



SOLUTION APPROACH DOCUMENT

Azure AD Authentication with Node 16

Report Submitted

by

V HARSHA VARDHAN	(VTU17909)
K ABHILASH REDDY	(VTU17913)
P BALAJI	(VTU12423)
V MUKESH KUMAR	(VTU17918)
K VISHNU VARA PRASAD	(VTU16858)

Under the guidance of

Ms.. N. THAMIZHARASI	(Assistant Professor)
Ms. K. PREMA	(Assistant Professor)
Mr. R. PRIYADHARSHINI	(Assistant Professor)



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING**

**VEL TECH RANGARAJAN Dr. SAGUNTHALA R&D
INSTITUTE OF SCIENCE AND TECHNOLOGY**



ABSTRACT

Many organisations use multiple applications to do their daily tasks like Microsoft teams and outlook. It is time consuming and complicated to login every time we want to access that application. So to overcome this situation, This component is a web application to implement single sign-on(SSO) technology to allow the user to login and authenticate themselves once and access other applications without having to login for each application or service whenever they want to access that application or service. This project is to create a reusable component using Node.Js and Azure AD to authenticate the user. Using Passport.js as an authentication middleware we can provide support for Azure AD authentication and authorisation. Our project's main agenda is to create a reusable component which makes user authentication with Azure AD much more easier and efficient.

Keywords: Azure AD, Authentication NodeJS, Passport.Js, Reusable Component.



LIST OF FIGURES

Fig 2.1	Overall Workflow Diagram	7
Fig 2.2	Sequence Diagram	8
Fig 2.3	Class Diagram	9
Fig 2.4	Activity Diagram	10



ABBREVIATIONS

AzureAD	Azure Active Directory
SDK	Software Development Kit
SSO	Single Sign On
API	Application Programming Interface
CSS	Cascading Style Sheet
JS	JavaScript
HTML	Hyper Text Markup Language



TABLE OF CONTENT

ABSTRACT	ii
LIST OF FIGURES	iii
ABBREVIATIONS	iv
1 INTRODUCTION	
1.1 ABOUT THE DOCUMENT	
1.1.1 PURPOSE & SCOPE	1
1.1.2 OVERVIEW	1
2 COMPONENT DESIGN	
2.1 COMPONENT DESIGN DIAGRAM	
2.1.1 OVERALL WORKFLOW	7
2.1.2 SEQUENCE DIAGRAM	8
2.1.3 LOW LEVEL DESIGN	9
2.1.4 ACTIVITY DIAGRAM	10
3 TECHNOLOGY AND FRAMEWORK	
3.1 FRONT-END	
3.1.1 HTML,CSS	11
3.1.2 JAVASCRIPT	11
3.2 BACK-END	
3.2.1 NODE 16	11
3.2.2 AZURE AD	12
3.3 TOOLS USED	
3.3.1 VISUAL STUDIO CODE	12
4 SOLUTION APPROACH	
4.1 APPROACH DESCRIPTION	9



CHAPTER 1

INTRODUCTION

1.1 ABOUT THIS DOCUMENT

1.1.1 PURPOSE & SCOPE

- The main purpose of this document is to give an overview of the project to create a component to implement Azure Ad authentication using Node. This document covers the objective of project, technologies & frameworks used and approach to be used.
- The scope of building this document is to give a detailed explanation on project objectives, technologies & framework used and approach to be used.

1.1.2 OVERVIEW

Azure AD authentication with Node 16 is a Web application created using Node 16 and Azure AD that authenticates the user. This project uses a secure authentication and authorization framework Azure AD and implements Single Sign On for an easy and seamless experience for the user. The primary goal is to make the user authentication process easier and to reduce time and make the authentication much more efficient.



CHAPTER 2

COMPONENT DESIGN

2.1 COMPONENT DESIGN DIAGRAM

This section describes the component design and the specific way of designing.

2.1.1 OVERALL WORKFLOW

This overall workflow diagram describes the complete working process of the component. Initially, Azure AD will check for existing access tokens and based on existing token Azure AD will provide access to the service application if there are no existing access tokens the Azure AD will ask for the user to login using their credentials and if the credentials are valid, Azure AD will save the access tokens and gives authorization to the service application

