# Project: AES Rijndael (encryption and decryption)

## Team profile

Our team consists of two students:

**Murzaly Aray** - creativity, stress tolerance, sociability, logical thinking, knowledge of programming languages such as: Python, Java, C ++, HTML, CSS, Java script. Ability to work with databases, knowledge of SQL, knowledge of Figma.

**Kentbayev Arthur** – Full Stack Web Developer. Strong sides: logical thinking, responsibility. Knowledge of such programming languages: Java, Python, C ++, PostgreSQL, HTML, CSS, JavaScript. Experience in working with bootstrap framework, Vue js, Node js, Jekyll.

## Introduction. Purpose, Scope and Objectives

## In the second year of the second semester, we were introduced to a new subject, Information security. We have studied many different interesting ciphers. We studied the strengths and weaknesses of different ciphers. We also went through block encryption, which is called RSA. We found this type of encryption more reliable than others and decided to take the AES theme. The main thing in our project is to implement this cipher, in the form of a website in which any user can easily understand and get the desired result. A person will write down his text and key in text field. In our work, we will use Visual Studio Code.

## Project description

## Advanced Encryption Standard, also known as Rijndael — is a symmetric block encryption algorithm adopted as an encryption standard by the US government based on the results of an AES competition. This algorithm is well analyzed and is now widely used, as was the case with its predecessor DES.

The algorithm converts one 128-bit block to another using the secret key that is needed for this conversion. To decrypt the resulting 128-bit block, a second conversion with the same secret key is used.

The block size is always 128 bits. The key size is also fixed. To encrypt any text with any password, you can do the following:

* get a hash from a password
* convert a hash to a key according to the rules described in the AES standard
* split text into 128-bit blocks
* encrypt each block with the cipher function

Of course, the text length may not be a multiple of 128 bits. In such cases, you can add zeros to the text to the desired length, and add several bytes to the encrypted data with the encrypted size of the original text.

We will write the application in javascript. We will create a website with a nice and user friendly interface using bootstrap 4, html, css. The site will contain a detailed description of the cipher and a detailed output of each cipher action. There will be two text fields. The first text field is plaintext, and the second is the user's key. There will be two main buttons Encrypt and Decrypt.

After clicking on one of these buttons, the user will see the result and a detailed output with a description of each action.