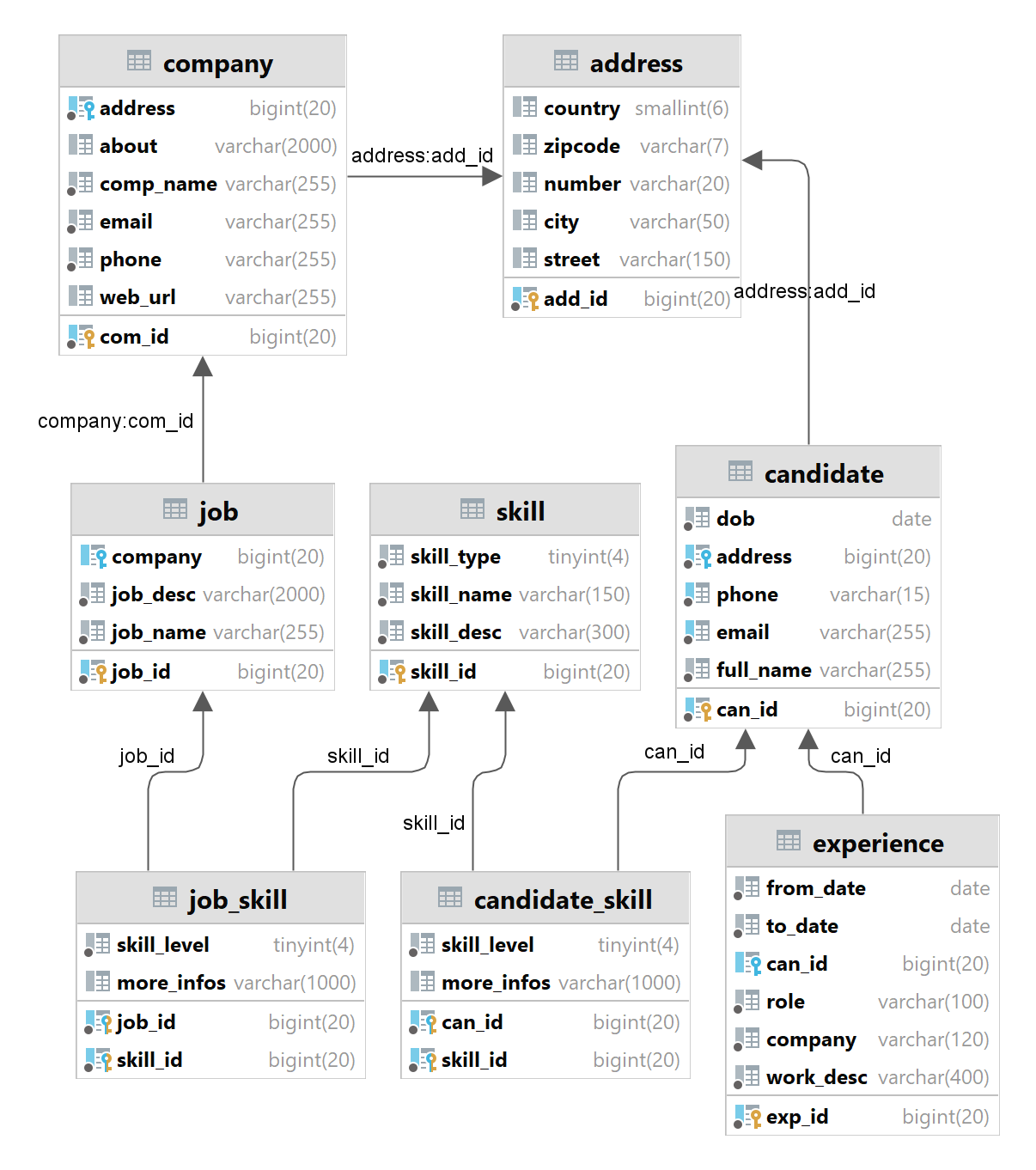
