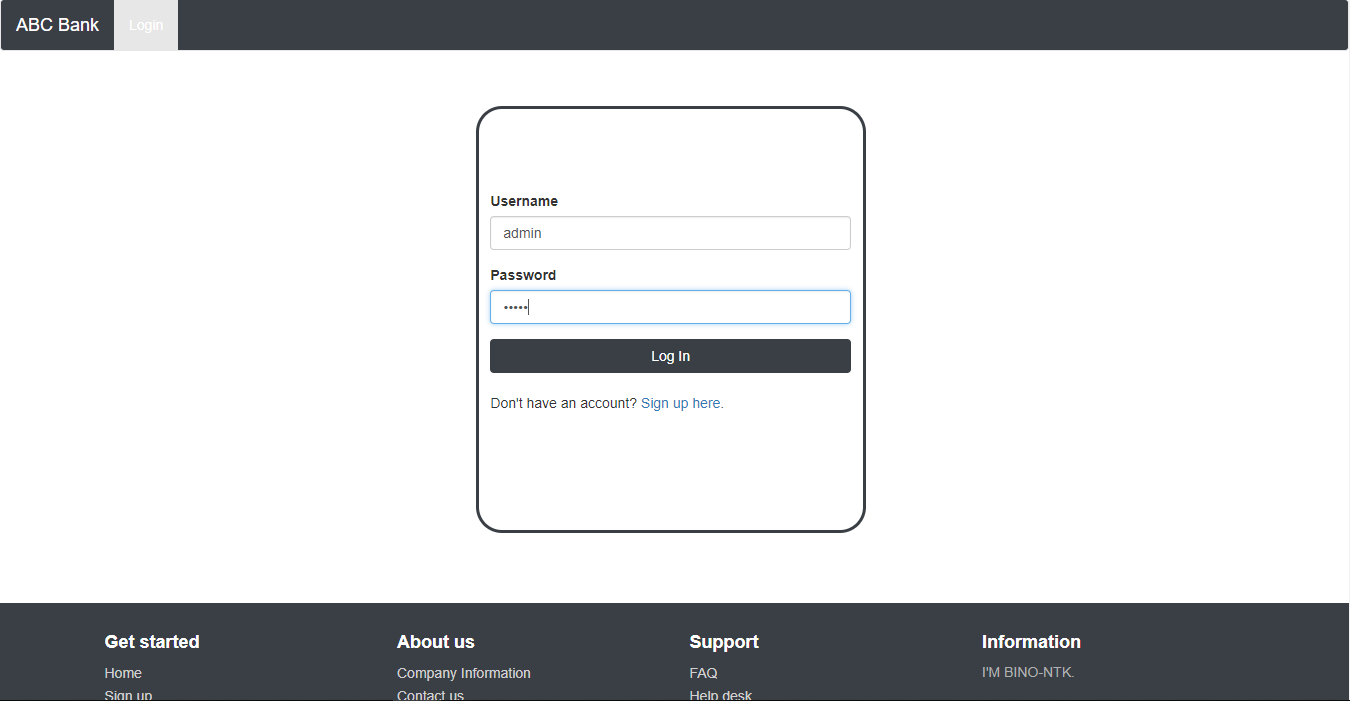
**HƯỚNG DẪN SỬ DỤNG UI**

**------------**

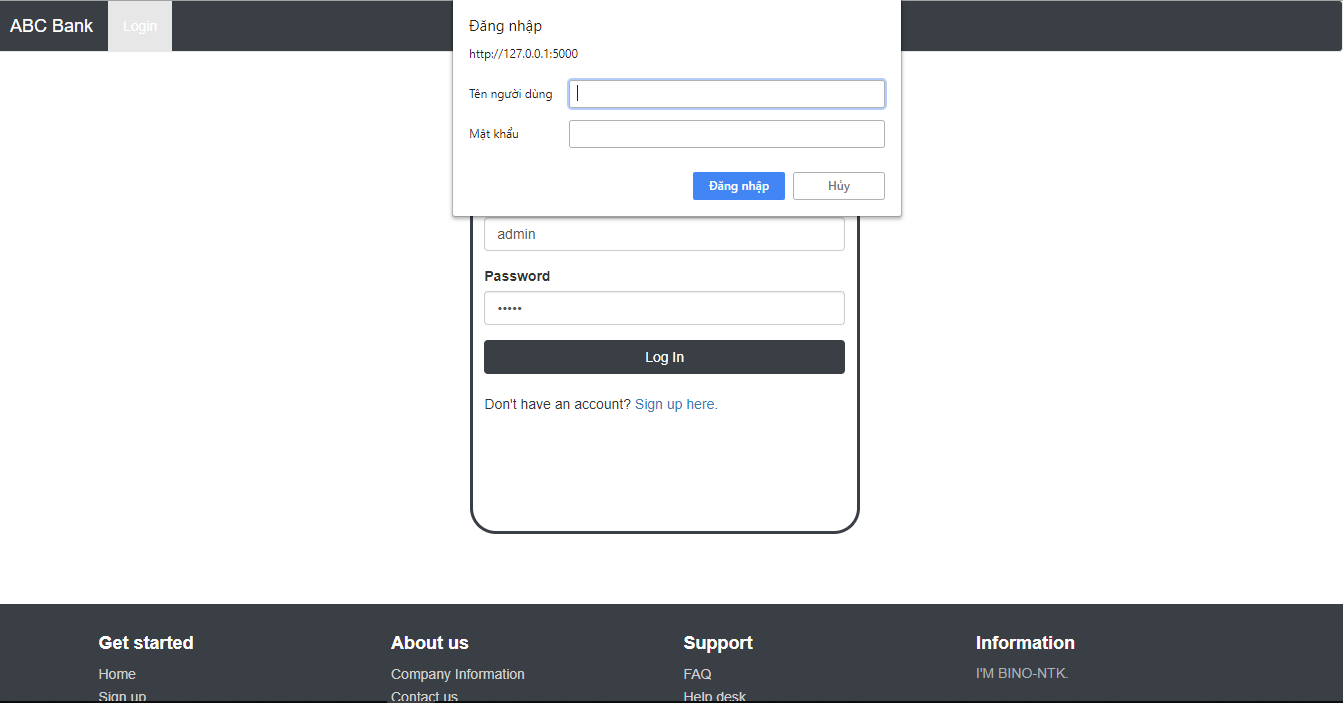
1. **Trang đăng nhập đơn giản:**

*(Thực hiện hướng dẫn tạo cơ sở dữ liệu trước)*



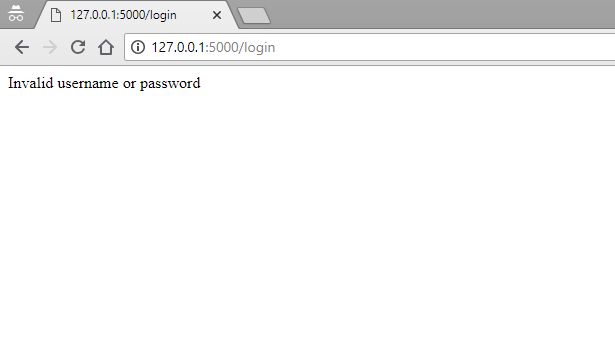
Có 2 loại tài khoản dành cho admin và người dùng thông thường, tài khoản mặc định cho admin là admin-admin(username và password tương ứng). Và 1 tà khoản với role normal là (test\_01-123456). Để test từng loại tài khoản hãy đăng nhập tương ứng.

Ngoài ra người dùng có thể tạo tài khoản mới bằng cách chọn và *Sign up here* trên giao diện nhưng tài khoản mới tạo này chỉ có role thông thường(Đọc thêm phần 1 số lưu ý). Trước hết hãy đăng nhập bằng tài khoản admin-admin để test trang admin với tất cả các tính năng của hệ thống. Sau khi bấm Log in trình duyệt sẽ yêu cầu bảo mật:

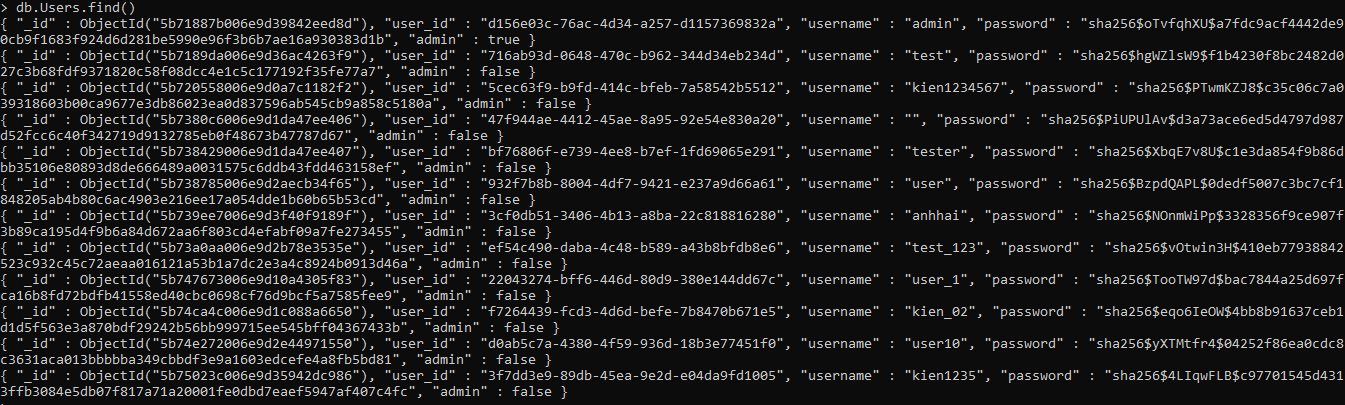


Tiếp tục nhập usename và password như cũ admin-admin.