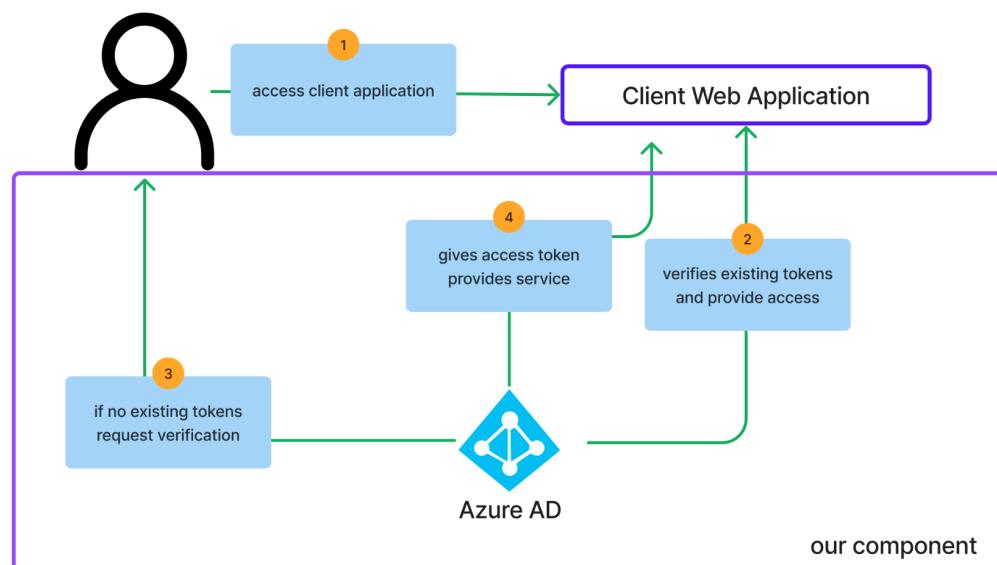


Fig 2.1 : Overall Workflow Diagram



2.1.2 SEQUENCE DIAGRAM

This sequence diagram explains the sequence of actions performed in this component. When the user tries to access the service application, Azure AD will check for the existing access tokens and if there are any existing access tokens Azure will provide access to application else Azure will ask for the user to authenticate using their Azure AD credentials and then if the credentials are valid, then Azure AD will provide service to the end user based on the access tokens.

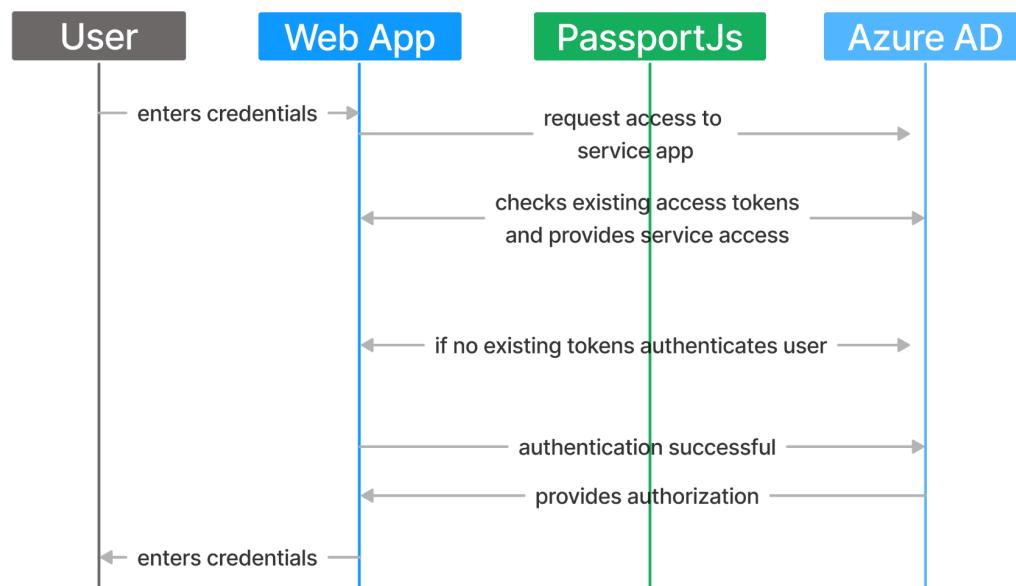


Fig 2.2: Sequence Diagram



2.1.3 LOW LEVEL DESIGN

This class diagram represents the attributes and operations/functions made by each class. The user is an entity with the id, username, mail, password, and accessToken as the attributes. Passport.JS is the Middleware between nodeJS and Azure AD to verify or authenticate users using the methods authenticate() and authorize(). Then the Azure AD is the cloud platform that contains the methods such as init(), login(), logout() and ensureloggedin() to initialise the process, login the user, logout the user and ensure that the user is logged in.

