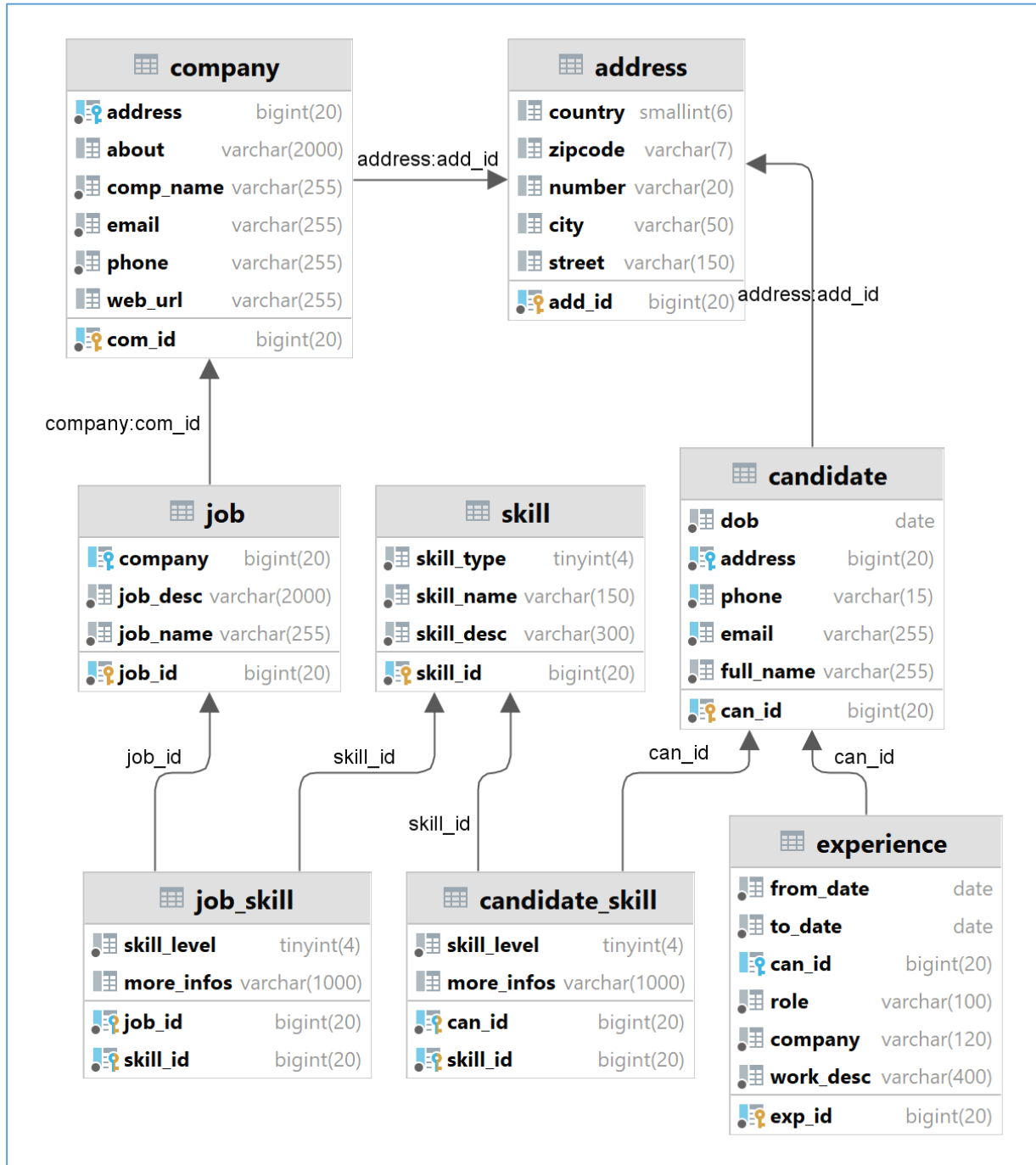


LAB WEEK 05

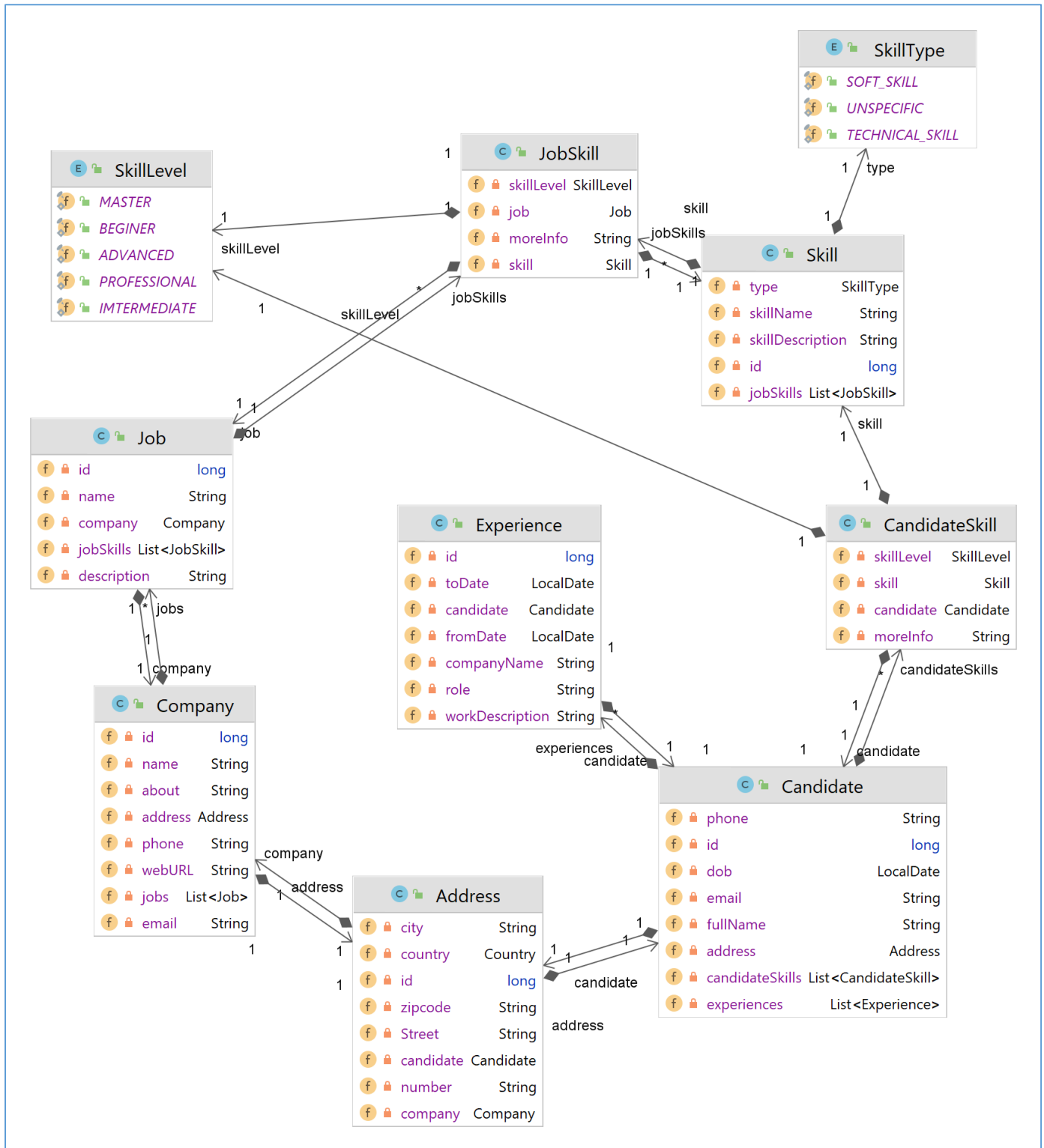
Cho csdl như hình. Hãy tạo một ứng dụng web sử dụng spring boot rồi thực hiện các công việc sau



