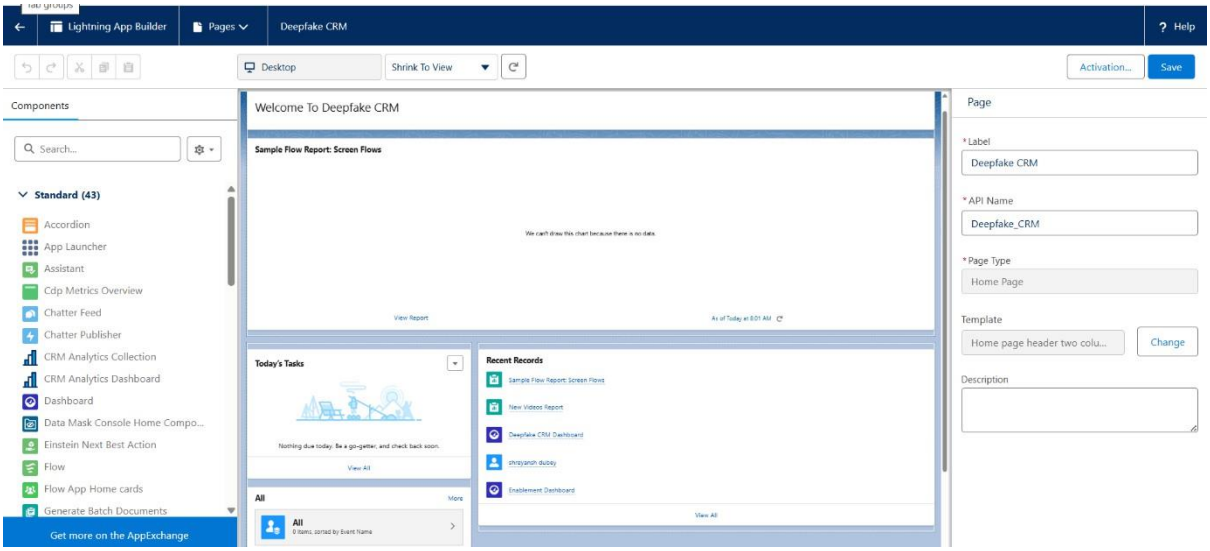
Phase 6: User Interface Development - Salesforce

