**Отчет по реализации аутентификации с использованием Passport.js**

В данном отчете описывается реализация системы аутентификации с использованием библиотеки Passport.js, которая позволяет пользователям входить в приложение через сторонние сервисы, такие как GitHub, GitLab и Яндекс.

**Архитектура**

**Используемые технологии**

* **Node.js**: серверная платформа для построения веб-приложений.
* **Express.js**: фреймворк для создания веб-приложений на Node.js.
* **Passport.js**: middleware для аутентификации в Node.js.
* **passport-github2**, **passport-gitlab2**, **passport-yandex**: стратегии для аутентификации через GitHub, GitLab и Яндекс соответственно.
* **Express-session**: middleware для управления сессиями пользователей.

**Реализация**

**Настройка сервера**

Сервер настроен с помощью Express.js и использует сессии для хранения информации о пользователе:

const express = require("express"); const session = require("express-session"); const app = express(); app.use( session({ secret: "sh", resave: false, saveUninitialized: true }) );

**Инициализация Passport**

app.use(passport.initialize());

app.use(passport.session());

**Стратегии аутентификации**

Для аутентификации через GitHub, GitLab и Яндекс используются следующие стратегии:

**GitHub**

passport.use( new GitHubStrategy( { clientID: "YOUR\_GITHUB\_CLIENT\_ID", clientSecret: "YOUR\_GITHUB\_CLIENT\_SECRET", callbackURL: "http://localhost:8000/auth/github/callback", }, (accessToken, refreshToken, profile, done) => { return done(null, profile); } ) );

**Яндекс**

passport.use(

new YandexStrategy(

{

clientID: "YOUR\_YANDEX\_CLIENT\_ID",

clientSecret: "YOUR\_YANDEX\_CLIENT\_SECRET",

callbackURL: "http://localhost:8000/auth/yandex/callback",

},

(accessToken, refreshToken, profile, done) => {

return done(null, profile);

}

)

);

**Сериализация и десериализация пользователя**

passport.serializeUser((user, done) => { done(null, user); }); passport.deserializeUser((user, done) => { done(null, user); });

**Маршруты**

Маршруты для аутентификации через каждый из сервисов реализованы следующим образом:

* **GitHub**:
  + /auth/github для начала аутентификации
  + /auth/github/callback для обработки колбэка после аутентификации
* **GitLab**:
  + /auth/gitlab для начала аутентификации
  + /auth/gitlab/callback для обработки колбэка после аутентификации
* **Яндекс**:
  + /auth/yandex для начала аутентификации
  + /auth/yandex/callback для обработки колбэка после аутентификации

**Главная стр.**

На главной странице отображается приветствие для аутентифицированного пользователя или ссылки для входа, если пользователь не авторизован:

app.get("/", (req, res) => {

if (req.isAuthenticated()) {

res.send(`<h1>Привет, ${req.user.username}!</h1><a href="/logout">Выйти</a>`);

} else {

res.send(`

<h1>Вы не авторизованы</h1>

<a href="/auth/github">Войти через GitHub</a><br>

<a href="/auth/gitlab">Войти через GitLab</a><br>

<a href="/auth/yandex">Войти через Яндекс</a>

`);

}

});  
  
**Завершение сессии**

Реализована возможность выхода из системы:

app.get("/logout", (req, res) => { req.logout(() => { res.redirect("/"); }); });