Nếu đăng nhập sai sẽ hiện thông báo lỗi:



Nếu đăng nhập sai hãy tắt của sổ trình duyệt đó đi và mở 1 tab mới sau đó đăng nhập lại(nên test bằng tab ẩn danh) và đăng nhập lại.



Database với password được mã hóa(pwd được hash trước khi lưu vào db)

{

"\_id": { "$oid": "5b71887b006e9d39842eed8d" },

"user\_id": "d156e03c-76ac-4d34-a257-d1157369832a",

"username": "admin",

"password": "sha256$oTvfqhXU$a7fdc9acf4442de90cb9f1683f924d6d281be5990e96f3b6b7ae16a930383d1b",

"admin": true

} {

"\_id": { "$oid": "5b7189da006e9d36ac4263f9" },

"user\_id": "716ab93d-0648-470c-b962-344d34eb234d",

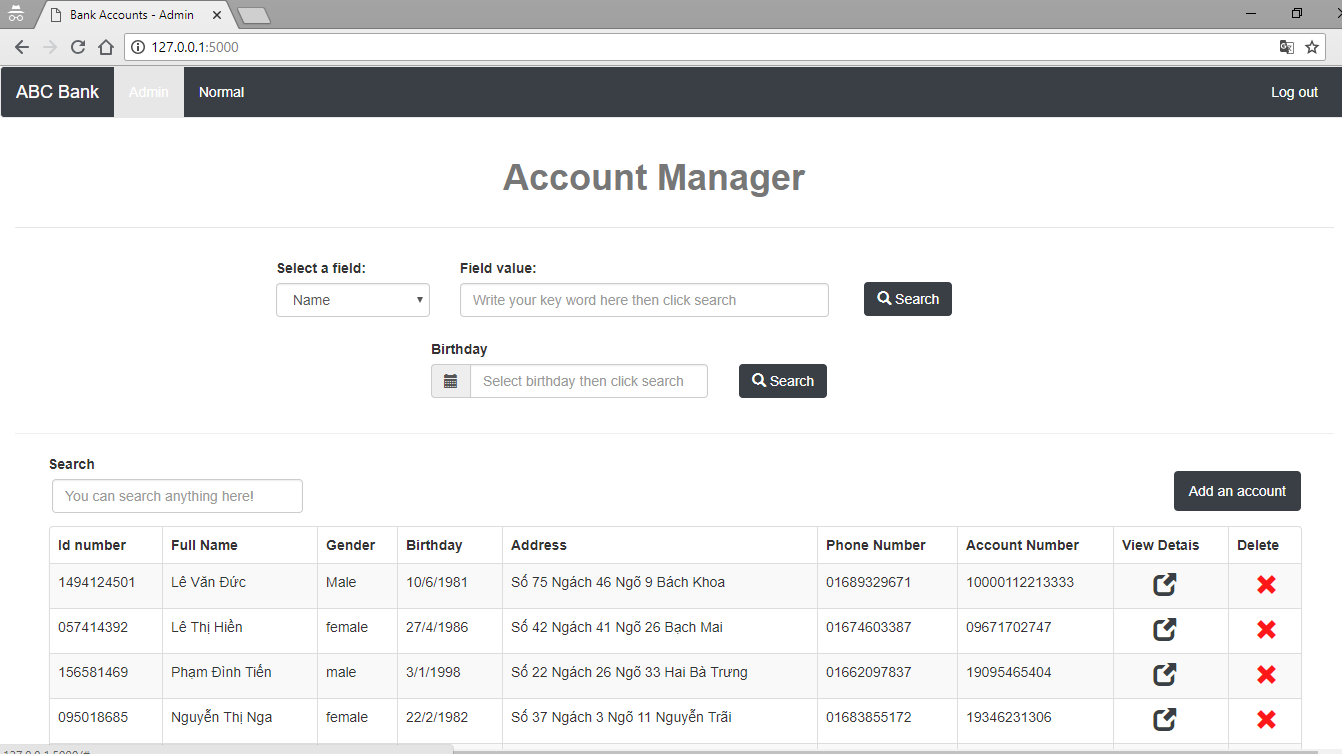
"username": "test",

"password": "sha256$hgWZlsW9$f1b4230f8bc2482d027c3b68fdf9371820c58f08dcc4e1c5c177192f35fe77a7",