*Project: **Deepfake CRM: Intelligent Salesforce Integration Project***

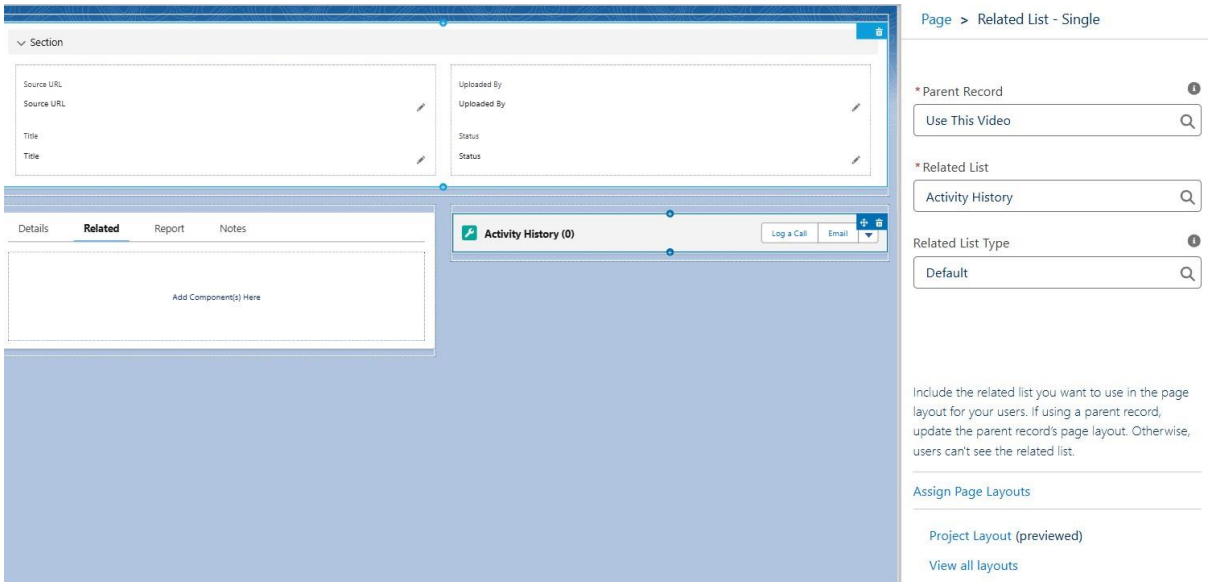
1. Lightning App Builder

- Drag-and-drop tool to create **App, Home, or Record pages**.
- Types of pages:
  - **App Page:** Standalone page in App Launcher.
  - **Home Page:** Landing page with dashboards, reports, and components.
  - **Record Page:** Custom layout for specific objects (e.g., Video).
- Features:
  - Dynamic visibility
  - Tabs for organization
  - Custom LWC integration
- **Tip:** Use tabs template for Record Pages to organize Details, Related Lists, and custom components.



2. Record Pages

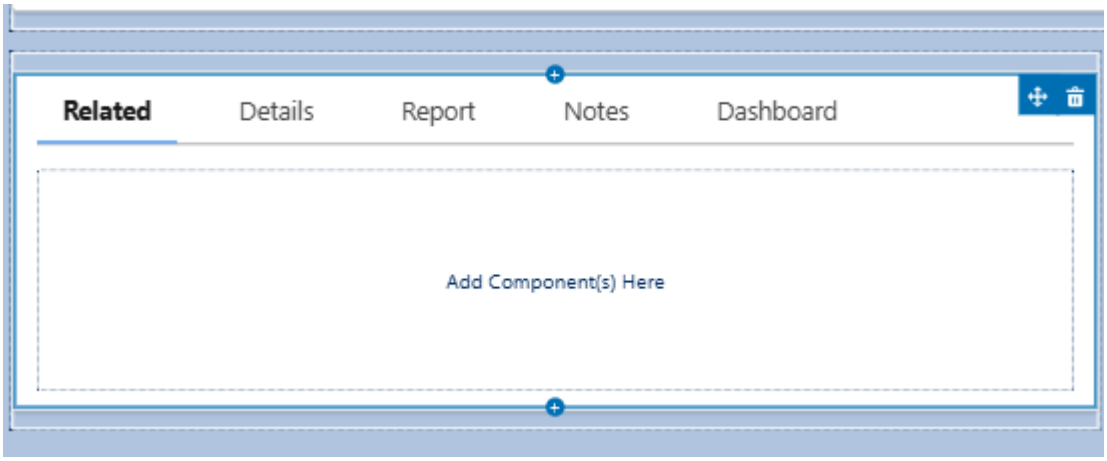
- Customizes layout for **object records**.
- Include:
    - **Highlights Panel:** Key fields at top.
    - **Tabs:** Organize Details, Related Lists, and Custom LWCs.
    - **Custom Components:** Video Player, Alerts, Detection status.
- Assign pages to **profiles or apps**.