1. Tạo các entities sao cho khi thực thi sẽ tạo ra các bảng như hình
2. Viết các repositories interface
3. Viết các lớp services
4. Tạo các trang web cho phép công ty đăng tin tuyển người với các skill mong muốn
5. Các ứng viên khi log vào sẽ được gợi ý các công việc có skill phù hợp với mình
6. Giúp các công ty tìm các ứng viên có skill phù hợp rồi gửi mail mời.
7. Đề xuất một số skill mà ứng viên chưa có để học.

8. Và các yêu cầu khác

Class diagram tham khảo



Có thể dùng CountryCode thay cho Country enum bằng cách thêm một dependency

```
implementation 'com.neovisionaries:nv-i18n:1.29'
```

Link: <https://github.com/TakahikoKawasaki/nv-i18n>

Các dependencies của project như sau

```
implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
implementation 'org.springframework.boot:spring-boot-starter-web'
implementation 'org.springframework.boot:spring-boot-starter-web-services'
implementation 'org.mariadb.jdbc:mariadb-java-client:3.2.0'

compileOnly 'org.projectlombok:lombok'
developmentOnly 'org.springframework.boot:spring-boot-devtools'
annotationProcessor 'org.projectlombok:lombok'

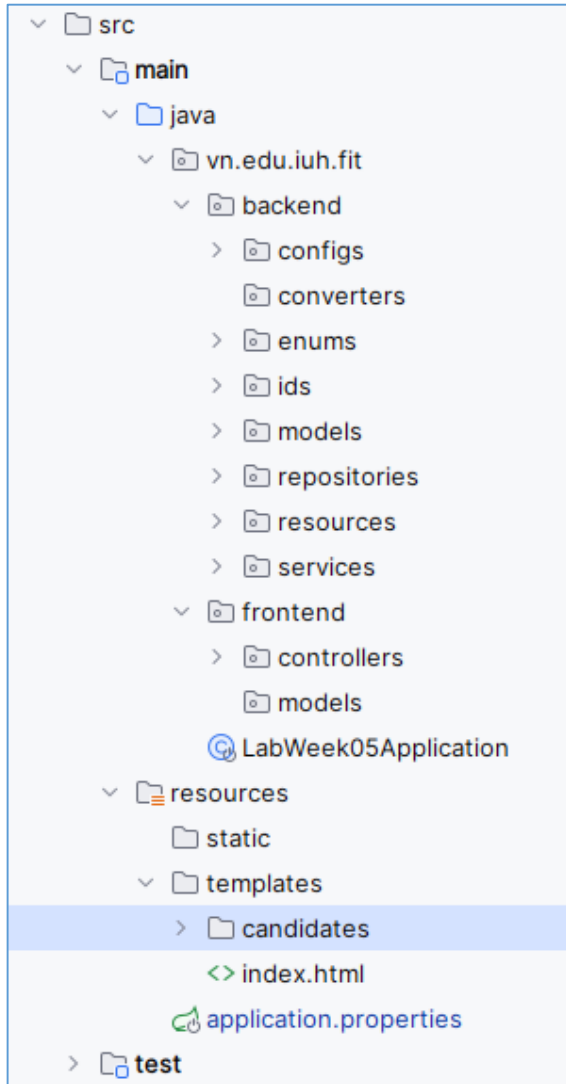
//for country code
implementation 'com.neovisionaries:nv-il8n:1.29'

//For database REST
implementation 'org.springframework.data:spring-data-rest-core:4.1.4'

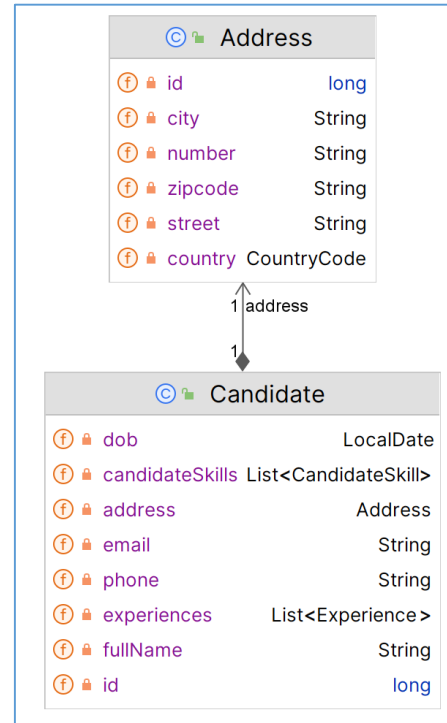
testImplementation 'org.springframework.boot:spring-boot-starter-test'
testImplementation 'org.springframework.security:spring-security-test'
```

