



# Database Models – Summary

This section provides an overview of the main database models used in the application, along with their purpose and relationships.

## User Model

- Inherits from Django's `AbstractUser` .
- Main user table for authentication and role management.
- Passwords are stored as hashed values; JWT Authentication is used for API access.
- Roles: Manager, Recruiter, Candidate, Client, Interviewer, Admin.
- Relationships: One-to-One with `UserProfile` , One-to-Many with `JobPostings` , linked as Manager to `Organization` , linked as Client to `ClientDetails` , used in `JobApplications` as Applicant/Sender/Receiver.

## User Profile Model

- One-to-One field with `User` .
- Stores additional information about users such as name, gender, phone, and address.
- Each user has exactly one profile.

## Organization Model

- Stores details of agencies or companies using HireSync.
- Each organization has a manager (linked to `User` ).
- Multiple recruiters linked via `ForeignKey` to `User` with `role='recruiter'` .
- Has multiple `JobPostings` .

## Recruiters Model

- Stores the recruiter information
- Connected to `Organization`

## Client Details Model

- Stores information of clients who are customers of an organization.
- Each client is a `User` (`ForeignKey` to `User` ).
- A client can belong to multiple organizations through `ClientOrganizations` .

## Client Organizations Model

- Connects clients and organizations in a many-to-many relationship.
- `ForeignKey` to `ClientDetails` and `Organization` .
- Used to track approval status for each client-organization pair.

## Client Organization Terms Model

- Stores negotiated or standard terms between a client and an organization.
- Includes CTC range, terms details, and negotiation flag.
- Linked to a specific `ClientOrganizations` entry.

## Job Postings Model

- Represents job postings created by clients or managers.
- `ForeignKey` to `Organization` and `User` (Client).
- Has multiple `JobLocations` and is linked to `SkillMetricsModel` , `JobPostTerms` , and `AssignedJobs` .

## Job Locations Model

- Stores different location entries for a job posting.
- Includes fields like location name, positions, and location type.
- Linked to a `JobPosting` and used in `JobApplications` .

## **Assigned Jobs Model**

- Tracks which recruiters are assigned to which job locations.
- Many-to-Many relationship with `User` (recruiters).
- Linked to `JobLocations` and `JobPostings` .

## **Skill Metrics Model**

- Stores required skills for each job posting.
- Indicates whether each skill is primary or secondary.
- Linked to a `JobPosting` .

## **Interviewer Details Model**

- Stores interviewer assignments for different interview rounds.
- Linked to a `JobPosting` and `User` (Interviewer).

## **Job Post Terms Model**

- Stores the agreed terms specific to a job posting.
- Includes details like CTC range and other conditions.
- Linked to `JobPosting` .

## **Candidate Resume Model**

- Stores resumes uploaded by candidates.
- Includes candidate name and other parsed details.
- Used in `JobApplications` .

## **Candidate Skillset Model**

- Stores individual skills of a candidate.
- Indicates whether each skill is primary or secondary.
- Linked to a `Candidate Resume` .

## **Interviews Scheduled Model**

- Stores interview scheduling details.
- Linked to candidate, recruiter, interviewer, and job application.
- Includes interview status and round information.

## **Job Applications Model**

- Represents applications submitted by candidates for job locations.
- Links to `JobLocations` , `User` (Candidate), `Candidate Resume` , and interview schedules.

- Tracks application status, sender, and 2receiver.

### **Candidate Profile Model**

- Stores extended profile information of a candidate.
- Linked to `User` (Candidate).
- Used in evaluations and interview processes.

### **Candidate Documents Model**

- Stores the documents of the candidate
- Linked to `Candidate Profile`

### **Candidate Certificates Model**

- Stores the certificates of the candidate
- Linked to `Candidate Profile`

### **Candidate Experiences Model**

- Stores the experiences of the candidate
- Linked to `Candidate Profile`

### **Candidate Evaluation Model**

- Stores evaluation results for a candidate's application.
- Linked to `JobApplications` , `CandidateProfile` , and scheduled interviews.
- Includes marks or scoring fields.

### **Selected Candidate Model**

- Tracks candidates selected for a job.
- Includes joining details, status, replacement info, and flags.
- Linked to `Candidate` and `JobApplication` .

### **Invoice Generated Model**

- Stores invoices generated after a candidate is selected and joined.
- Includes invoice code, linked selected candidate, organization, client, and associated terms.

### **Accountants Model**

- Each organization has an accountant.
- Linked to `Organization` and `Custom user`

## Replacement Candidates Model

- All the replacement of the candidates for the particular job post is stored here
- There are fields like `replacement_with` and `replaced_by` connected with `JobApplication` so that we can retrieve the job post, location from there.
- Status to track the status of the replacement

## Tickets Model

- Stores tickets raised by users
- Linked to `Users`

## Messages Model

- Stores the list of messages
- Linked to `Tickets` model

## Blog Posts Model

- Stores all the blogs that are created
- Connected with `User` (who are created)

## Notifications Model

- Stores notifications along with the categories it belongs to, so that we can track that which category the notification belongs to. we will use this in rendering the labels icons in the sidenav of the user
- Connected to the `User` (receiver field)
- Stores the status of the notification (seen = True or False)

## LinkedIn Integrations Model

- Stores the agencies linkedin oauth tokens (need to encrypt the data while storing the models)
- Connected to `Organization`

## Hiresync LinkedIn Cred Model

- Stores the linkedin credentials of the hiresync. This is used to upload the post for both agencies linkedin account and hiresync linkedin account.

## Features

- Stores the list of features along with the code and description.

## Plan

- Stores the available plans to the agencies while logged in.
- Many to many connection with the `Features` Model.

## PlanFeature

- Here based on the plan, we will store the limit for every feature.  
*At Present we haven't implemented any frontend dashboard to add the plans. Add using the backend django admin panel*

## Organization Plan

- Stores the organization along with the selected plan and some other details.
- Connected to `Organization` and `Plan`

## Plan History

- Plan history is used to store the history of the selected plans for the `Organization` and `User` .