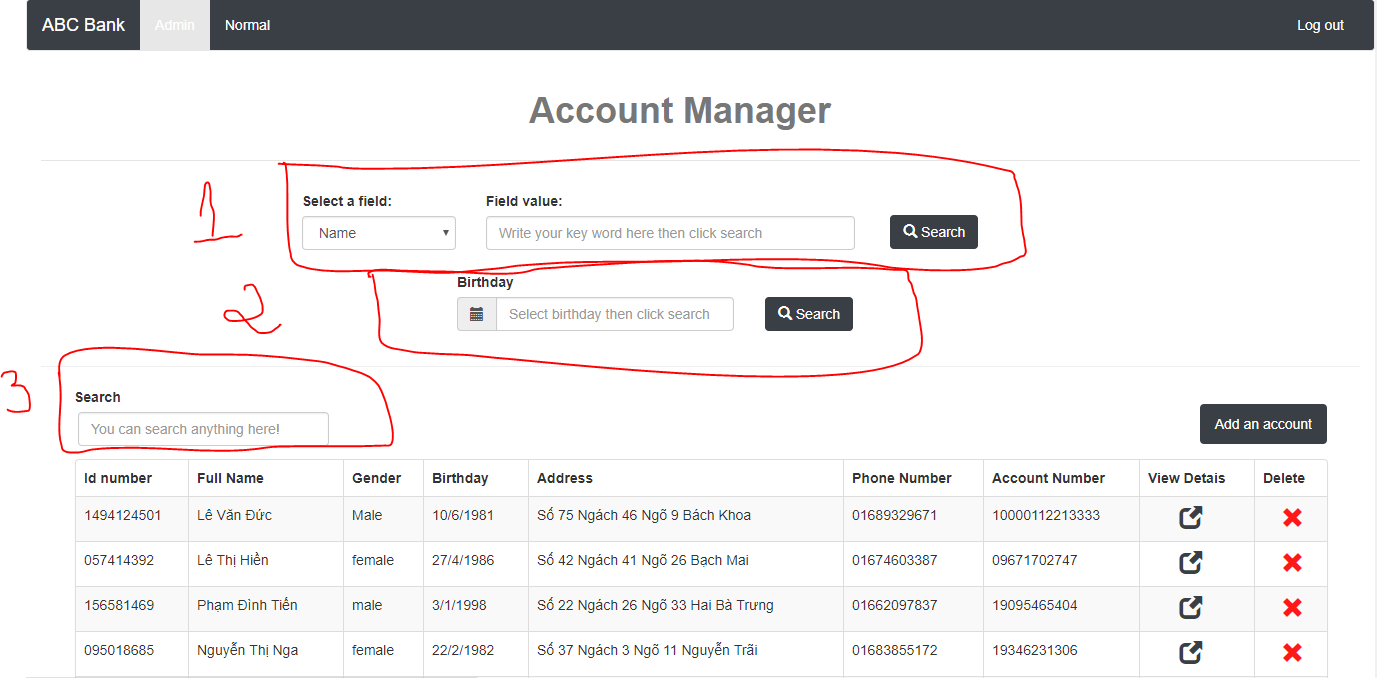
"admin": false

}

1. **Tìm kiếm**

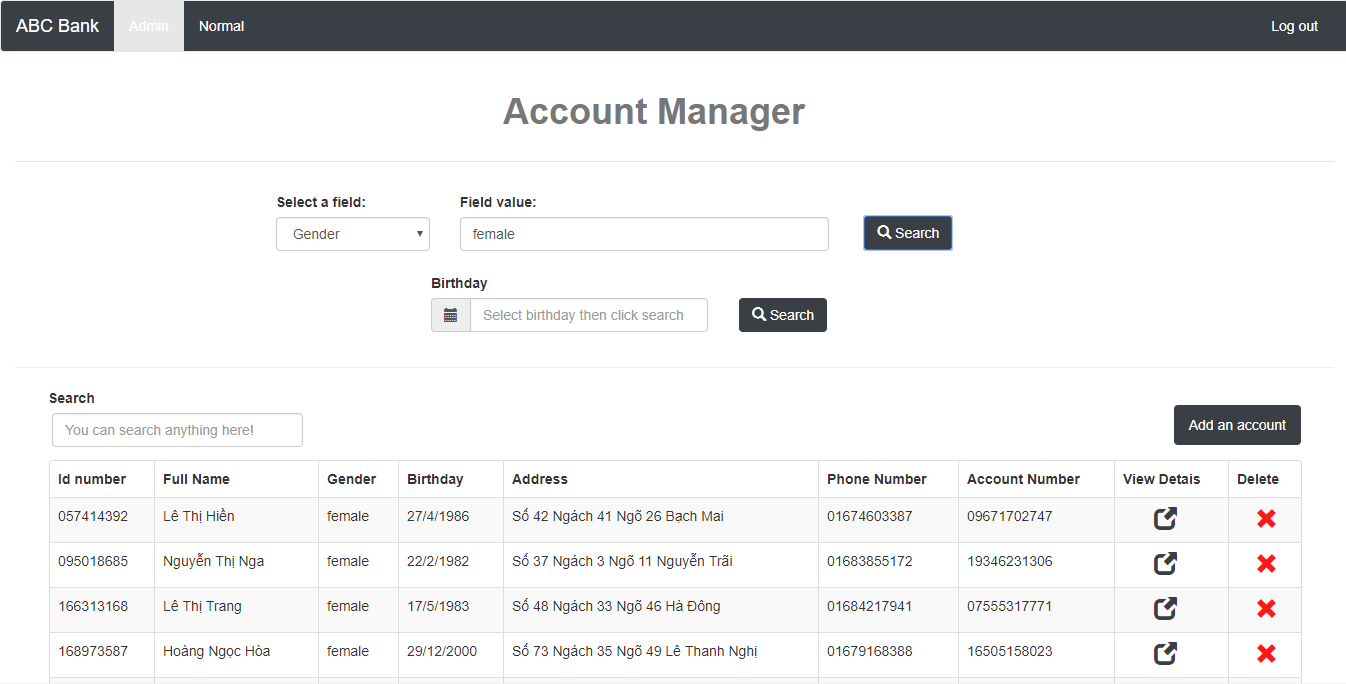


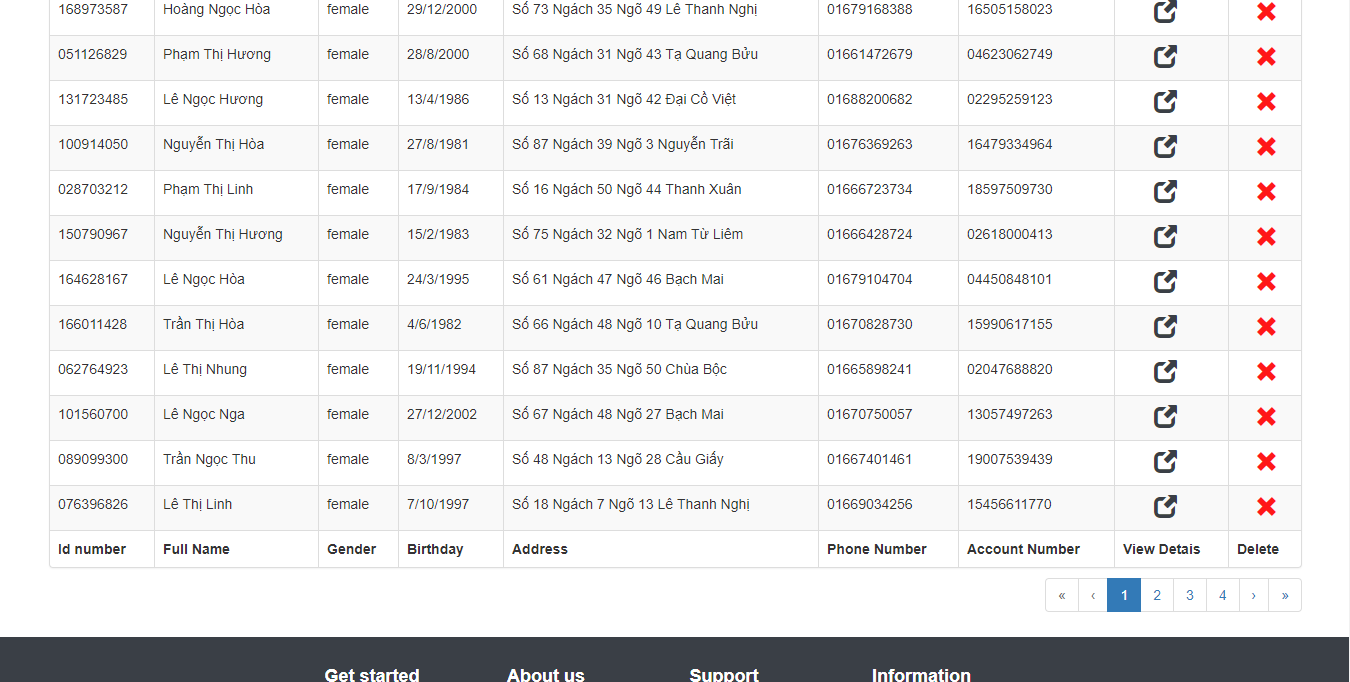
Giao diện admin page sau khi đăng nhập thành công.



Hỗ trợ 3 loại search(Hình vẽ):

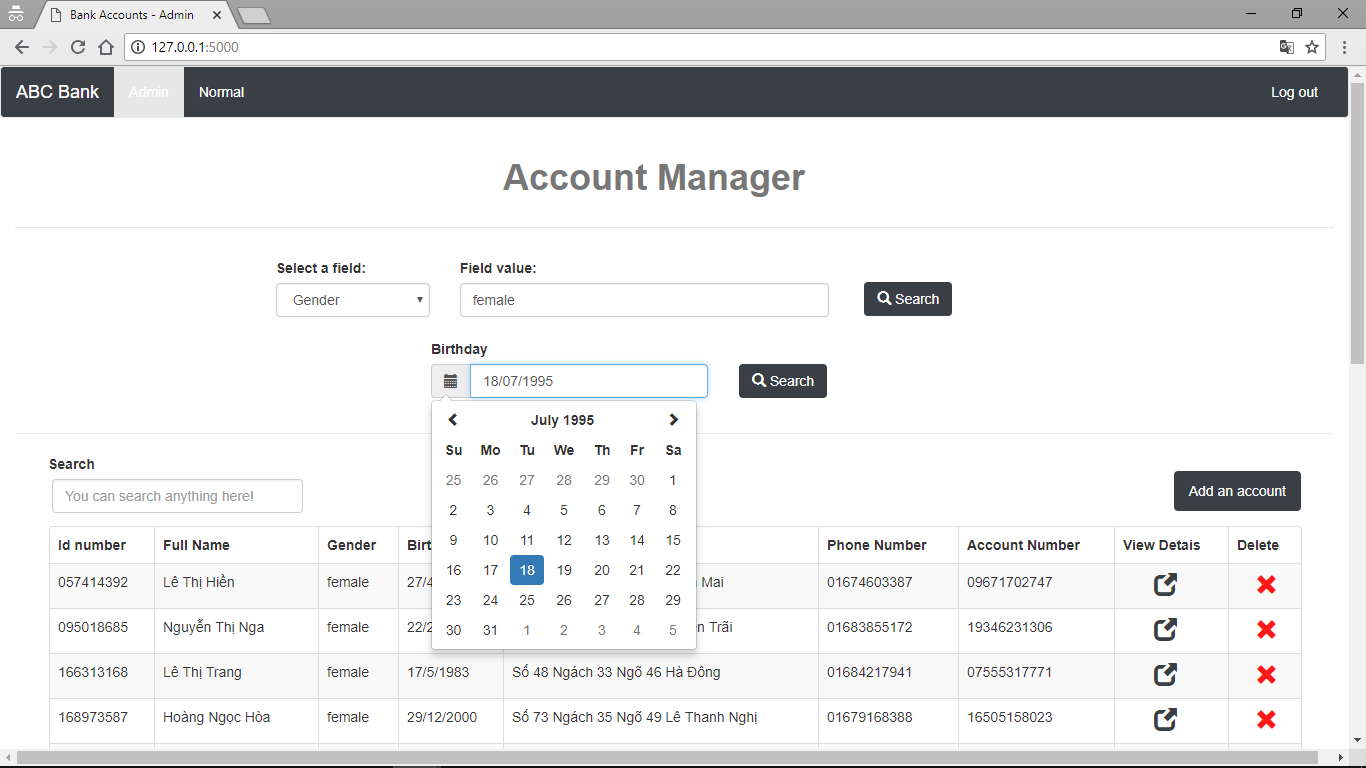
1. Search theo trường bất kỳ, chọn 1 trường muốn search và ghi nội dung sang ô input bên cạnh sau đó bấm search kết quả sẽ trả về ở bảng phía dưới(Mặc định web sẽ load tất cả các tài khoản). Kết quả khi search theo trường gender: female