PAGING EXAMPLE

Trong hướng dẫn này, chúng ta làm việc với phân trang trong Java web.



Hình 1



Hình 2

Cấu trúc project của thực hành phần 5 tham khảo như hình 1 (Cấu trúc này có thể thay đổi theo yêu cầu khác).

Trong gói models (entities) của phần backend, ta có hai entities Address và Candidate có mối quan hệ như class diagram ở hình 2.

Để phục vụ cho việc phân trang, ta thêm vào 1000 mẫu tin ngẫu nhiên. Code cho việc này như sau

```
package vn.edu.iuh.fit;

import com.neovisionaries.i18n.CountryCode;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.Bean;
import vn.edu.iuh.fit.backend.models.Address;
```

```

import vn.edu.iuh.fit.backend.models.Candidate;
import vn.edu.iuh.fit.backend.repositories.AddressRepository;
import vn.edu.iuh.fit.backend.repositories.CandidateRepository;

import java.time.LocalDate;
import java.util.Random;

@SpringBootApplication
public class LabWeek05Application {

    public static void main(String[] args) {
        SpringApplication.run(LabWeek05Application.class, args);
    }

    @Autowired
    private CandidateRepository candidateRepository;
    @Autowired
    private AddressRepository addressRepository;
    @Bean
    CommandLineRunner initData() {
        return args -> {
            Random rnd = new Random();
            for (int i = 1; i < 1000; i++) {
                Address add = new Address(rnd.nextInt(1, 1000) + "", "Quang Trung", "HCM",
                    rnd.nextInt(70000, 80000) + "", CountryCode.VN );
                addressRepository.save(add);

                Candidate can = new Candidate("Name #" + i,
                    LocalDate.of(1998, rnd.nextInt(1, 13), rnd.nextInt(1, 29)),
                    add,
                    rnd.nextLong(1111111111L, 9999999999L) + "",
                    "email_" + i + "@gmail.com");
                candidateRepository.save(can);
                System.out.println("Added: " + can);
            }
        };
    }
}

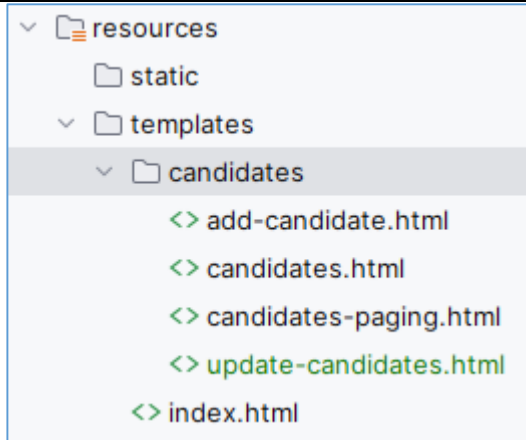
```

Như vậy, với đoạn code này, ta có 1000 mẫu tin trong csdl.

