

# auth.service.js

```
import UsersModel from '../models/users.model.js'
import bcrypt from 'bcrypt'
import SessionModel from '../models/sessions.model.js'

const AuthService = {}

AuthService.login = async (username, password) => {
  try {
    let user = await UsersModel.findOne({ username })

    if (!user) throw new Error('User Not Found!')

    let matchPass = await bcrypt.compare(password, user.password)

    if (matchPass)
      return { _id: user._id, username: user.username, name: user.name }

    throw new Error('password NOT matched!')
  } catch (error) {
    console.log({ error })
    throw new Error('Failed To Login')
  }
}

AuthService.createSession = async (userID, userAgent) => {
  try {
    let session = await SessionModel.create({
      user: userID,
      userAgent: userAgent,
    })
    return session
  } catch (error) {
```

```

        throw error
    }
}

AuthService.checkSession = async (sessionID) => {
    try {
        let session = await SessionModel.findOne(
            {
                _id: sessionID,
                valid: true,
            },
            '_id'
        )
        .populate({ path: 'user', select: '_id name username' })
        .lean()

        return session ? session : new Error('Authentication Failed')
    } catch (error) {
        throw error
    }
}

export default AuthService

```