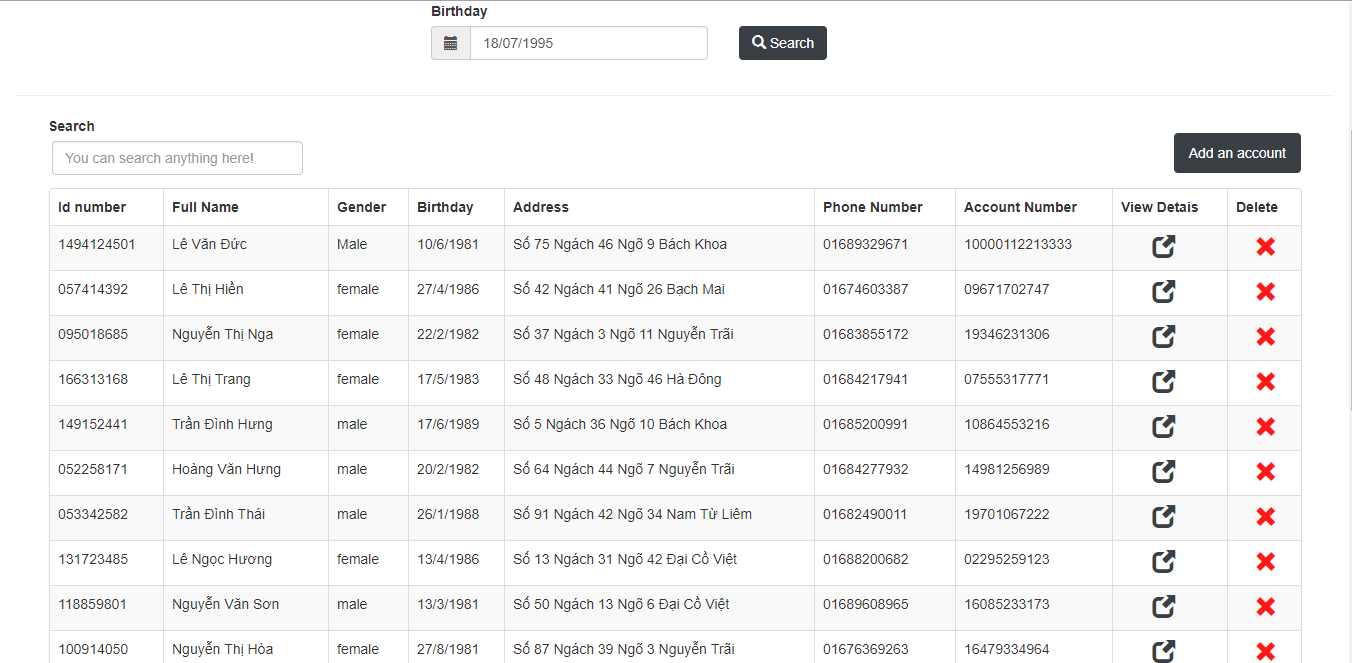




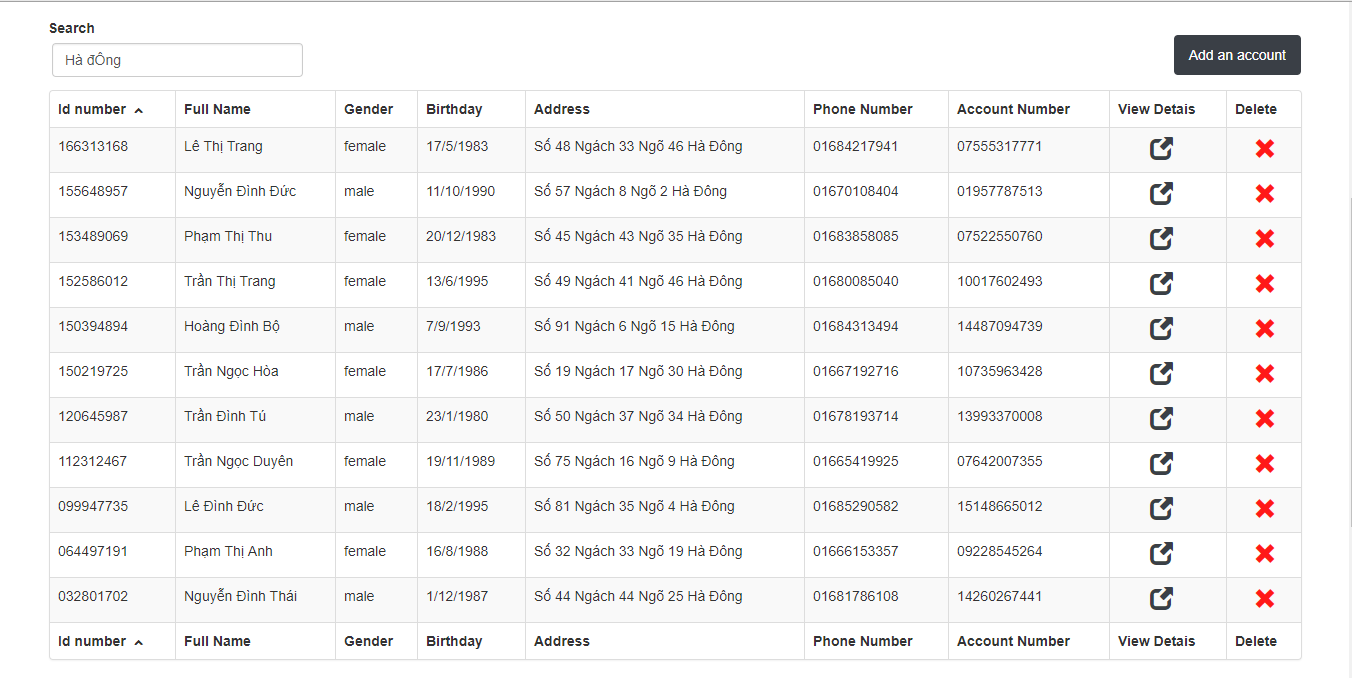
Kết quả tìm được trả về dạng bảng có phân trang.

1. Search theo trường birthday hỗ trợ dạng datetime picker, kết quả trả về sẽ là danh sách các tài khoản khách hàng có sinh nhật vào ngày đó hoặc những người có tuổi lớn hơn.(Xử lý trong API: lấy [current time – birthday] tính ra số ngày rồi lấy phần nguyên khi chia cho 365)





