationMixin.Navigate]({
      type: 'standard__recordPage',
      attributes: {
        recordId: courseId,
        objectApiName: 'Course__c',
        actionName: 'view'
      }
    });
  }
}
```

- Use case: Clicking on a recommended course navigates to its Salesforce record page.