---

3. Tabs

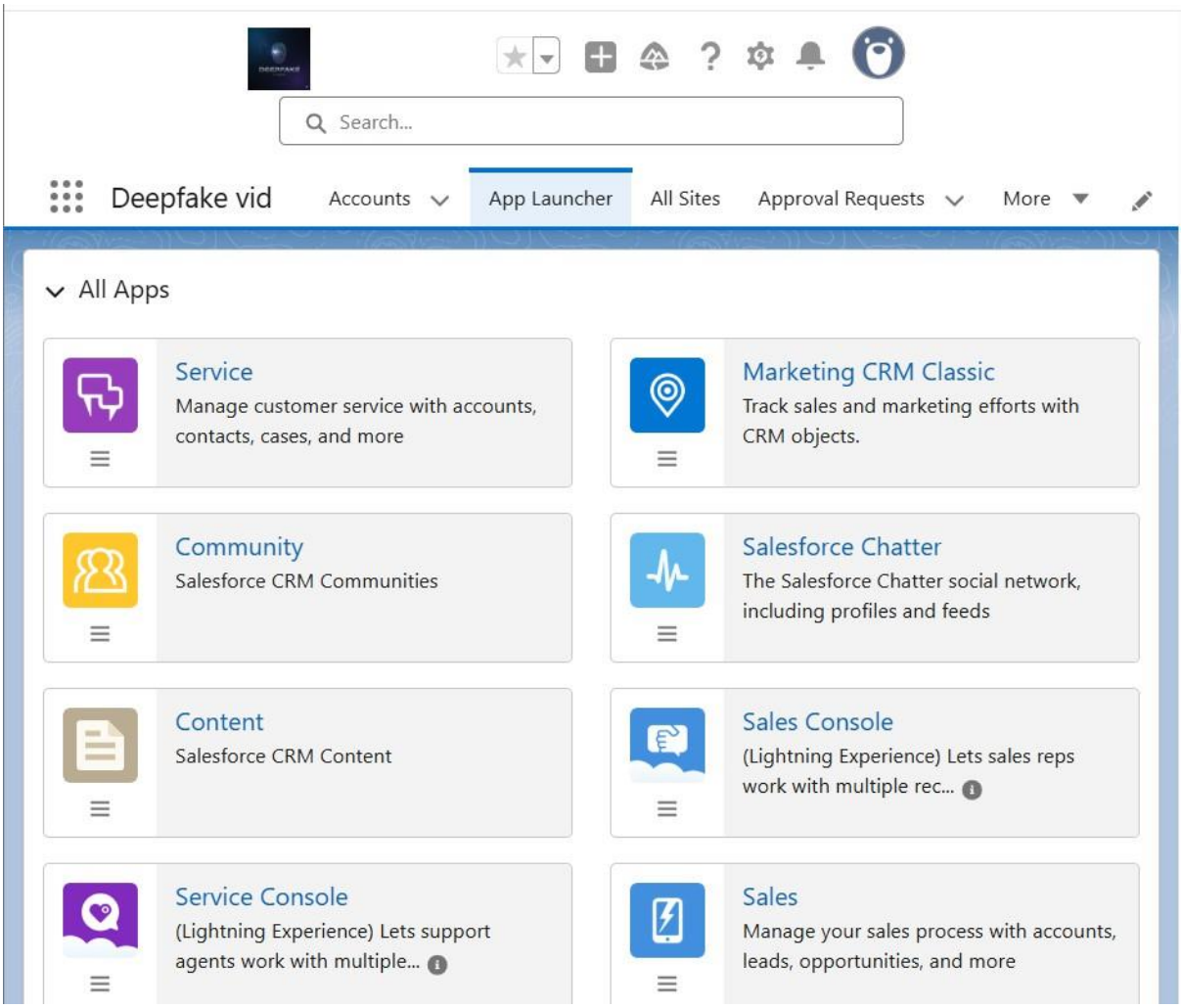
- Organize information efficiently.
- Examples for Video object:
    - Details (Record Fields)
    - Related (Alerts, Comments)
    - Video Player (LWC)
    - Detection/Info (Custom LWC or Rich Text)
    - Report
    - Notes
    - Dashboard



---

4. Home Page Layouts

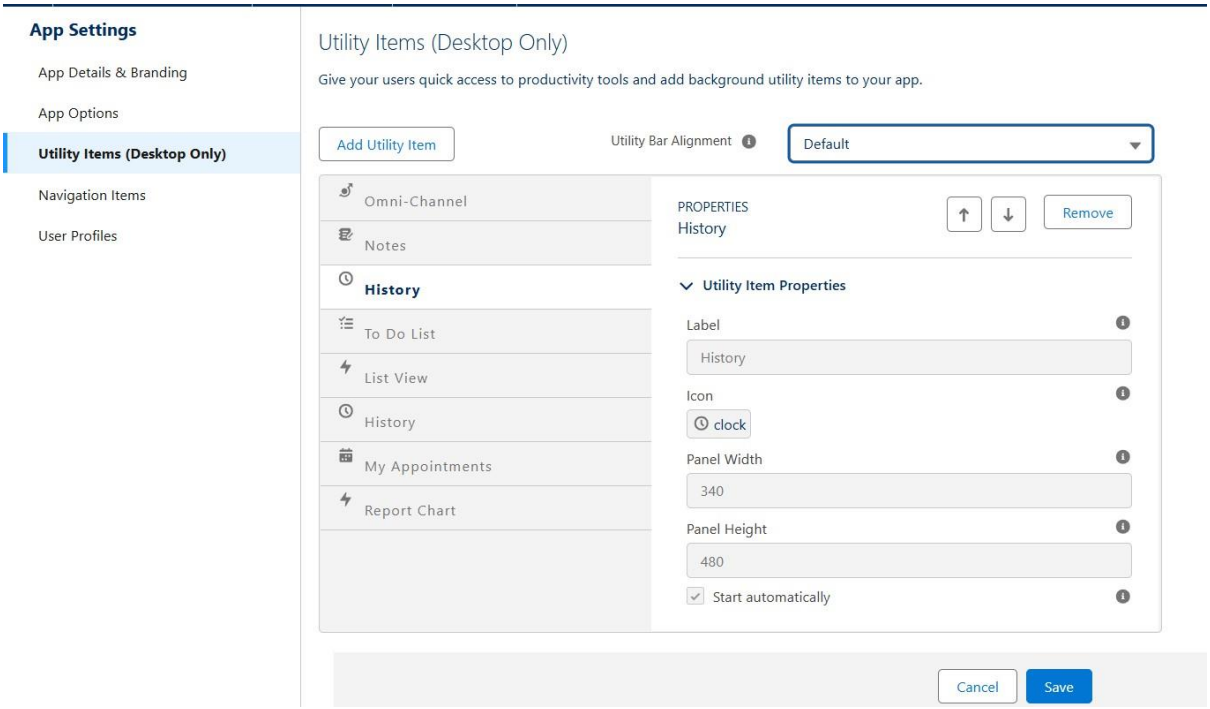
- Custom landing page for users.
- Components:
  - Reports, Dashboards
  - Announcements
  - Custom LWC (e.g., Recent Videos)
- Can assign different layouts for different **profiles**.