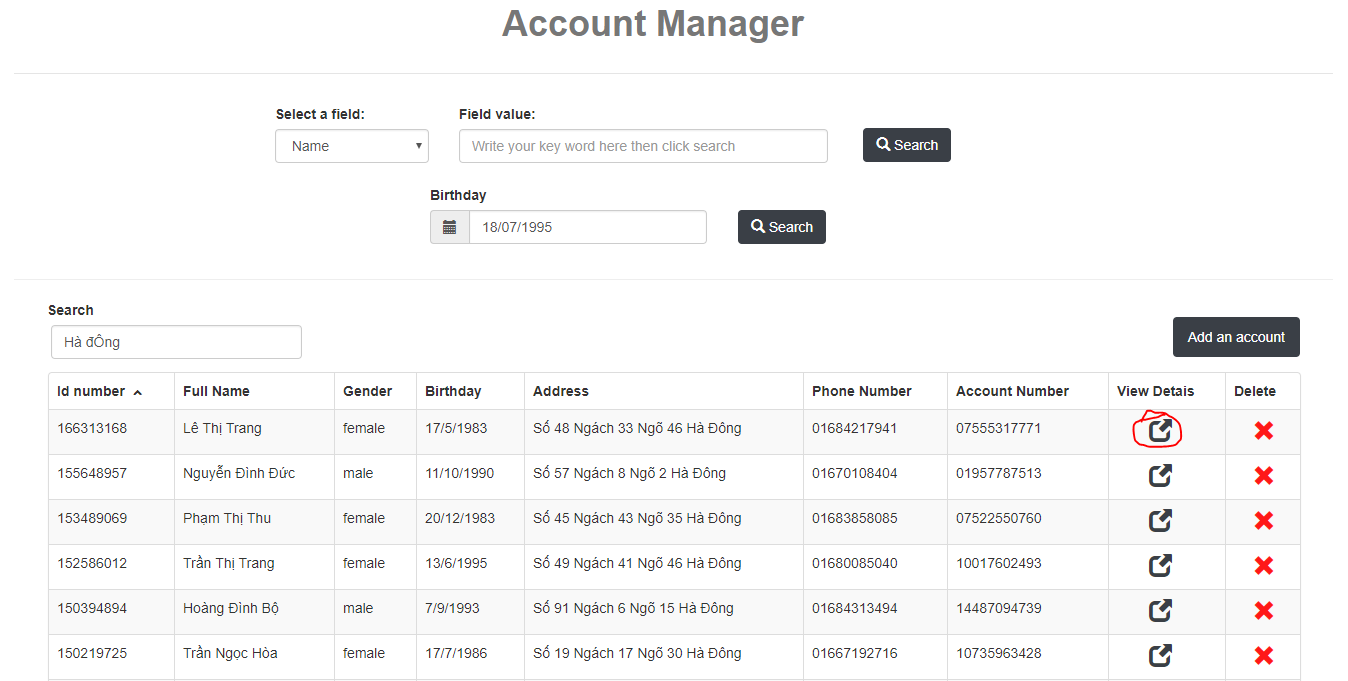
1. **­­­­** Haichức năng search đầu dựa trên API flask, chức năng search thứ 3 là search hỗ trợ client side, tìm kiếm nội dung bất kỳ và sắp xếp theo trường bất kỳ.

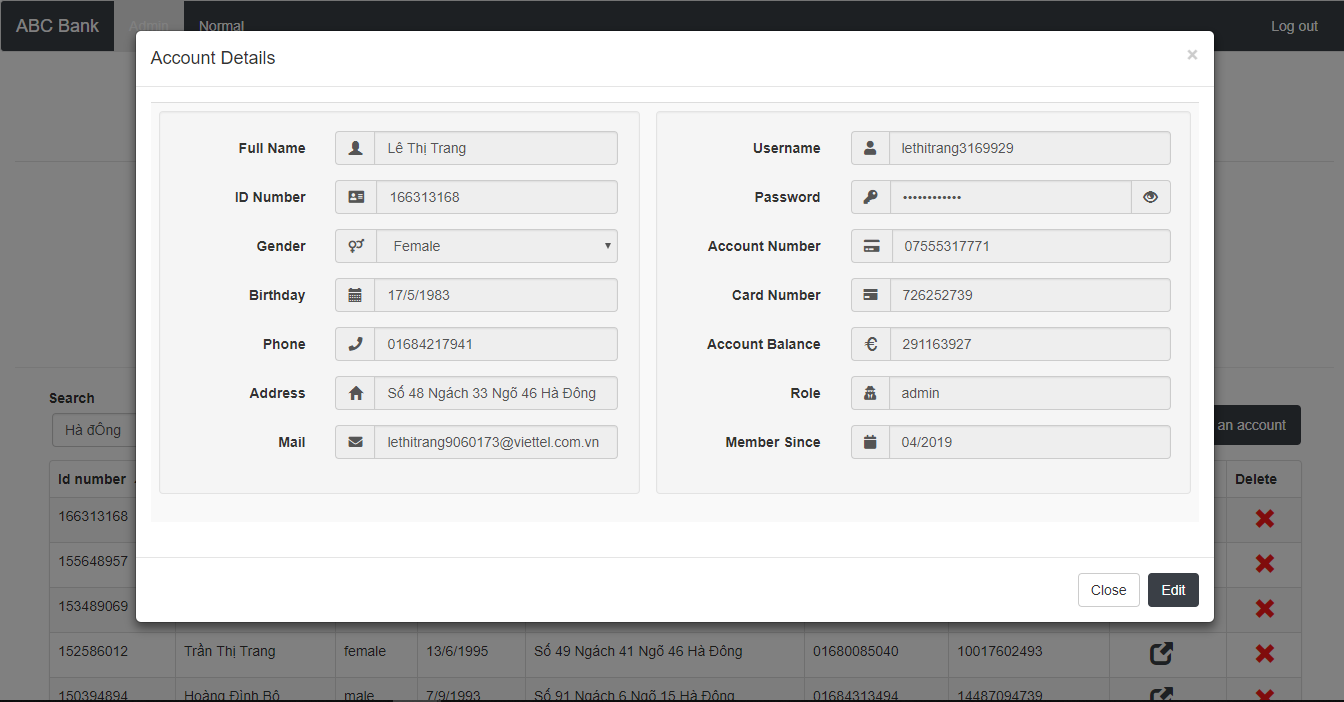


Click vào từng cột để thực hiện sort.