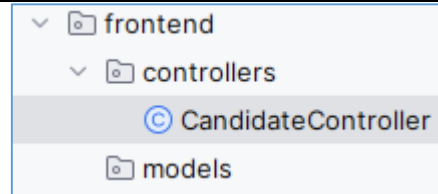
Trong phần resources của project, ta nhóm các tác vụ tập trung liên quan đến candidates trong một gói riêng cùng tên như hình 3.

Trong thư mục này, các files:

- candidates.html dành cho việc hiển thị các ứng viên không phân trang
- candidates-paging.html dành cho việc hiển thị các ứng viên có phân trang



Hình 3



Hình 4

** Ở đây chúng ta dùng thư viện bootstrap cho việc hiển thị cũng như các thao tác khác cho client.

Trong gói services, ta thêm lớp `CandidateServices` với nội dung

```
package vn.edu.iuh.fit.backend.services;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.*;
import org.springframework.stereotype.Service;
import vn.edu.iuh.fit.backend.models.Candidate;
import vn.edu.iuh.fit.backend.repositories.CandidateRepository;

import java.util.Collections;
import java.util.List;

@Service
public class CandidateServices {
    @Autowired
    private CandidateRepository candidateRepository;

    public Page<Candidate> findAll(int pageNo, int pageSize, String sortBy, String
sortDirection) {
        Sort sort = Sort.by(Sort.Direction.fromString(sortDirection), sortBy);
        Pageable pageable = PageRequest.of(pageNo, pageSize, sort);

        return candidateRepository.findAll(pageable);
    }

    public Page<Candidate> findPaginated(Pageable pageable) {
        int pageSize = pageable.getPageSize();
        int currentPage = pageable.getPageNumber();
        int startItem = currentPage * pageSize;
        List<Candidate> list;
        List<Candidate> candidates = candidateRepository.findAll();

        if (candidates.size() < startItem) {
            list = Collections.emptyList();
        } else {
            int toIndex = Math.min(startItem + pageSize, candidates.size());
            list = candidates.subList(startItem, toIndex);
        }
    }
}
```

```

        Page<Candidate> candidatePage
            = new PageImpl<>(list, PageRequest.of(currentPage, pageSize),
candidates.size());

        return candidatePage;
    }
}

```

LOAD DỮ LIỆU KHÔNG PHÂN TRANG

Trong gói front-end, ta thêm các gói như hình 4. Code cho CandidateController tham khảo như sau

```

package vn.edu.iuh.fit.frontend.controllers;

@Controller
public class CandidateController {
    @Autowired
    private CandidateRepository candidateRepository;
    @Autowired
    private CandidateServices candidateServices;

    @GetMapping("/list")
    public String showCandidateList(Model model) {
        model.addAttribute("candidates", candidateRepository.findAll());
        return "candidates/candidates";
    }
}

```

Phương thức này sẽ load toàn bộ candidates trong csld đưa vào một list có tên candidates

Trên view (candidates/ candidates.html), ta thiết kế trang html có code như sau:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Candidates</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwvykc2MPK8M2HN"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
        crossorigin="anonymous"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<div th:switch="{candidates}" >
    <h2 th:case="null">No candidates yet!</h2>
    <div th:case="*">
        <h2>Candidates</h2>
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>Id</th>