---

5. Utility Bar

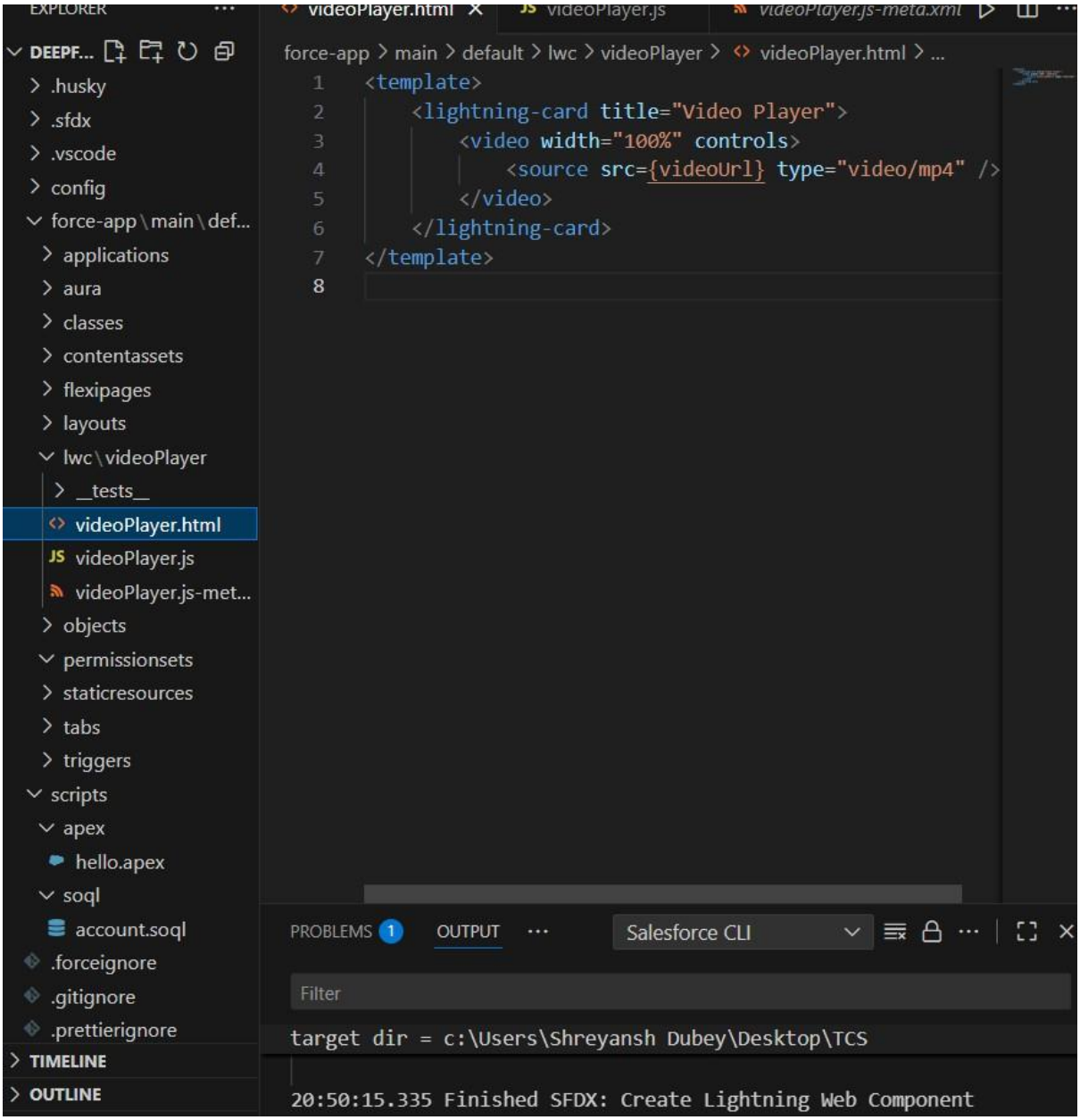
- Persistent footer in Lightning apps.
- Quick access tools for productivity.
- Examples for Video project:
    - Video Uploader (LWC)
    - Recent Videos
    - Alerts from Detection System
- Add via **App Manager** → **Utility Bar** → **Add Utility**.