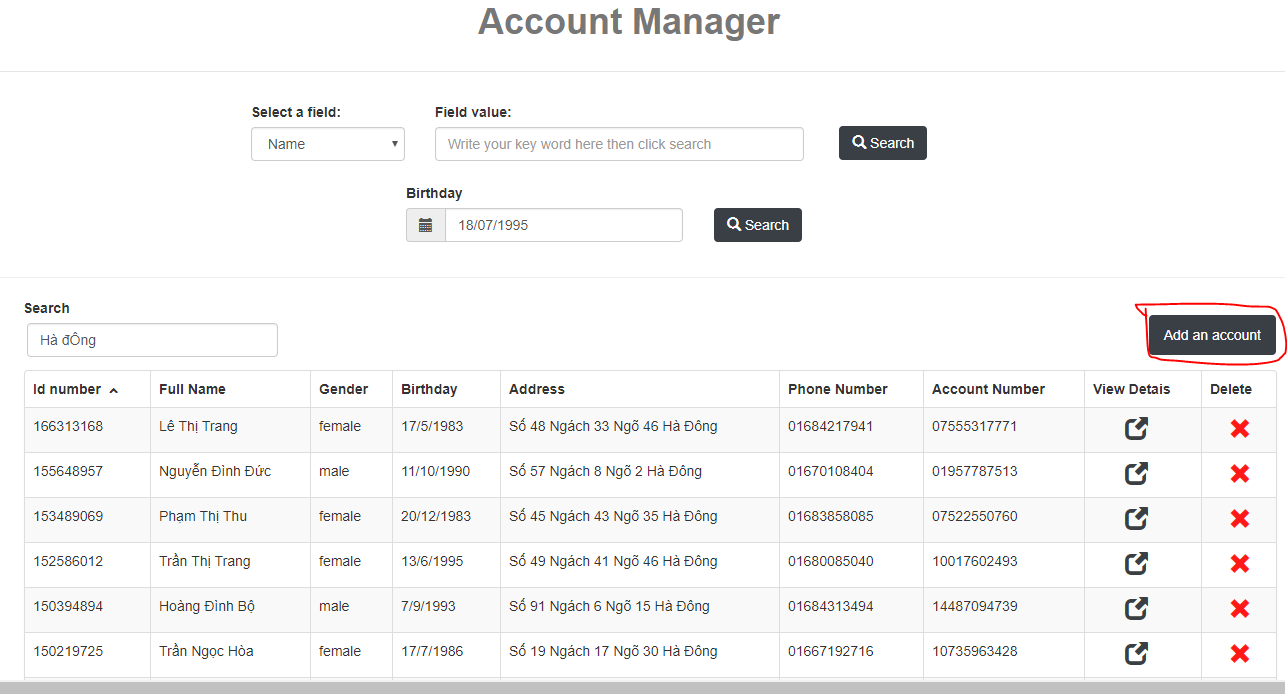
1. **Xem/Thêm/Sửa/Xóa**

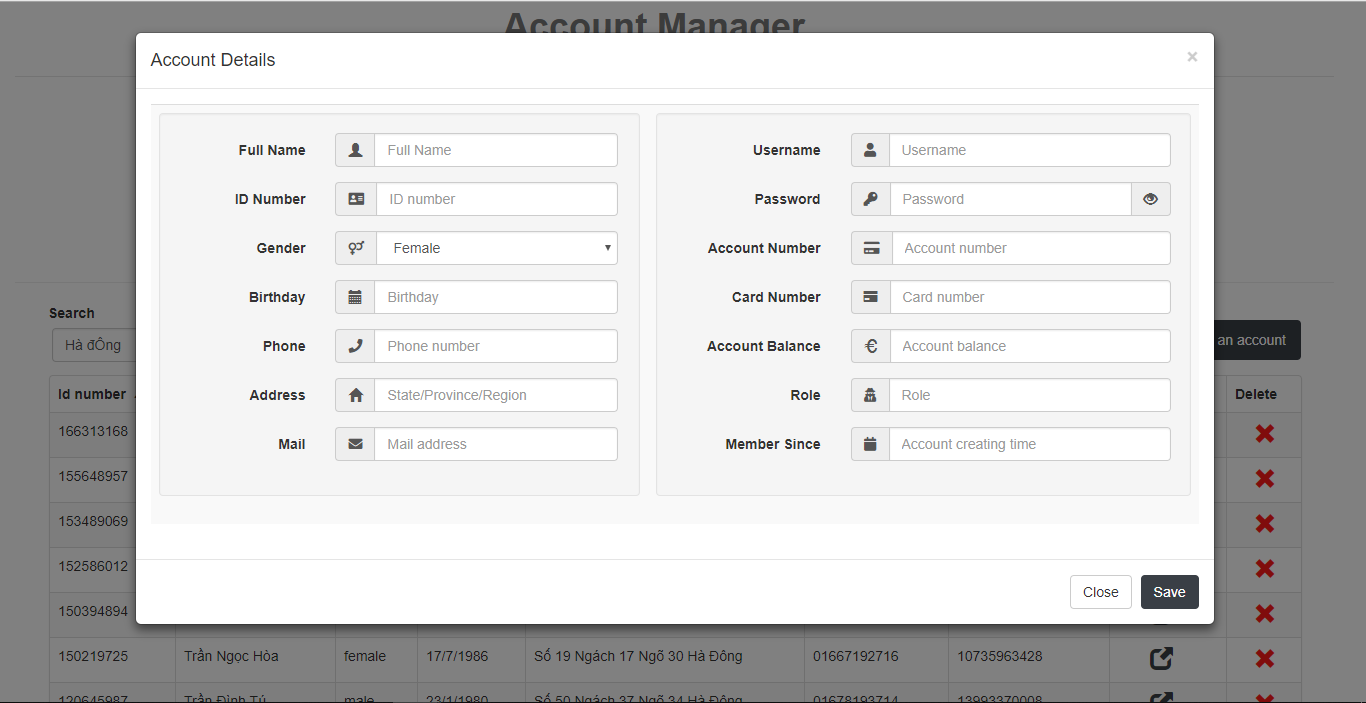
* **Xem**



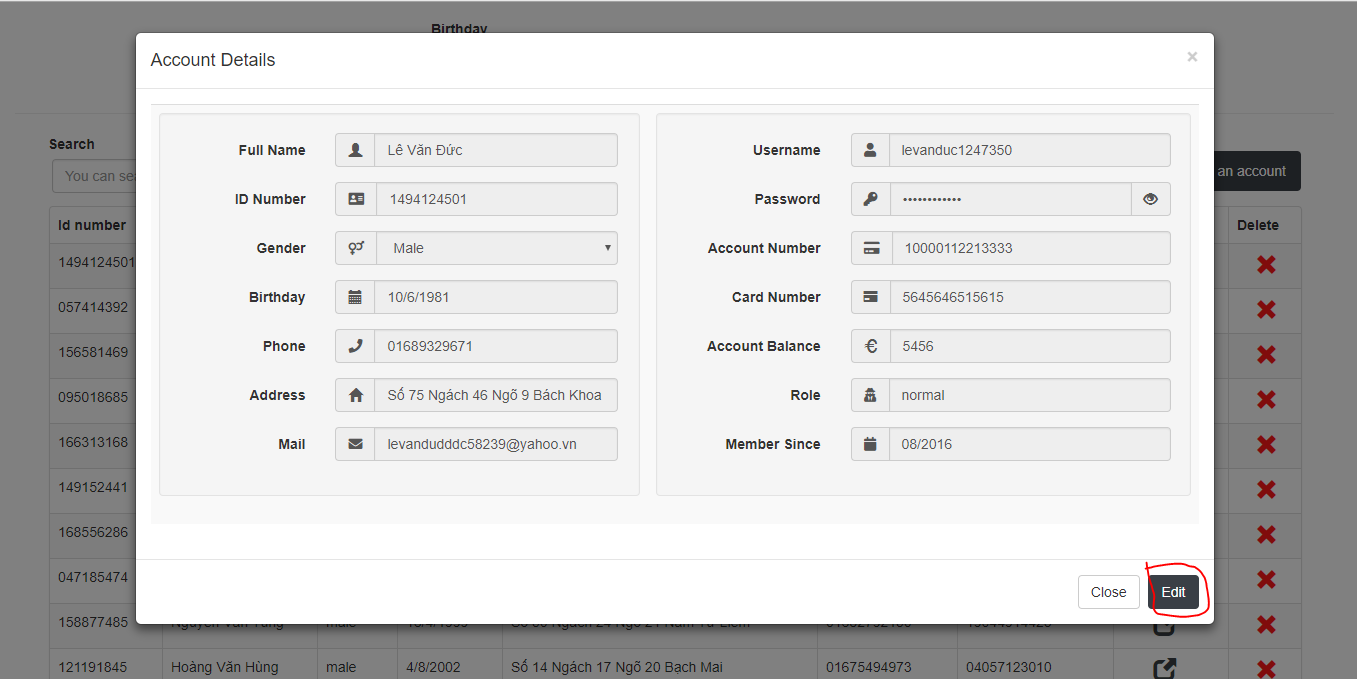


* **Thêm**

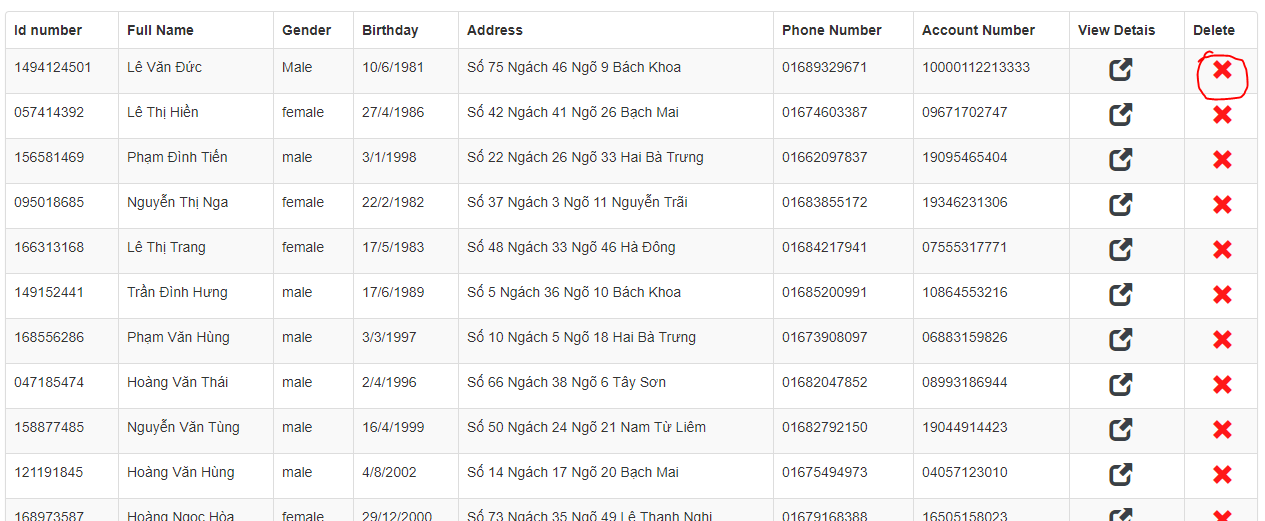


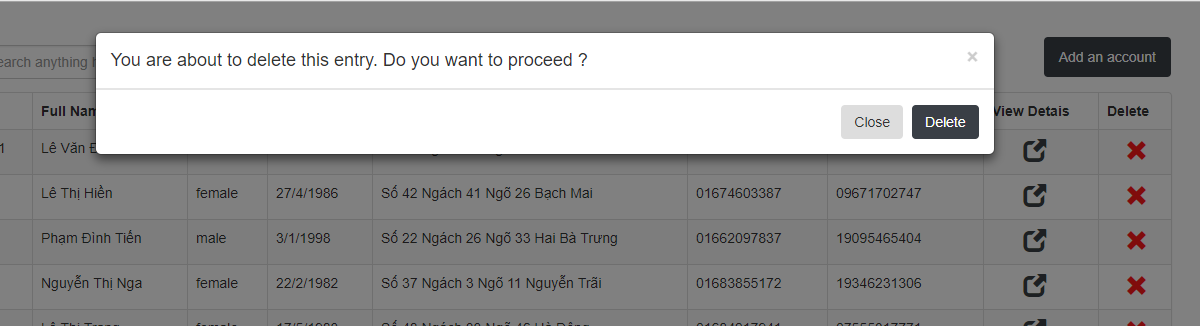


* **Sửa**