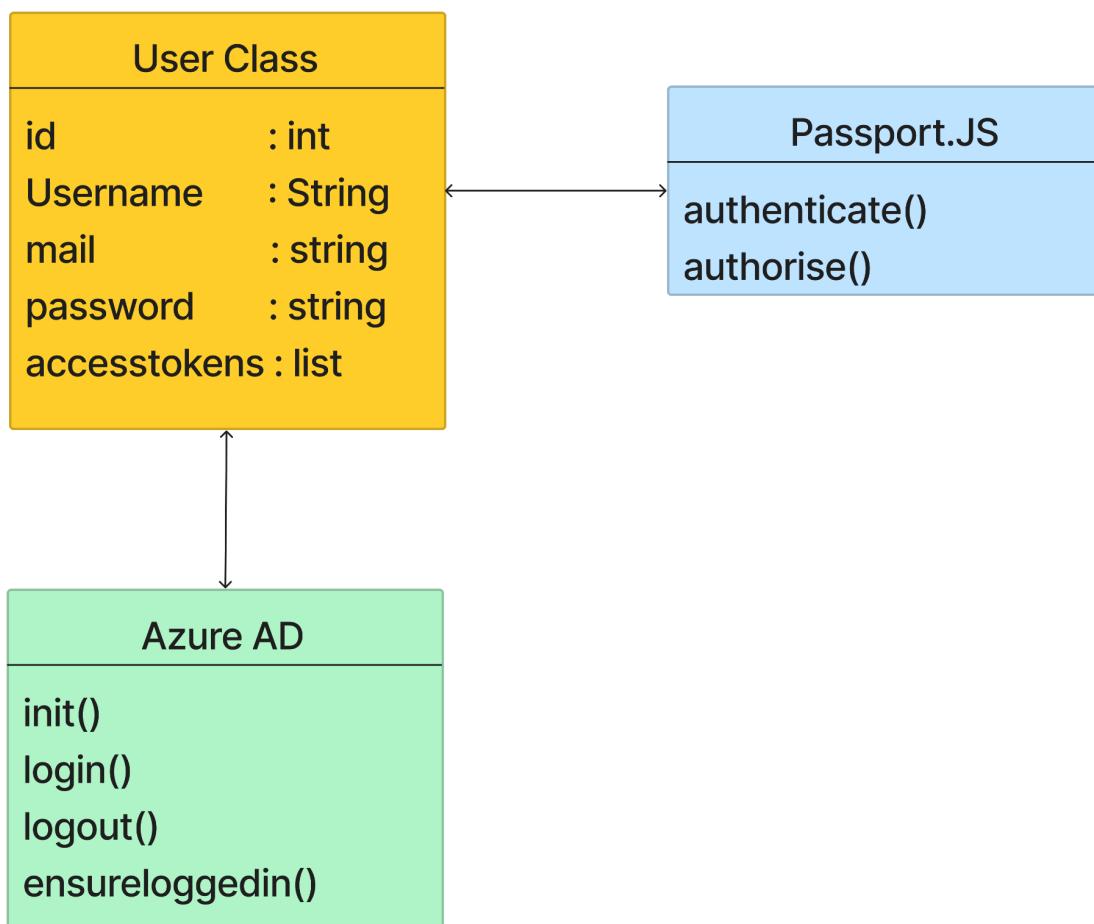


Fig 2.3: Class Diagram

2.1.4 ACTIVITY DIAGRAM



The activity diagram shows how the activity process starts and ends in the web application. Activities of each module are mentioned.

