

Порахуй досвід

Ти працюєш над прототипом вебзастосунку для наймання працівників. Перша фіча, яку тобі потрібно імплементувати — обрахунок загального стажу кандидата. На вході ти отримуватимеш дані про періоди роботи кандидата на попередніх місцях у такому форматі:

- Jul 2015 - Sep 2017
- Oct 2017 - May 2019
- Nov 2017 - Nov 2018

Загальний стаж потрібно розрахувати у повних роках, включаючи початковий і кінцевий місяць у кожному записі. При цьому, якщо людина деякий період працювала на кількох роботах, цей період має враховуватися лише один раз. Наприклад, для наведених записів стаж рахуватиметься так:

- Jul 2015 - Sep 2017 - 27 місяців
- Oct 2017 - May 2019 - 20 місяців
- Nov 2017 - Oct 2018 - 12 місяців, але оскільки вони повністю входять у попередній період, то не додаються до загального стажу

У результаті маємо $27 + 20 = 47$ місяців, або 3 повних роки.

Функціональність для обрахунку стажу повинна бути частиною вебзастосунку із JSON API. Він матиме один ендпоінт `/candidate`, який приймає `GET` і `POST` запити.

Ось приклад `POST` запиту:

```
POST /candidates/ HTTP/1.1
Host: localhost:8000
Content-Type: application/json
Accept: */*
Content-Length: 179

{
  "name": "Jane Doe",
  "workExperience": [
    {"start": "Jul 2018", "end": "Aug 2019"},
    {"start": "Sep 2019", "end": "Aug 2021"},
    {"start": "Oct 2019", "end": "Dec 2021"}
  ]
}
```

Як результат обробки цього запиту інформація про кандидата має бути збережена у базу даних.

У відповідь на **GET** запит застосунок має повертати список попередньо збережених кандидатів у такому форматі:

```
{
  "candidates": [
    {
      "id": 11,
      "name": "Jane Doe",
      "totalExperience": 4,
      "workExperience": [
        {
          "start": "Jul 2017",
          "end": "Aug 2019"
        },
        {
          "start": "Sep 2019",
          "end": "Aug 2021"
        },
        {
          "start": "Oct 2019",
          "end": "Dec 2021"
        }
      ]
    },
    {
      "id": 12,
      "name": "Richard Roe",
      "totalExperience": 3,
      "workExperience": [
        {
          "start": "Jul 2015",
          "end": "Sep 2017"
        },
        {
          "start": "Oct 2017",
          "end": "May 2019"
        },
        {
          "start": "Nov 2017",
          "end": "Nov 2018"
        }
      ]
    }
  ]
}
```

Кандидати мають бути посортовані за загальним стажем (атрибут **totalExperience**) у спадному порядку (від найдосвідченіших до найменш досвідчених).

Технічні вимоги

Як вебфреймворк для виконання цього завдання рекомендується Django, але допускаються й інші, такі як Flask чи FastAPI. Для збереження даних найліпше використати SQLite. Рішення повинне містити інструкцію, як запустити проект на новому середовищі.

Примітки

У рамках цього завдання можна припустити, що вхідні дані завжди будуть мати всі необхідні атрибути у правильному форматі. Тому валідація вхідних даних не входить в рамки задачі і не буде враховуватися при оцінці. Також не буде враховуватися імплементація будь-яких додаткових функцій крім описаних у завданні. Набір тестів, з іншого боку, хоч не є обов'язковим, але буде вважатися плюсом.