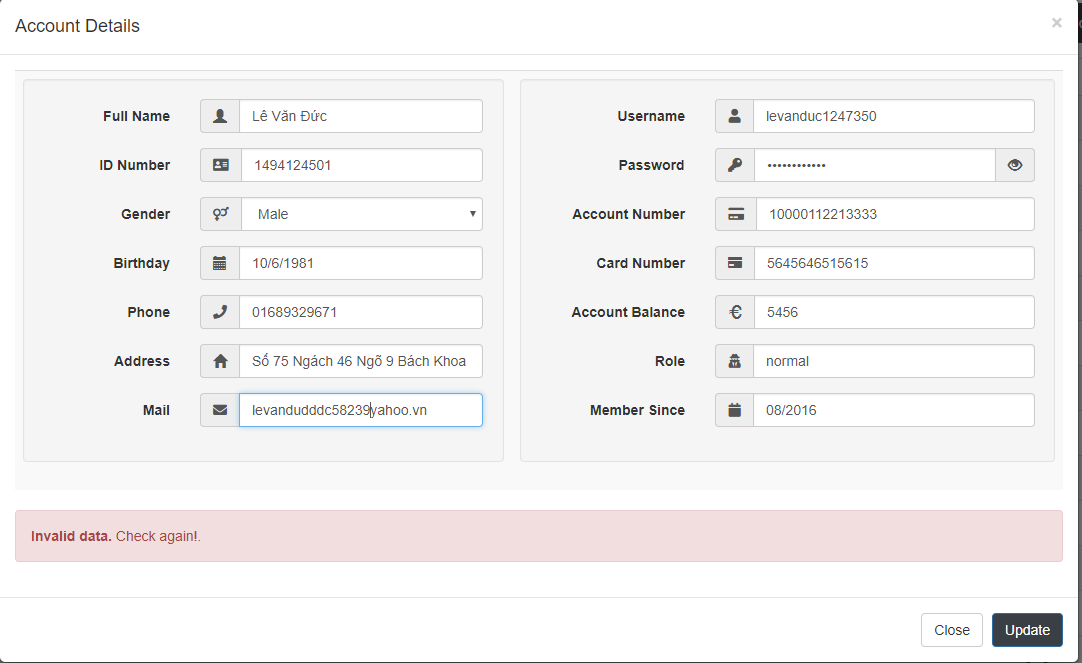
* **Xóa**

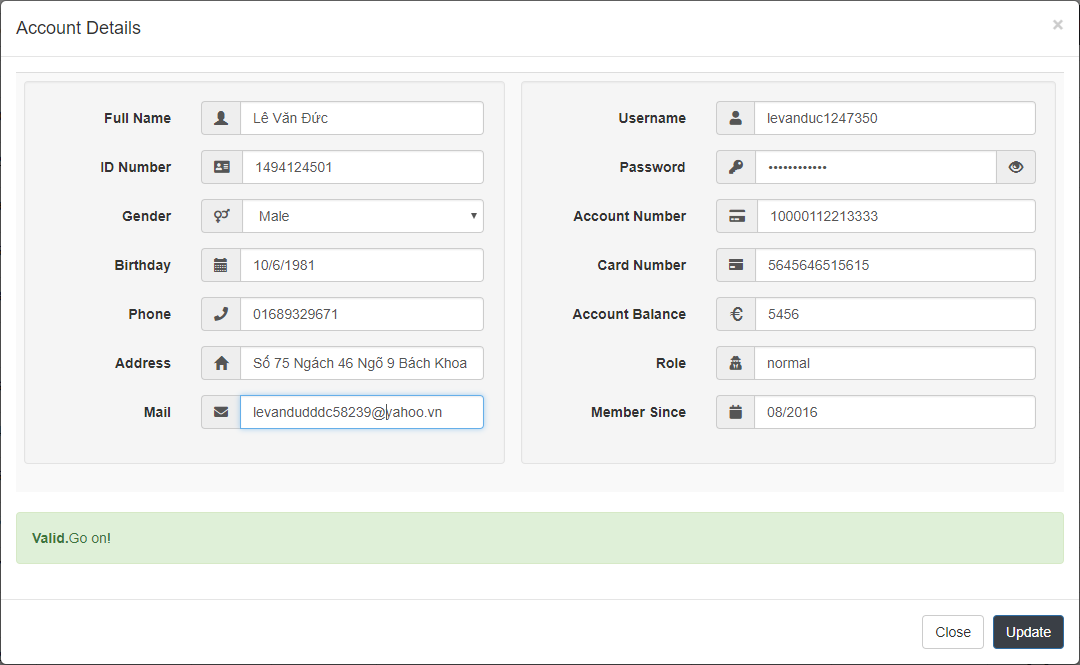


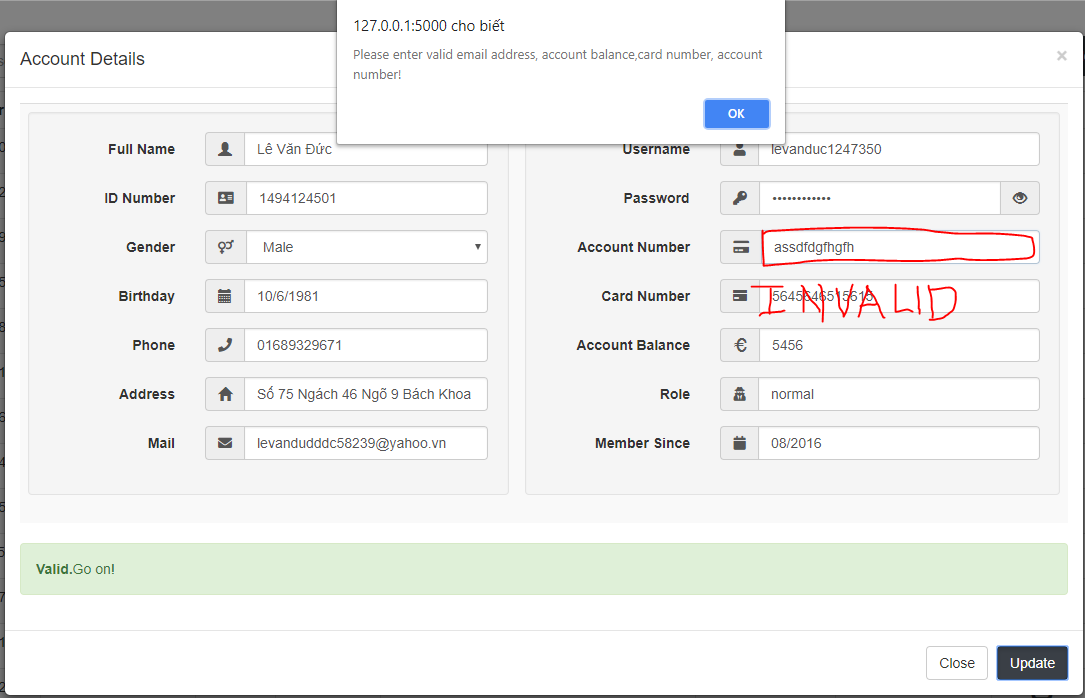


Khi thêm dữ liệu mới sẽ check các trường sau không được trùng nhau: số CMT *idNumber*, *mail*, *accountNumber*, *cardNamber* và *username*.

