Account Lockout Policy

Backend:

In user model, we will store the attempts time of entering password, the last lock time and whether the use is locked:

```
  attempts: {
    type: Number,
    min: 0,
    max: 3,
    default: 0,
  },
  lockTime: {
    type: Date,
  },
  isLocked: {
    type: Boolean,
    default: false,
  },
```

The main principals of lockout policy are:

- If the user inputs the wrong password continously for three times, the account with the usename will be locked and the time of this moment will be recorded.
- If the account with the username is locked, the user will not be able to login.
- If the account with the username is locked and current time is one minute later than the recorded last lock time, the account will be unlocked and the attempts time will reset to 0.
- If the account with the username is not locked, and the user input the right password, the attempts time should be reset to 0.

And the algorithm of lockout is in code below:

```
const userLogin = async (req, res) => {
  ...

  if (user.isLocked) {
    if ((Date.now() - new Date(user.lockTime).getTime()) / 1000 > 6) {
      await User.updateOne({ name: req.body.name }, { attempts: 0 });
      await User.updateOne({ name: req.body.name }, { isLocked: false });
    } else {
      return res
        .status(429)
        .send({ success: false, msg: "Try again after 1 minute" });
```

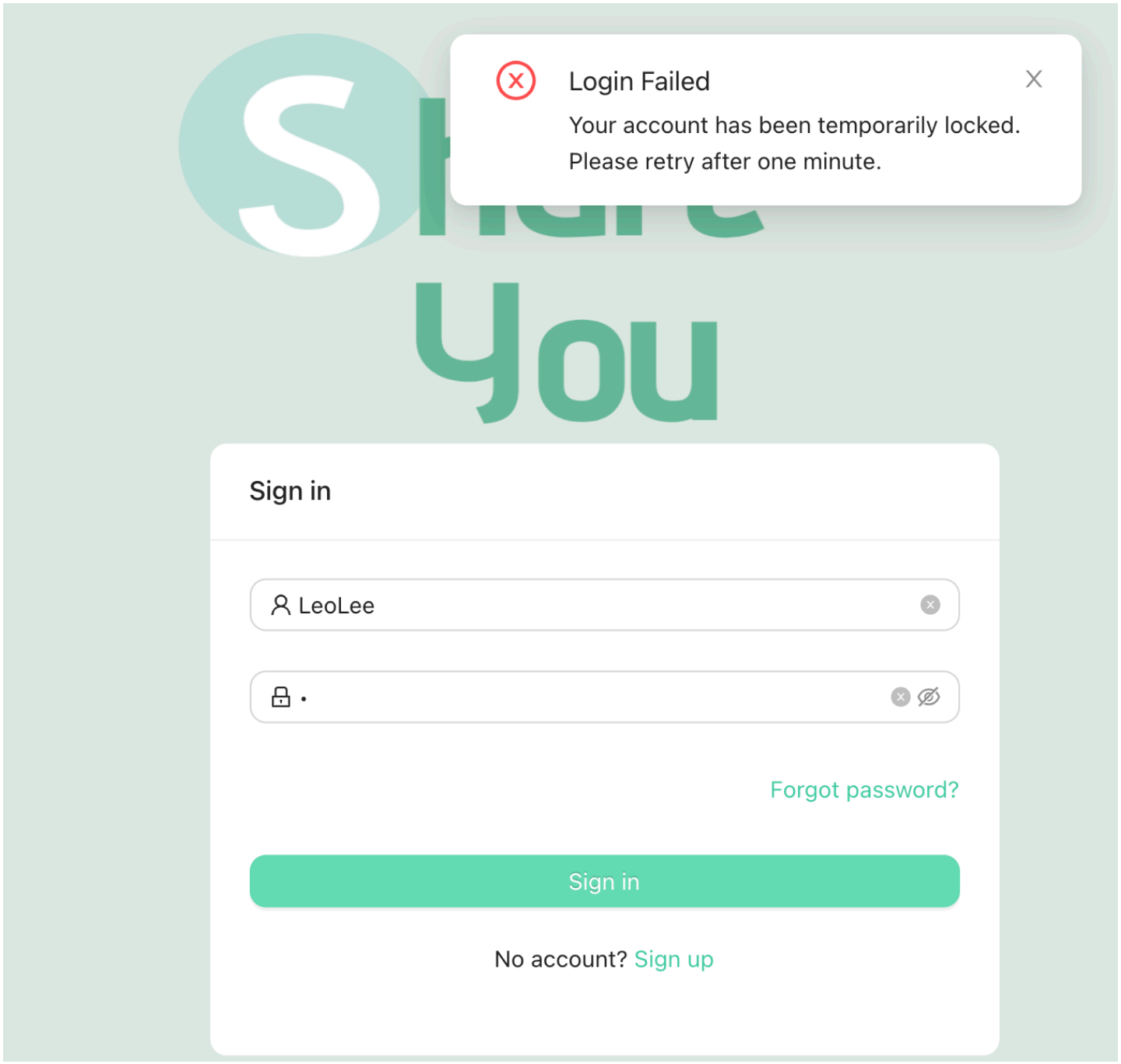
```
    }
  }
  const updatedUser = await User.findOne({ name: req.body.name });
  // Check if the password is correct
  const validPass = await bcrypt.compare(req.body.password, user.password);
  if (!validPass) {
    if (updatedUser.attempts !== 3) {
      await User.updateOne(
        { name: req.body.name },
        { attempts: updatedUser.attempts + 1 }
      );
      if (updatedUser.attempts !== 2) {
        return res
          .status(400)
          .send({ success: false, msg: "Invalid username or password" });
      }
    }
    await User.updateOne({ name: req.body.name }, { isLocked: true });
    await User.updateOne({ name: req.body.name }, { lockTime: new Date() });
    return res
      .status(429)
      .send({ success: false, msg: "Try again after 1 minute" });
  }

  ...

  await User.updateOne({ name: req.body.name }, { attempts: 0 });
  return res
    .status(200)
    .send({ success: true, msg: "Login Successful", token });
```

Frontend:

If user is locked, the message of "Try again after 1 minute" will show.


Login Failed
Your account has been temporarily locked.
Please retry after one minute.
Sign in
LeoLee
Forgot password?
Sign in
No account? Sign up