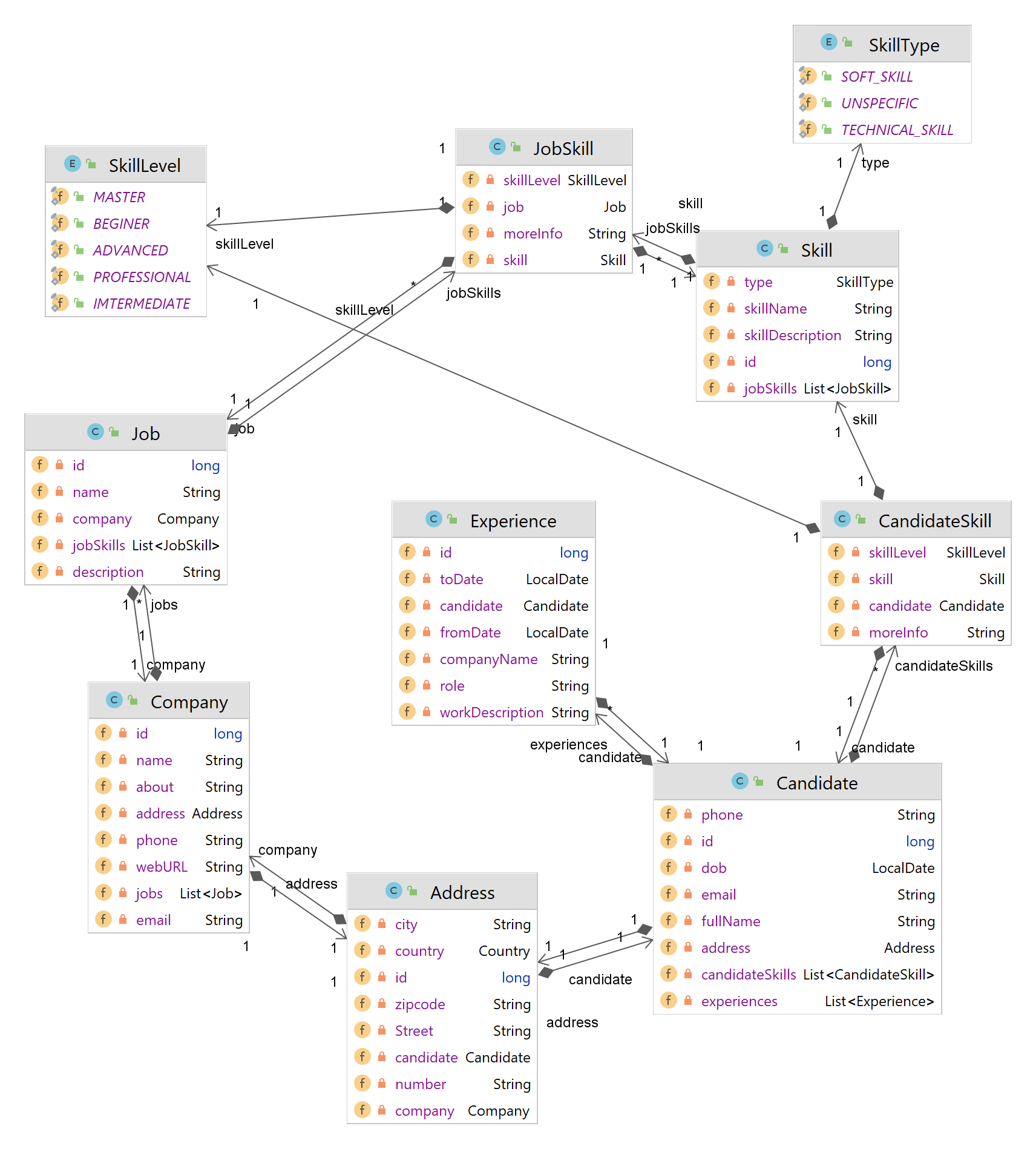
# LAB WEEK 05