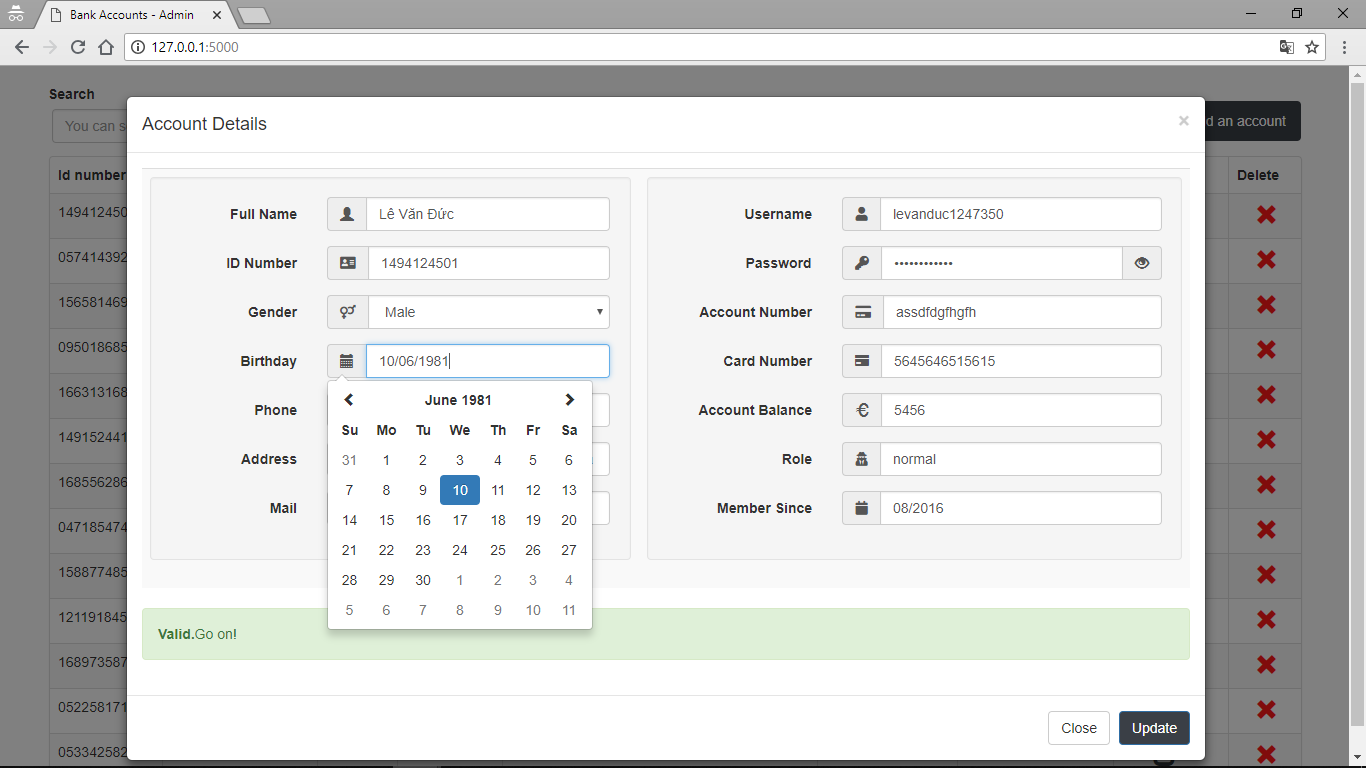
Validate các trường *idNumber*, *phoneNumber*, *cardNumber*, *accountNumber*, *accountBalance>0*, mail sử dụng regex. Dữ liệu sẽ trong bảng sẽ được update ngay sau mỗi thao tác nếu hợp lệ.



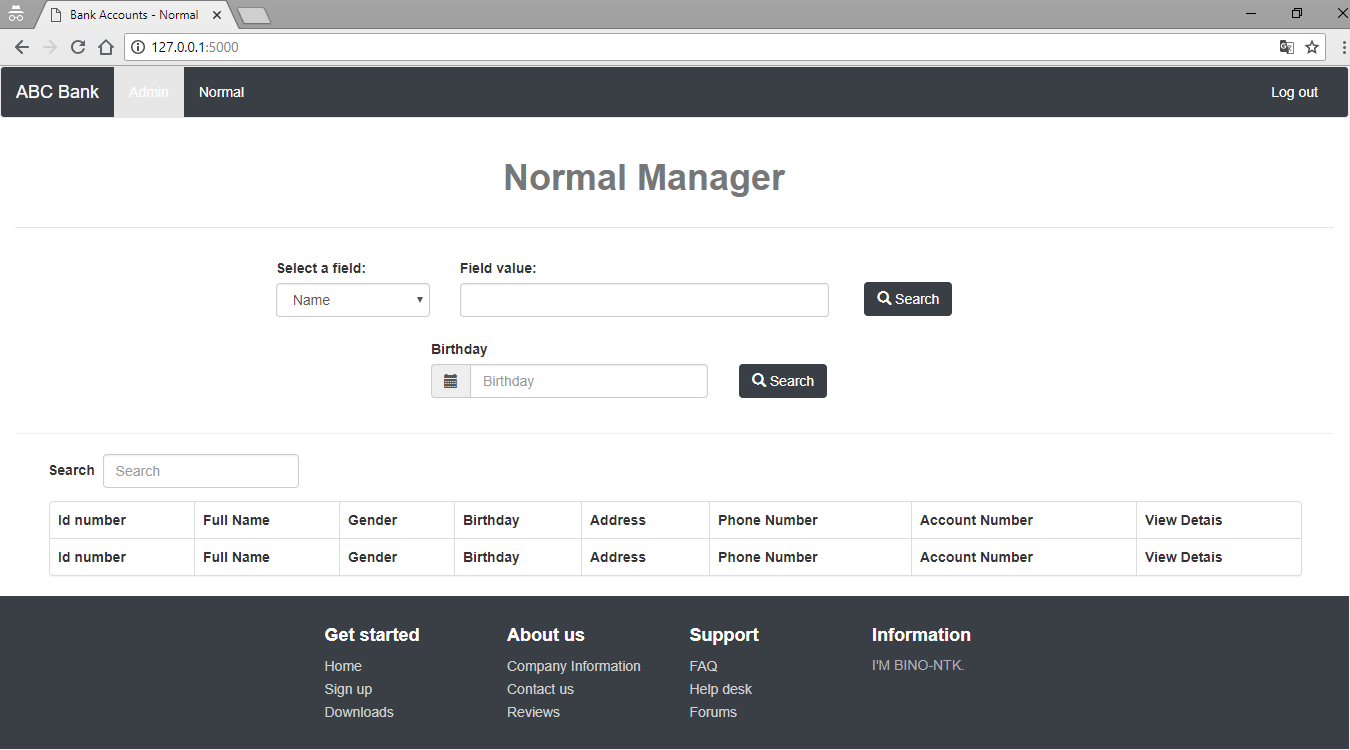




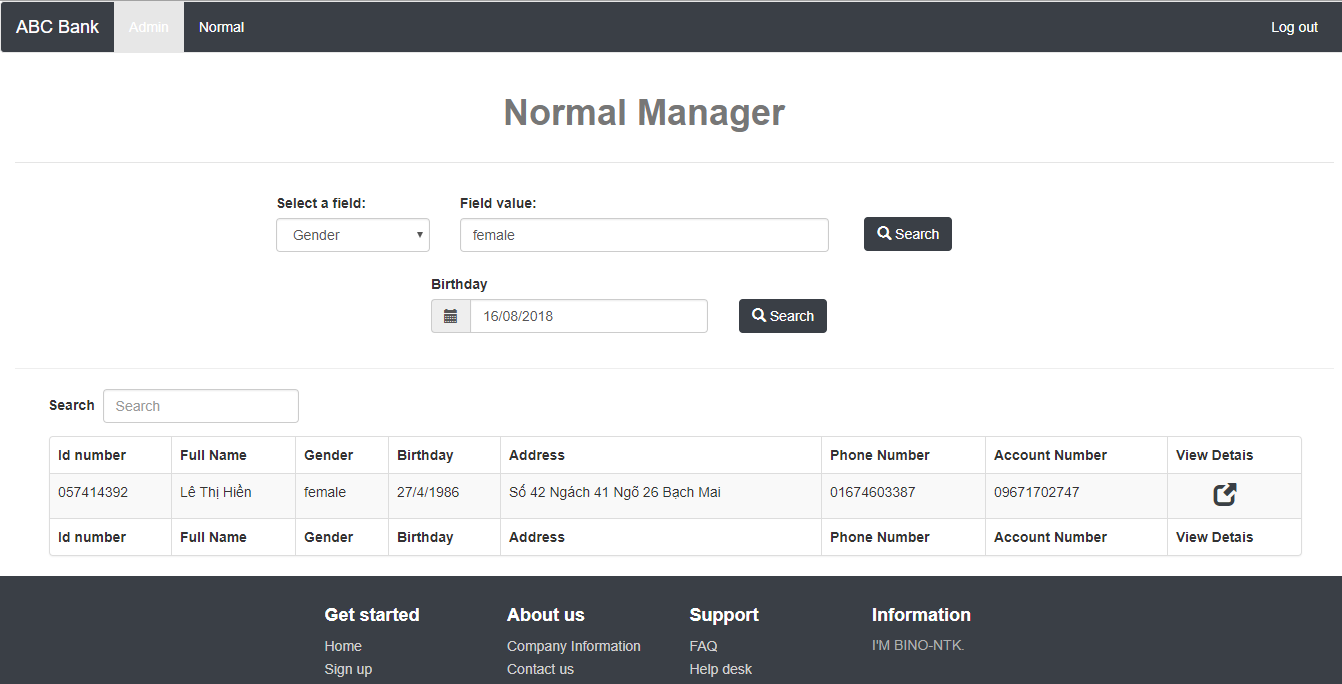
Thao tác update hay thêm sẽ không thành công(đã xử lý backend)



1. **Phân quyền(API&Portal)**



Giao diện khi đăng nhập bằng tài khoản normal.



1. **Paging(Client side)**

Phần paging server side đang bị lỗi