# Introduction

Building a secure authentication system from scratch is complex, time-consuming, and prone to security risks. This project demonstrates a modern approach by leveraging IBM App ID for secure, scalable, and easy-to-integrate authentication in web applications.

# Problem

Developers need a reliable way to manage user identities, secure passwords, and control access to their applications without becoming security experts.

# Project Objective:

To build a functional web application that delegates all authentication tasks to **IBM App ID**, a professional Identity as a Service (IDaaS) platform, to demonstrate a modern and secure development approach.

## Proposed Solution: IBM App ID

This project uses **IBM App ID** to handle all aspects of user authentication.

- **What is it?** A managed cloud service for adding authentication and authorization to web and mobile apps.
- **Why use it?**
  - **Enhanced Security:** Leverages IBM's security expertise to protect user credentials and sessions.
  - **Rapid Development:** Eliminates the need to write complex code for login, registration, and password management.
  - **Scalability:** Easily handles a growing number of users without infrastructure changes.
  - **Feature-Rich:** Supports social login (Google, Facebook), multi-factor authentication (MFA), and user profiles out of the box.

## System Architecture

The application follows a simple and secure authentication flow.

1. **User Request:** The user visits the Node.js application and tries to access a protected page.
2. **Redirect to App ID:** The application redirects the user to a secure login page hosted by IBM App ID.
3. **User Authentication:** The user signs in or registers on the App ID page.
4. **Token Issuance:** App ID verifies the user and sends an authorization token back to the Node.js application.
5. **Access Granted:** The application validates the token and grants the user access to the protected page.

# Technology Stack

This project was built using a modern set of tools and services.

- **Cloud Platform: IBM Cloud**
- **Authentication Service: IBM App ID**
- **Backend Runtime: Node.js**
- **Web Framework: Express.js**
- **Authentication Middleware: Passport.js**
- **Version Control: Git & GitHub**

## Implementation: The Core Integration

The connection between the Node.js app and IBM App ID is configured using the Passport.js middleware.

- This small block of code is the heart of the integration.
- It tells our application all the necessary details to communicate securely with the App ID service.

```
// server.js - Configuring the App ID Strategy
passport.use(new WebAppStrategy({
    tenantId: "YOUR_TENANT_ID",          // Identifies our App ID instance
    clientId: "YOUR_CLIENT_ID",         // Identifies our specific application
    secret: "YOUR_SECRET",              // A secret password for our application
    oauthServerUrl: "YOUR_OAUTH_URL",  // The App ID login server URL
    redirectUri: "http://localhost:3000/appid/callback" // Where to return after login
}));
```