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



1. Tạo các enities 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-i18n: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.

## Load dữ liệu không phân trang

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

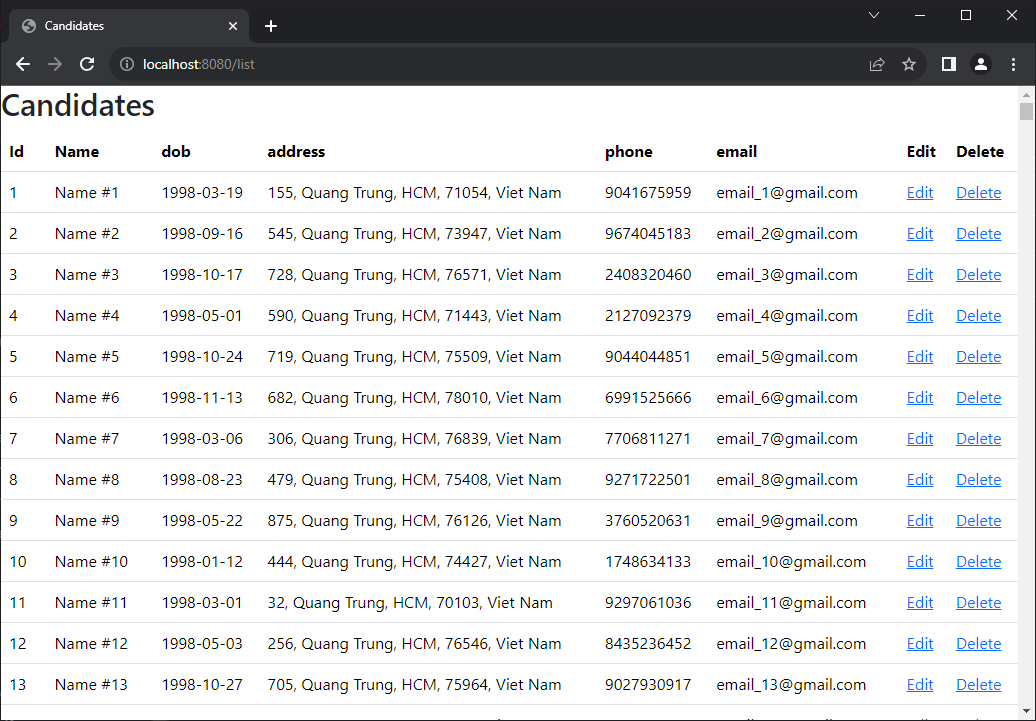
|  |
| --- |
| package vn.edu.iuh.fit.frontend.controllers;  @Controller public class CandidateController {  @Autowired  private CandidateRepository candidateRepository;  @Autowired  private CandidateServices candidateServices;   @GetMapping("/**list**")  public String showCandidateList(Model model) {  model.addAttribute("candidates", candidateRepository.findAll());  return "candidates/candidates";  } } |

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

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.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> |

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



## 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";  } |

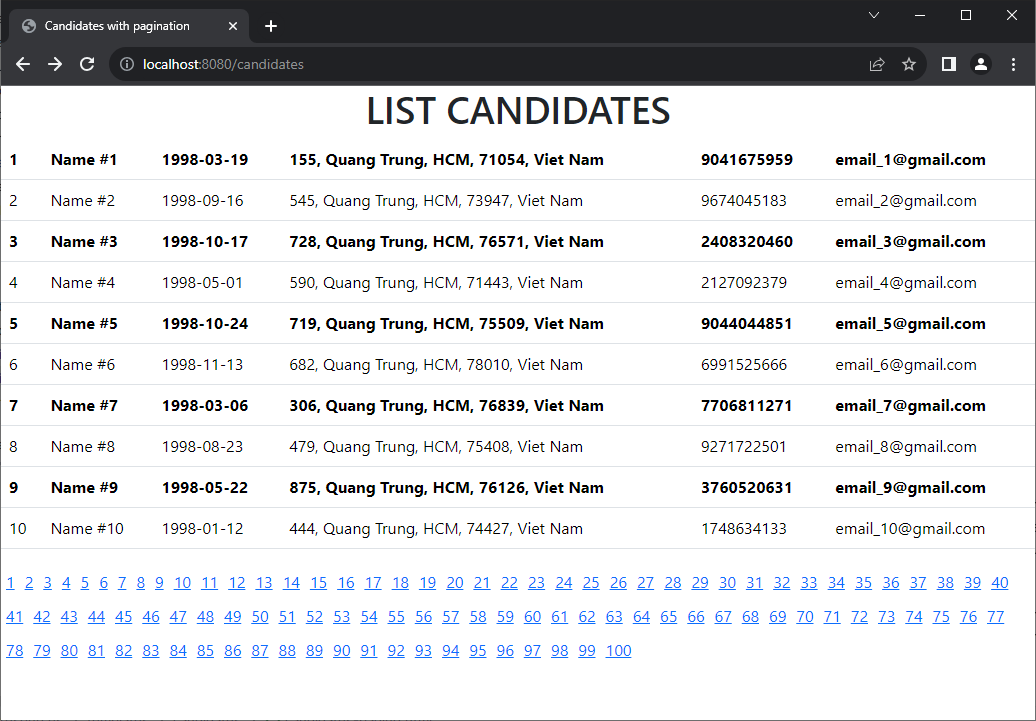
Trong phương thức showCandidateListPaging các tham số có ý nghĩa:

* 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



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