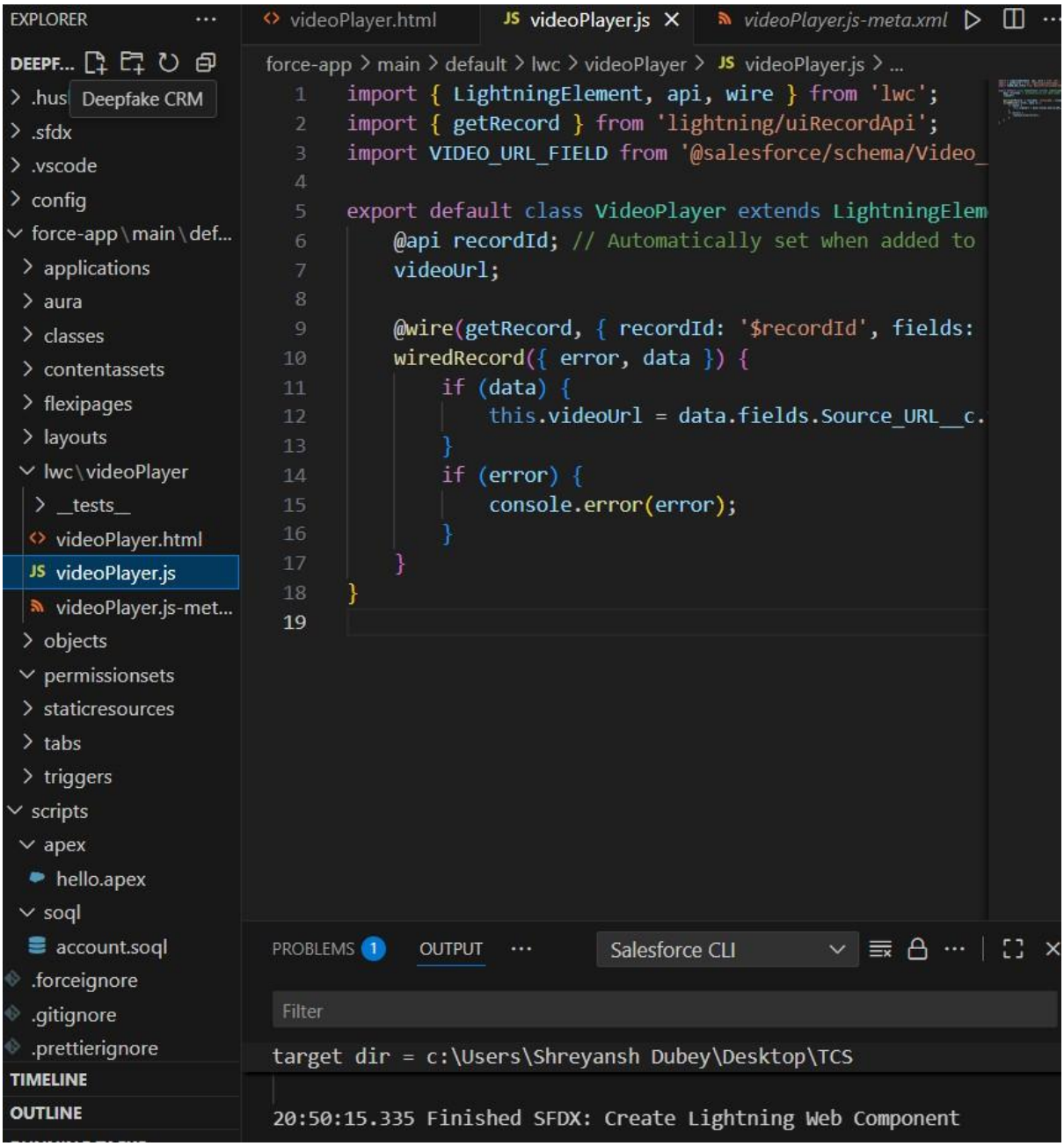


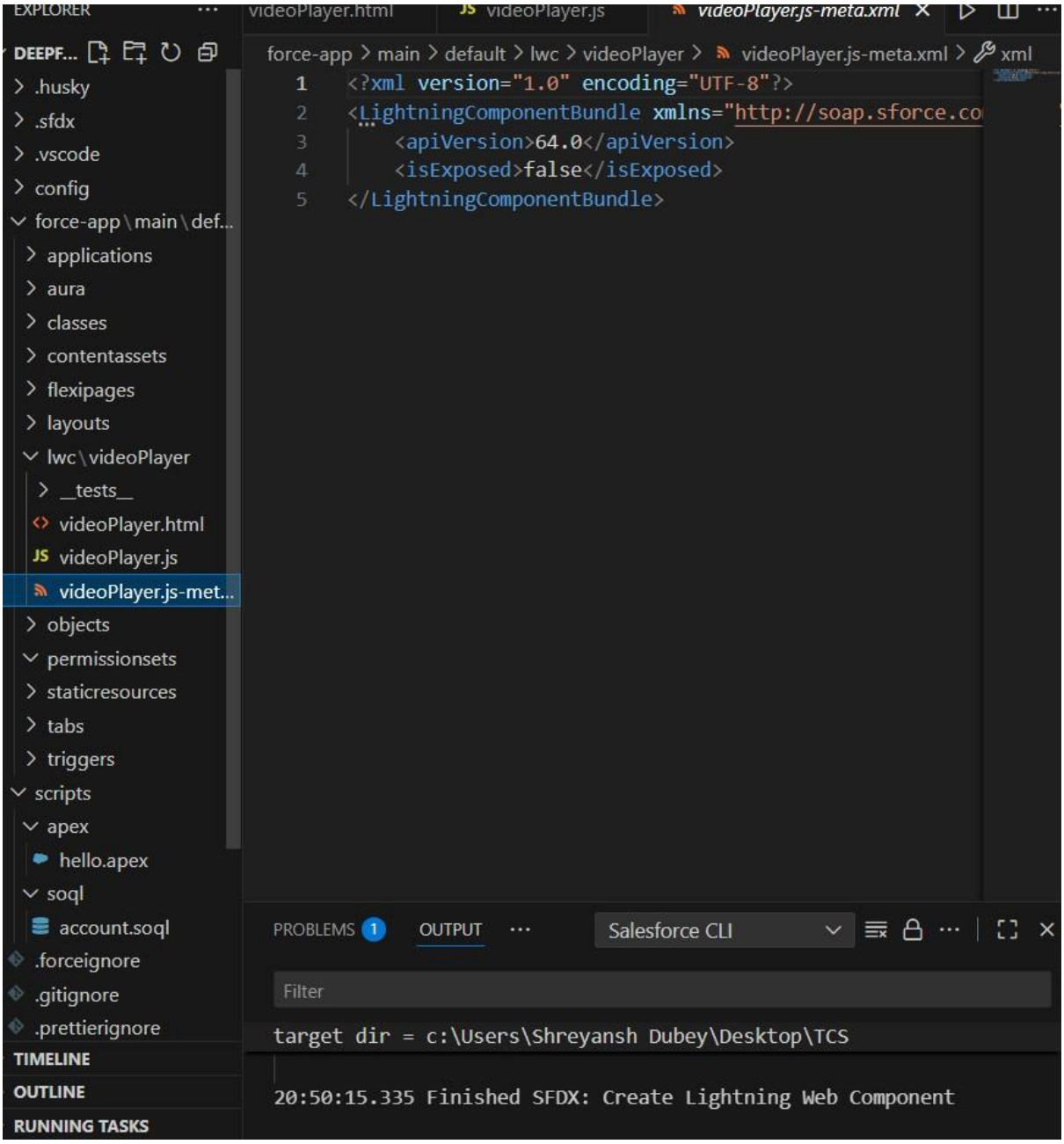
---

6. Lightning Web Components (LWC)

- Modern framework based on **JavaScript ES6+**.
- Files:
  - .html → UI
  - .js → Logic
  - .css → Styling (optional)
- Can be added to Record Pages, App Pages, and Utility Bar.
- Example: Video Player, Video Alerts, Upload Widget.


EXPLORER
videoPlayer.html
videoPlayer.js
videoPlayer.js-meta.xml
DEEPF...
force-app > main > default > lwc > videoPlayer > videoPlayer.html > ...
.husky
.sfdx
.vscode
config
force-app\main\def...
applications
aura
classes
contentassets
flexipages
layouts
lwc\videoPlayer
__tests__
videoPlayer.html
videoPlayer.js
videoPlayer.js-met...
objects
permissionsets
staticresources
tabs
triggers
scripts
apex
hello.apex
soql
account.soql
.forceignore
.gitignore
.prettierignore
TIMELINE
OUTLINE
<template>
<lightning-card title="Video Player">
<video width="100%" controls>
<source src={videoUrl} type="video/mp4" />
</video>
</lightning-card>
</template>
PROBLEMS 1
OUTPUT
Salesforce CLI
Filter
target dir = c:\Users\Shreyansh Dubey\Desktop\TCS
20:50:15.335 Finished SFDX: Create Lightning Web Component


