Phase 5: Apex Programming (Developer)

Custom Apex controllers were developed to fetch aggregated data for dashboards. This included counts of Jobs, Applications by stage, and Candidates.

Apex Classes Developed:

1. **Class Name:** `UserUtil`
   What it does: Returns the current user's Id, Name, and Role/Profile as a map for use in Lightning components.
   Methods used: `@AuraEnabled`, `UserInfo.getUserId()`, SOQL query on `User`, ternary operator, `Map.put()`.

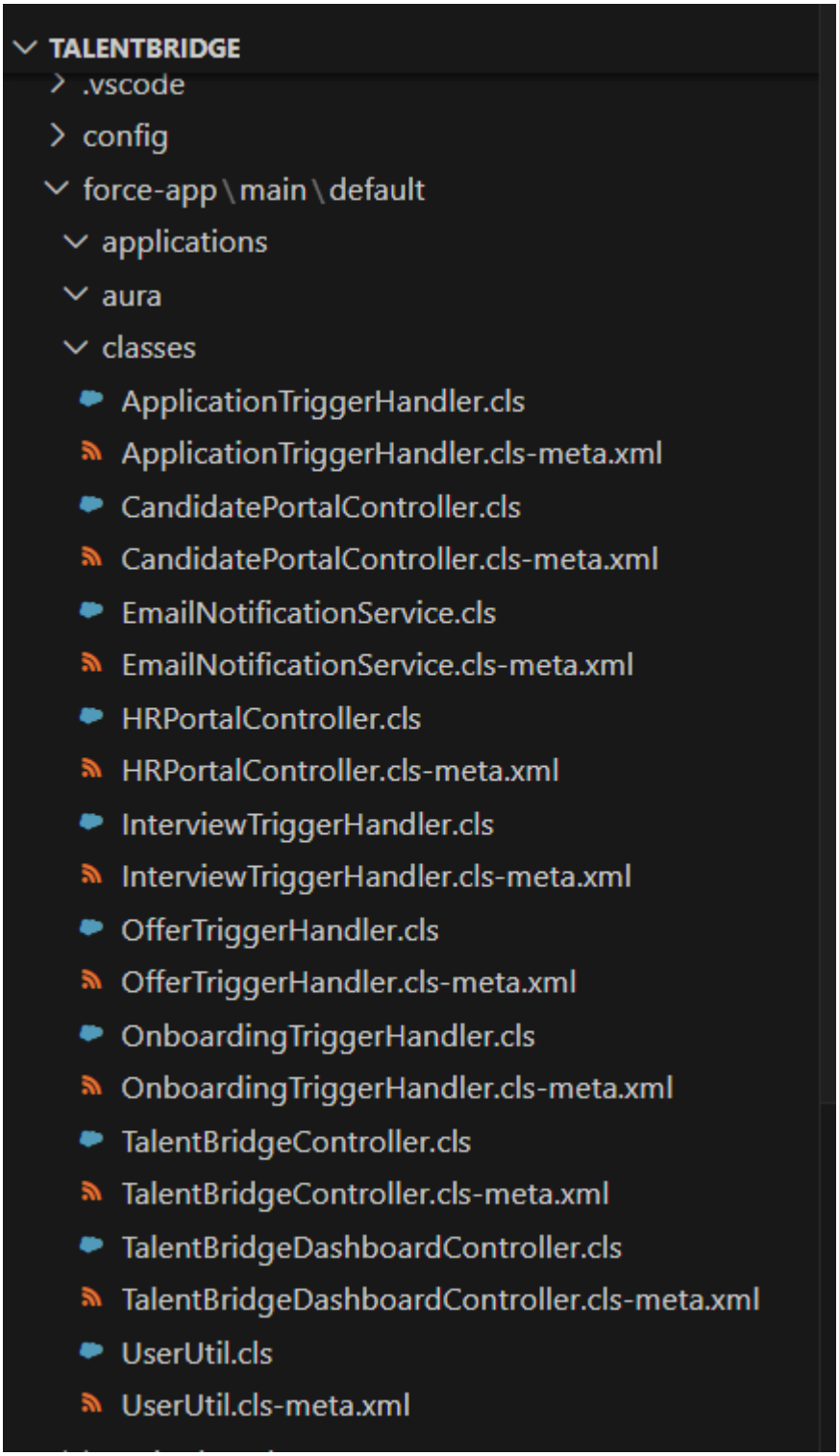
2. **Class Name:** TalentBridgeController
   What it does: Handles fetching Jobs, Candidates, and Applications for the Candidate Portal based on search input.
   Methods used: @AuraEnabled, String.escapeSingleQuotes(), SOQL queries with LIKE, ORDER BY, and LIMIT.

