# Implementation of Google Authentication

Rangana Bandara

Reg.no:2337

**Importance of Authentication**: Protect user data and resources.

**What is Google Authentication?**: A method that allows users to sign in using their Google account.

## JWT (JSON Web Token)

:is a standard for securely transmitting information between parties as a JSON object. It's commonly used for authentication and authorization in web applications.

- **Structure:**
  - o **Header:** Contains metadata about the token, such as the algorithm used to sign it.
  - o **Payload:** Contains claims about the user, such as their ID, email, and roles.
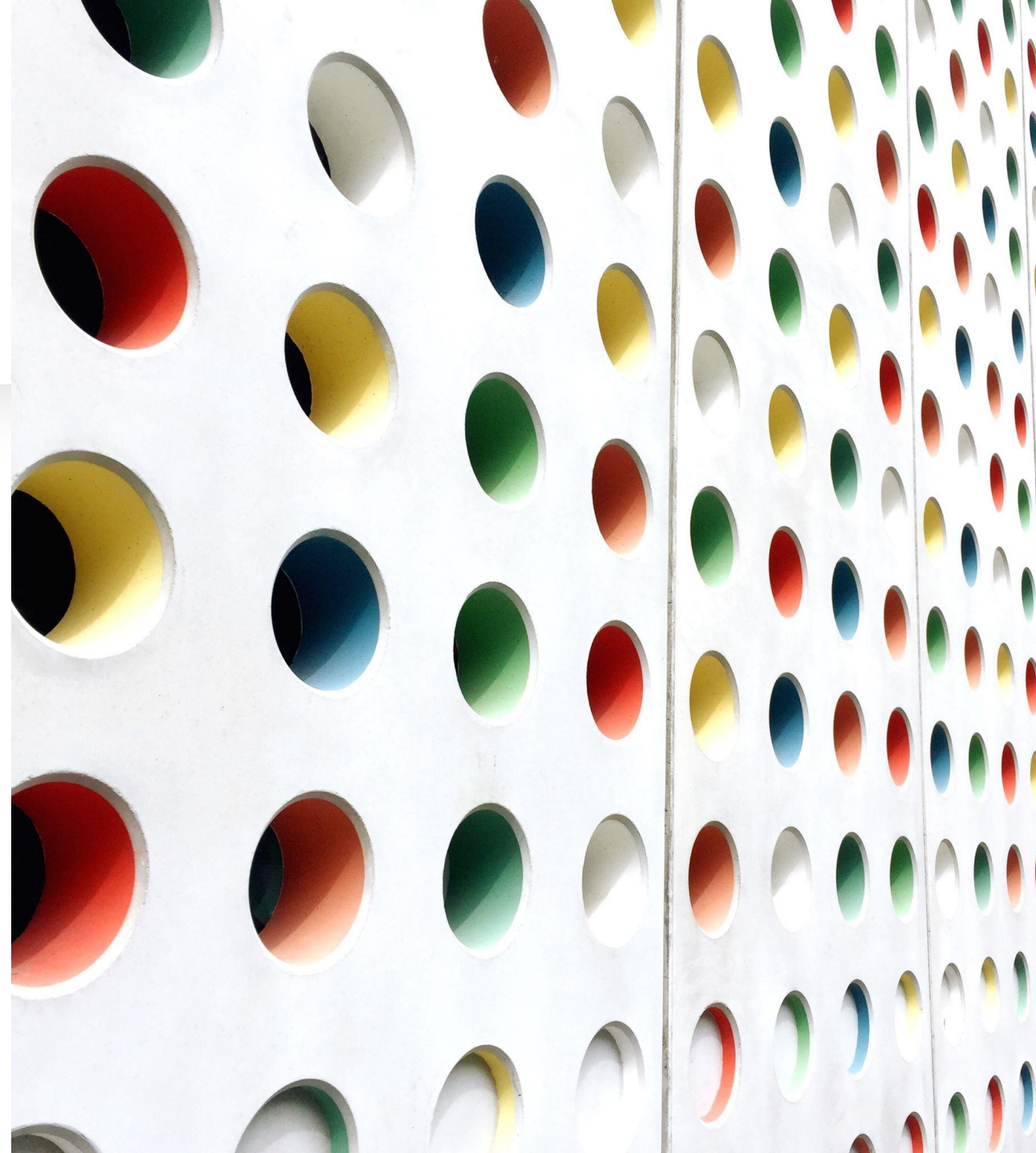  - o **Signature:** Ensures the token's integrity and authenticity.

# Dependencies and Libraries

- **JavaScript (Node.js):**

- `passport-google-oauth20`: A Passport.js strategy for Google authentication.

- `google-auth-library`: A Node.js library for interacting with Google APIs.

- **Python:**

- `google-auth-oauthlib`: A Python library for OAuth 2.0 and OpenID Connect.

- `google-auth-httplib2`: A Python library for interacting with Google APIs using HTTPlib2.

# Cont.

- **Java:**

- `google-api-services-oauth2-v2-rev517-1.23.0.jar`: A Java library for the Google OAuth 2.0 API.

- **C#:**

- `Google.Apis.IdentityToolkit.v3`: A C# library for the Google Identity Toolkit.

- `Google.Apis.Oauth2.v2`: A C# library for the Google OAuth 2.0 API.

# Library for Html

- <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script> <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-auth.js"></script>

# Implementation

Setting Up Google API

Create Google Cloud Project

Enable APIs: Google+ API or Identity services.

Create OAuth 2.0 Credentials: Obtain Client ID and Secret.

Redirect URI: Configure where to send users after authentication

Handling Backend Logic

# Conclusion

**RECAP**: BENEFITS OF GOOGLE AUTHENTICATION IN MODERN APPLICATIONS.

**ENCOURAGEMENT**: ADVOCATE FOR ADOPTING SECURE AUTHENTICATION METHODS.