EXPLORER
DEEPF...
.hus
Deepfake CRM
.sfdx
.vscode
config
force-app \ main \ def...
applications
aura
classes
contentassets
flexipages
layouts
lwc \ videoPlayer
__tests__
videoPlayer.html
videoPlayer.js
videoPlayer.js-met...
objects
permissionsets
staticresources
tabs
triggers
scripts
apex
hello.apex
soql
account.soql
.forceignore
.gitignore
.prettierignore
TIMELINE
OUTLINE
videoPlayer.html
videoPlayer.js
videoPlayer.js-meta.xml
force-app > main > default > lwc > videoPlayer > videoPlayer.js > ...
1 import { LightningElement, api, wire } from 'lwc';
2 import { getRecord } from 'lightning/uiRecordApi';
3 import VIDEO_URL_FIELD from '@salesforce/schema/Video_
4
5 export default class VideoPlayer extends LightningElem
6 @api recordId; // Automatically set when added to
7 videoUrl;
8
9 @wire(getRecord, { recordId: '$recordId', fields:
10 wiredRecord({ error, data }) {
11 if (data) {
12 this.videoUrl = data.fields.Source_URL__c.
13 }
14 if (error) {
15 console.error(error);
16 }
17 }
18 }
19
PROBLEMS 1
OUTPUT
Salesforce CLI
Filter
target dir = c:\Users\Shreyansh Dubey\Desktop\TCS
20:50:15.335 Finished SFDX: Create Lightning Web Component



---

7. Apex with LWC

- Apex handles **server-side logic**.
- **Read-only data:** Use **@AuraEnabled(cacheable=true)** with wire adapter.
- **User actions:** Use **imperative Apex calls** for buttons/events.
- Example:
    - Fetch Video details

    - Update Video status
    - Analyze video

---

8. Events in LWC

- **Custom events:** Child → Parent communication.
- **Pub/Sub / Application events:** Component → unrelated component communication.
- Use case: Update Detection Status across multiple components when video is analyzed.

G. Wire Adapters

- Reactive data fetching.
- Examples:
    - getRecord
    - getObjectInfo
    - getPicklistValues
- Automatically updates component when data changes.

---

10. Imperative Apex Calls

- Explicit Apex call triggered by user action.
- Example:
    - Analyze video
    - Approve video
- Flexible control over when server-side code runs.

---

11. Navigation Service

- Use NavigationMixin in LWC.
- Programmatic navigation:
    - Standard pages (record, object list)

    - URLs
    - Lightning pages
- Example: Redirect user after saving or analyzing a video.