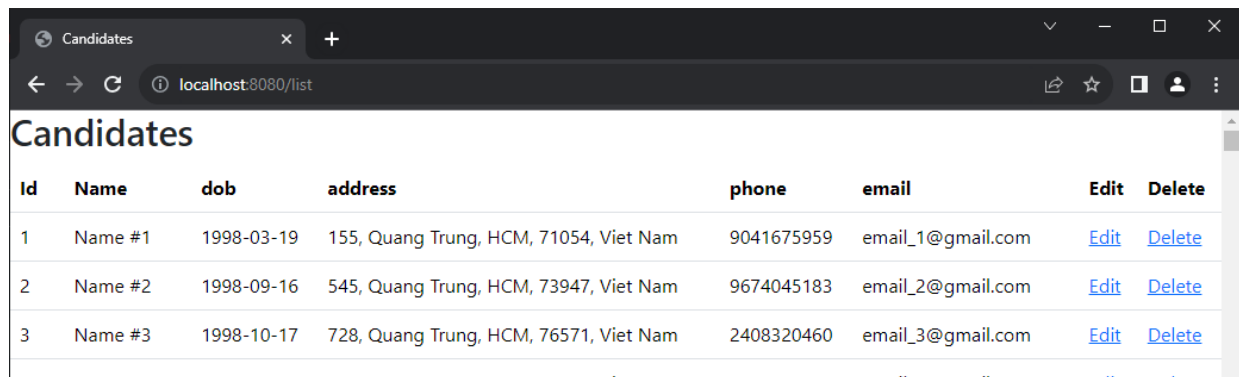
```

```

        <th>Name</th>
        <th>dob</th>
        <th>address</th>
        <th>phone</th>
        <th>email</th>
        <th>Edit</th>
        <th>Delete</th>
    </tr>
</thead>
<tbody>
<tr th:each="candidate : ${candidates}">
    <td th:text="${candidate.id}"></td>
    <td th:text="${candidate.fullName}"></td>
    <td th:text="${candidate.dob}"></td>
    <td th:text="${candidate.address}"></td>
    <td th:text="${candidate.phone}"></td>
    <td th:text="${candidate.email}"></td>
    <td><a th:href="@{/edit/{id} (id=${candidate.id})}">Edit</a></td>
    <td><a th:href="@{/delete/{id} (id=${candidate.id})}">Delete</a></td>
</tr>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

Trang html này sẽ hiển thị tất cả candidates dạng như sau (<http://localhost:8080/list>)



Id	Name	dob	address	phone	email	Edit	Delete
1	Name #1	1998-03-19	155, Quang Trung, HCM, 71054, Viet Nam	9041675959	email_1@gmail.com	Edit	Delete
2	Name #2	1998-09-16	545, Quang Trung, HCM, 73947, Viet Nam	9674045183	email_2@gmail.com	Edit	Delete
3	Name #3	1998-10-17	728, Quang Trung, HCM, 76571, Viet Nam	2408320460	email_3@gmail.com	Edit	Delete
4	Name #4	1998-05-01	500, Quang Trung, HCM, 71113, Viet Nam	3137003370	email_4@gmail.com	Edit	Delete

LOAD DỮ LIỆU CÓ PHÂN TRANG

Trong lớp CandidateController ta thêm phương thức như sau:

```

@GetMapping("/candidates")
public String showCandidateListPaging(Model model,
    @RequestParam("page") Optional<Integer> page,
    @RequestParam("size") Optional<Integer> size) {
    int currentPage = page.orElse(1);
    int pageSize = size.orElse(10);
    Page<Candidate> candidatePage= candidateServices.findPaginated(
        PageRequest.of(currentPage - 1, pageSize)
    );

    model.addAttribute("candidatePage", candidatePage);

    int totalPages = candidatePage.getTotalPages();
    if (totalPages > 0) {

```



```

        List<Integer> pageNumbers = IntStream.rangeClosed(1, totalPages)
            .boxed()
            .collect(Collectors.toList());
        model.addAttribute("pageNumbers", pageNumbers);
    }
    return "candidates/candidates-paging";
}

```

Trong phương thức `showCandidateListPaging` các tham số có ý nghĩa:

- `model`: dành cho việc chuyển dữ liệu qua lại giữa views (trang html) và tầng controllers
- `page`: là trang hiện tại, mặc định là 1 (0 trong index)
- `size`: là kích thước trang (mặc định là 10)