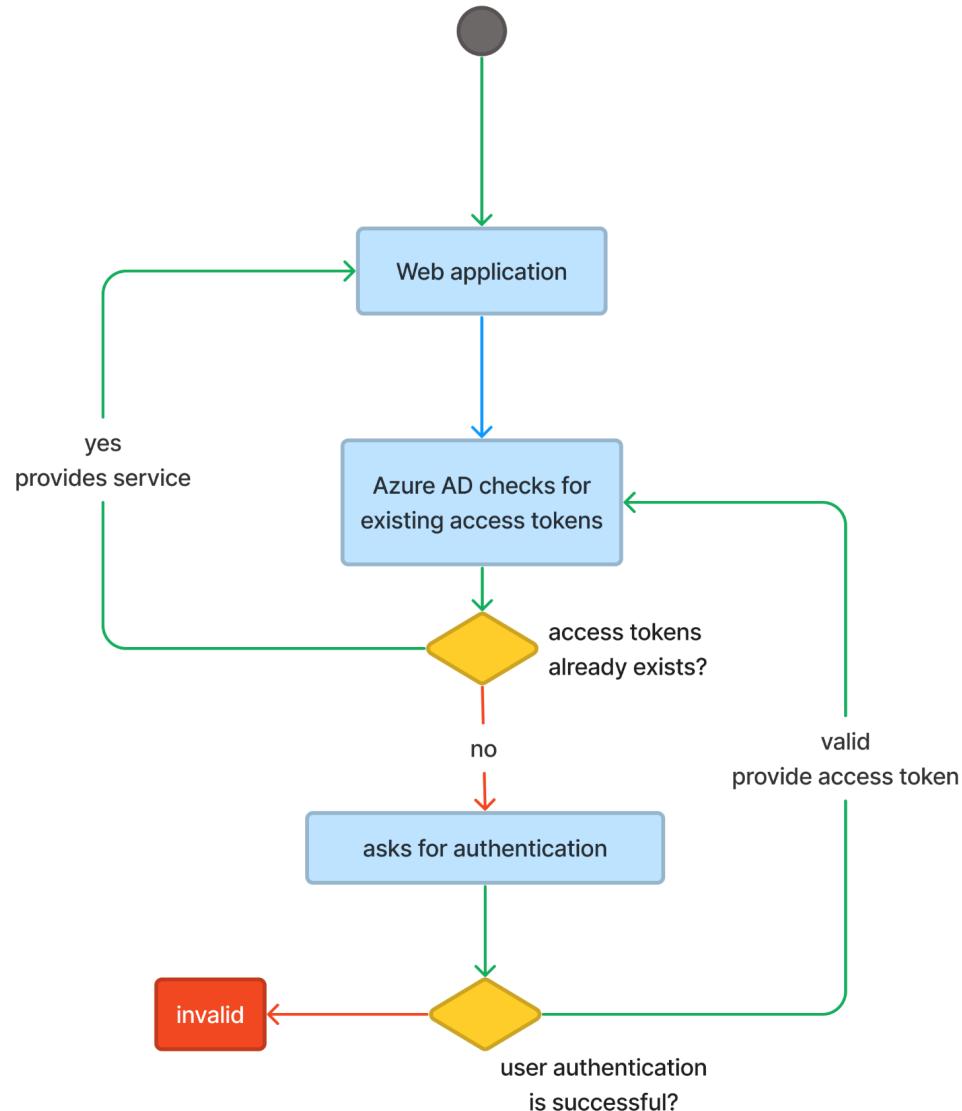


Fig 2.4: Activity Diagram

CHAPTER 3

TECHNOLOGY & FRAMEWORK



3.1 FRONT-END

3.1.1 HTML, CSS

- HTML is Hyper text markup language and CSS is Cascading Style Sheet
- HTML and CSS are used to design web applications and style them

Purpose

- By using HTML and CSS, web pages of our application are designed

3.1.2 JavaScript

JavaScript is a programming language that is used to make the web application dynamic and interactive

3.2 BACK-END

3.2.1 NODE.JS

Version 16

- Node JS is a open source server environment that can run on all platforms such as MacOs, Windows and Linux
- Node JS is a backend JavaScript environment and executes JavaScript code outside a web browser

Purpose

- Node.JS will be used as primary backend-technology for the building of server side application

3.2.2 AZURE AD

- Azure AD stands for Azure Active Directory which is a cloud based identity and access management service.



- Azure AD provides a wide range of authentication options including phone number authentication, mail authentication

Purpose

Azure AD is used to manage user authentication and authorization providing a secure way to protect the web application

3.3 TOOLS USED

3.3.1 VISUAL STUDIO CODE

- Visual studio code is a code editor with all language support functionality. It supports development processes like debugging, version control, and testing.
- No separate framework or packages are needed, it can automatically plugin all libraries and frameworks.

CHAPTER 4

SOLUTION APPROACH

4.1 APPROACH DESCRIPTION



This section provides a detailed view of how the web application works and how every technology is used for authentication and authorization::

1. Workflow diagram briefs about the working process of the web application.
2. For web page creation, HTML, CSS and JavaScript are used to make the component dynamic and interactive.
3. Node Js will be used to create a server side application
4. Azure AD will be used for authentication and authorisation of the user and to verify access tokens and to provide service based on the access token.
5. Passport.JS is a middleware between nodeJs and Azure AD for the authentication of the user based on the credentials entered by the user or the access tokens.
6. A sample test application will be created and used to test the component.
7. It undergoes unit and functional testing. Then detect the issue caused and fix sonar, SAST, and DAST issues.
8. The test scenarios and expected behaviour of the component will be mentioned below later.