3. **Class Name:** TalentBridgeDashboardController
   What it does: Generates and returns dashboard analytics, including total counts and daily trends for Jobs, Applications, and Candidates.
   Methods used: @AuraEnabled, SOQL COUNT() and AggregateResult with GROUP BY, map and list operations to structure the results.

4. **Class Name:** `CandidatePortalController`
   What it does: Manages candidate-facing operations—fetches picklist values, lists active jobs, retrieves job details, creates/updates candidates, allows candidates to apply for jobs, and retrieves candidate applications.
   Methods used:
   `@AuraEnabled(cacheable=true)`/`@AuraEnabled`, `Schema.DescribeFieldResult` and `getPicklistValues()`, SOQL queries on `Candidate__c`, `Jobs_c__c`, `Application__c`, `insert` and `update` DML operations, exception handling with `AuraHandledException`.

5. **Class Name:** `HRPortalController` — Manages HR Portal: create and list Jobs, fetch/update Applications, schedule and list Interviews, send Offers, manage Candidates/HR users, and create/list Onboarding tasks (exposed to Aura/LWC).
   Methods/Calls & Relations: Uses `@AuraEnabled` methods (createJob, getPostedJobs, getApplicationsForJob, updateApplicationStatus, getScheduledInterviews, getPendingOffers, sendOfferLetter, getAllCandidates, getHRUsers, scheduleInterview, getOnboardingTasks, createOnboardingTask); sObject relations — `Application__c` looks up `Candidate__c` and `Job_c__c`; `Interview__c` links to `Application__c`; `Offer_Management__c` links to `Application__c`; `Onboarding_Task__c` links to `Candidate__c`; `Jobs_c__c` is referenced by `Application__c` and `Offer_Management__c`; `User` is used as Interviewer and to identify HR users.

6. **Class Name:** `EmailNotificationService` — **What it does:** Sends automated emails (application status updates, interview schedules, offer letters, notify HR on offer acceptance, onboarding task notices); designed to be invoked from triggers/handlers.
   Methods used: `sendApplicationStatusEmail`, `sendInterviewScheduledEmail`, `sendOfferLetterEmail`, `sendOfferAcceptanceToHR`, `sendOnboardingTaskEmail` — uses SOQL to load `Application__c`/`Interview__c`/`Offer_Management__c`/`Onboarding_Task__c` and related `Candidate__r`/`Job_c__r`, queries `User` for HR emails, and sends email via `Messaging.SingleEmailMessage` + `Messaging.sendEmail`.



Apex Triggers used

1. **ApplicationTrigger:** Calls the handler after an Application record is updated to handle status changes or related actions.
2. **InterviewTrigger:** Calls the handler after a new Interview record is inserted to handle notifications or related logic.

3. **OfferTrigger:** Calls the handler after an Offer record is updated to handle status changes or notifications.
4. **OnboardingTrigger:** Calls the handler after a new Onboarding Task is inserted to handle notifications or related actions.

Apex Triger Handlers used

1. **ApplicationTriggerHandler:** Sends an email to the candidate whenever their application stage is updated.
2. **InterviewTriggerHandler:** Sends an email to the candidate whenever a new interview is scheduled.
3. **OfferTriggerHandler:** Sends offer-related emails—either the offer letter to the candidate when status is “Sent” or notifies HR when the offer is “Accepted.”
4. **OnboardingTriggerHandler:** Sends an email to the candidate whenever a new onboarding task is created.