Thêm trang `candidates-paging.html` như sau

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Candidates</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-
T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN"
        crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-
C6RzsynM9kWDrmNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
        crossorigin="anonymous"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<div th:switch="{candidates}" >
    <h2 th:case="null">No candidates yet!</h2>
    <div th:case="*">
        <h2>Candidates</h2>
        <table class="table table-hover">
            <thead>
                <tr>
                    <th>Id</th>
                    <th>Name</th>
                    <th>dob</th>
                    <th>address</th>
                    <th>phone</th>
                    <th>email</th>
                    <th>Edit</th>
                    <th>Delete</th>
                </tr>
            </thead>
            <tbody>
                <tr th:each="candidate : {candidates}">
                    <td th:text="{candidate.id}"></td>
                    <td th:text="{candidate.fullName}"></td>
                    <td th:text="{candidate.dob}"></td>
                    <td th:text="{candidate.address}"></td>
                    <td th:text="{candidate.phone}"></td>
                    <td th:text="{candidate.email}"></td>
                    <td><a th:href="@{/edit/{id} (id={candidate.id})}">Edit</a></td>

```

```

        <td><a th:href="@{/delete/{id}(id=${candidate.id})}">Delete</a></td>
    </tr>
</tbody>
</table>
</div>
</div>
</body>
</html>

```

Vào trang <http://localhost:8080/candidates> ta có kết quả phân trang như hình

LIST CANDIDATES					
1	Name #1	1998-03-19	155, Quang Trung, HCM, 71054, Viet Nam	9041675959	email_1@gmail.com
2	Name #2	1998-09-16	545, Quang Trung, HCM, 73947, Viet Nam	9674045183	email_2@gmail.com
3	Name #3	1998-10-17	728, Quang Trung, HCM, 76571, Viet Nam	2408320460	email_3@gmail.com
4	Name #4	1998-05-01	590, Quang Trung, HCM, 71443, Viet Nam	2127092379	email_4@gmail.com
5	Name #5	1998-10-24	719, Quang Trung, HCM, 75509, Viet Nam	9044044851	email_5@gmail.com
6	Name #6	1998-11-13	682, Quang Trung, HCM, 78010, Viet Nam	6991525666	email_6@gmail.com
7	Name #7	1998-03-06	306, Quang Trung, HCM, 76839, Viet Nam	7706811271	email_7@gmail.com
8	Name #8	1998-08-23	479, Quang Trung, HCM, 75408, Viet Nam	9271722501	email_8@gmail.com
9	Name #9	1998-05-22	875, Quang Trung, HCM, 76126, Viet Nam	3760520631	email_9@gmail.com
10	Name #10	1998-01-12	444, Quang Trung, HCM, 74427, Viet Nam	1748634133	email_10@gmail.com

[1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [15](#) [16](#) [17](#) [18](#) [19](#) [20](#) [21](#) [22](#) [23](#) [24](#) [25](#) [26](#) [27](#) [28](#) [29](#) [30](#) [31](#) [32](#) [33](#) [34](#) [35](#) [36](#) [37](#) [38](#) [39](#) [40](#)
[41](#) [42](#) [43](#) [44](#) [45](#) [46](#) [47](#) [48](#) [49](#) [50](#) [51](#) [52](#) [53](#) [54](#) [55](#) [56](#) [57](#) [58](#) [59](#) [60](#) [61](#) [62](#) [63](#) [64](#) [65](#) [66](#) [67](#) [68](#) [69](#) [70](#) [71](#) [72](#) [73](#) [74](#) [75](#) [76](#) [77](#)
[78](#) [79](#) [80](#) [81](#) [82](#) [83](#) [84](#) [85](#) [86](#) [87](#) [88](#) [89](#) [90](#) [91](#) [92](#) [93](#) [94](#) [95](#) [96](#) [97](#) [98](#) [99](#) [100](#)

Thay đổi địa chỉ url dạng: <http://localhost:8080/candidates?size=10&page=9> hoặc chọn các trang hiển thị để thử nghiệm.