Prajwalhonashetti143@gmail.com

Assignment 1: Create a JavaScript object representing a user with properties for name, email, and age. Write functions that manipulate this object, such as changing the name, updating the email, and calculating the user's birth year.

```
let user = {
   name: "Rohit Sharma",
   email: "rohit@example.com",
   age: 32
};

function changeName(newUser, newName) {
   newUser.name = newName;
}

function updateEmail(newUser, newEmail) {
   newUser.email = newEmail;
}

function calculateBirthYear(newUser) {
   const currentYear = new Date().getFullYear();
   return currentYear - newUser.age;
}
changeName(user, "Rohit Kumar");

updateEmail(user, "rohit.kumar@example.com");
const birthYear = calculateBirthYear(user);

console.log("Updated User:", user);
console.log("Birth Year:", birthYear);
```

Explanation:

User Object:

The user object has properties name, email, and age.

Functions:

- **changeName**(newName): Accepts a newName parameter and updates the name property of the user object.
- **updateEmail**(newEmail): Accepts a newEmail parameter and updates the email property of the user object.
- **calculateBirthYear**(): Calculates the birth year of the user based on the current year and the age property of the user object.

Usage:

- We demonstrate the usage by initially logging the original user object.
- Then, we call changeName and updateEmail functions to modify the user object.
- Finally, we calculate and log the user's birth year using calculateBirthYear.

output:

Initial User Information:

Name: Rohit Sharma

Email: rohit.sharma@example.com

Age: 32

Updated User Information:

Name: Rohit Kumar

Email: rohit.kumar@example.com

Birth Year Calculation:

Birth Year: 1992

Assignment 2: Use regular expressions in JavaScript to validate the email address entered in the form. It should check for the correct format of the email and display a message to the user if the format is incorrect.

```
function validateEmail(email) {
  // Regular expression for basic email validation
  const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  return regex.test(email);
}

// Example usage:
const email1 = 'john.doe@example.com';
const email2 = 'johndoe@example';

if (validateEmail(email1)) {
  console.log("Email address is valid.");
} else {
  console.log("Please enter a valid email address.");
}

if (validateEmail(email2)) {
  console.log("Email address is valid.");
} else {
  console.log("Please enter a valid email address.");
}
```

Explanation:

validateEmail Function:

- **validateEmail**(email): This function takes an email parameter and uses a regular expression (regex) to test if the email format is valid.
- The regular expression **/^[^\s@]+@[^\s@]+\.[^\s@]+$/** ensures that:
- ^[^\s@]+: Starts with one or more characters that are not whitespace or '@'.

- @[^\s@]+: Contains exactly one '@' symbol followed by one or more characters that are not whitespace or '@'.

\.[^\s@]+$: Ends with a dot '.' followed by one or more characters that are not whitespace or '@'.

Example Usage:

- email1 is a valid email address (john.doe@example.com), so it prints "Email address is valid."
- email2 is an invalid email address (johndoe@example), so it prints "Please enter a valid email address."