# Lab: Django Models Basics

This document defines the problems for the in-class lab for the [**Python ORM course @ Software University**](https://softuni.bg/modules/137/python-db).

Submit your solutions to the SoftUni [**Judge system**](https://judge.softuni.org/Contests/4299/Django-Models-Basics-Lab).

## Employee Model

**Note: the problem cannot be submitted to the Judge system.**

In the **main\_app** create a model called **"Employee"** which will contain information about each employee in a company. Each employee must have:

* **name** - a **character field** with a max length of **30 characters**, containing the name of the employee.
* **email\_address** - an **email field**, containing the employee's email address.
* **photo** - an **URL field**, containing the URL path to the employee image.
* **birth\_date** - a **date field**, containing the birth date of the employee.
* **works\_full\_time** - a **Boolean field** that holds information about the type of employment. If the employee works full-time the field should be set to **True**, otherwise - it should be set to **False**.
* **created\_on** - **date** **and** **time** **field**; the field should be **set to the current date and time when the object is first created**.

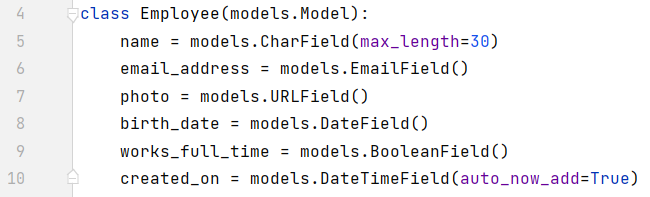
### Hints

First, let us open the **models.py** file in the **main\_app** Django application: A screenshot of a computer

Description automatically generated

Then, we will create a class that will subclass the **Model** class from the **django.db.models** directory: A screenshot of a computer

Description automatically generated

After that, we will write the fields of the model:

## Department Model

**Note: the problem cannot be submitted to the Judge system.**

In the same **main\_app** create a second model called **"Department"** containing information about each department in a company. Each department has:

* **code**
  + A **character field** with a max length of **4 characters**.
  + Should be set to be the **primary key**.
  + Each code is **unique**.
  + It indicates the code of the department.
* **name**
  + A **character field** with a max length of **50 characters**.
  + Each name should be **unique**.
  + It indicates the name of the department.
* **employees\_count**
  + A **positive** **integer** **field**.
  + If no value is provided, it should be set to a **default value of 1**.
  + Its **verbose name** should be set to **"Employees Count"**.
  + It indicates the predetermined number of employees that should work in that department.
* **location**
  + A **character field** with a max length of **20** **characters**.
  + **Optional** field.
  + Holds **predefined choices** of each **city name**: **"Sofia"**, **"Plovdiv"**, **"Burgas"**, **"Varna"**.
  + It indicates the department's location by city.
* **last\_edited\_on**
  + **Date** **and** **time** field.
  + It should be **set to the current date and time every time the object is saved**.
  + It should **not** **be editable**.

## Migrate the Models

Migrate the created models named **"Employee"** and **"Department"** to the database. **Check** the created database tables using **dbshell**. **Submit your project to the Judge system**.

### Hints

Open your terminal and run the following command to initiate the migration process: **"python manage.py makemigrations"**. This command will generate migration files based on the changes in your models:A screenshot of a computer

Description automatically generated

After generating the migration files, apply the migrations to the database using the following command: **"python manage.py migrate".** This command will create the database tables according to the model definitions:A screenshot of a computer

Description automatically generated

We can verify that the migration was successful by checking the database tables. We can use the **dbshell**.

Once we are confident that the migration was successful and the database tables are set up correctly, we will proceed to submit the project to the Judge system for evaluation.

First, we will **archive** theneeded files and directories to a **.zip file**. Note that we need to archive the **main\_app**, **orm\_skeleton**, **caller.py**, **manage.py**, and **requirements.txt**:

A screenshot of a computer

Description automatically generated

**Do not change the names of the files and folders when you submit them in the Judge system!**

Next, we will **add the .zip file to the Judge system** and click to **submit** the project:A screenshot of a computer

Description automatically generated

## Project Model

**Create a model** called **"Project"** containing information about each project in a department and **migrate it** to the database. Each project has:

* **name**
  + A **character field** with a max length of **100 characters**.
  + Each name should be **unique**.
  + It stores the name of the project.
* **description**
  + A **text field**.
  + **Optional** field.
  + It stores a detailed description of the project.
* **budget**
  + A **decimal** **field**.
  + **Optional** field.
  + It stores the project's budget with a maximum of **10 digits** and **2 decimal places**.
* **duration\_in\_days**
  + A **positive integer field**.
  + **Optional** field.
  + Its **verbose name** should be set to **"Duration in Days"**.
  + It stores the number of days the project is expected to last.
* **estimated\_hours**
  + A **float field**.
  + **Optional** field.
  + Its **verbose name** should be set to **"Estimated Hours"**.
  + It stores the estimated hours required for the project.
* **start\_date**
  + A **date field**.
  + Its **verbose name** should be set to **"Start Date"**.
  + **Optional** field.
  + If **no data is entered, by default** it should be **set to the current date when the object was first created**.
  + It stores the project's start date.
* **created\_on**
  + **Date** **and** **time** **field**.
  + It should be **set to the current date and time when the object is first created**.
  + It should **not** **be editable**.
* **last\_edited\_on**
  + **Date** **and** **time** **field**.
  + It should be **set to the current date and time every time the object is saved**.
  + It should **not** **be editable**.

### Hints

To allow the **start\_date** field to be **set manually** while also having a **default** **value of today's date** when no value is provided, we can use the **default** argument with the **date.today** function:

**start\_date = models.DateField(default=date.today)**