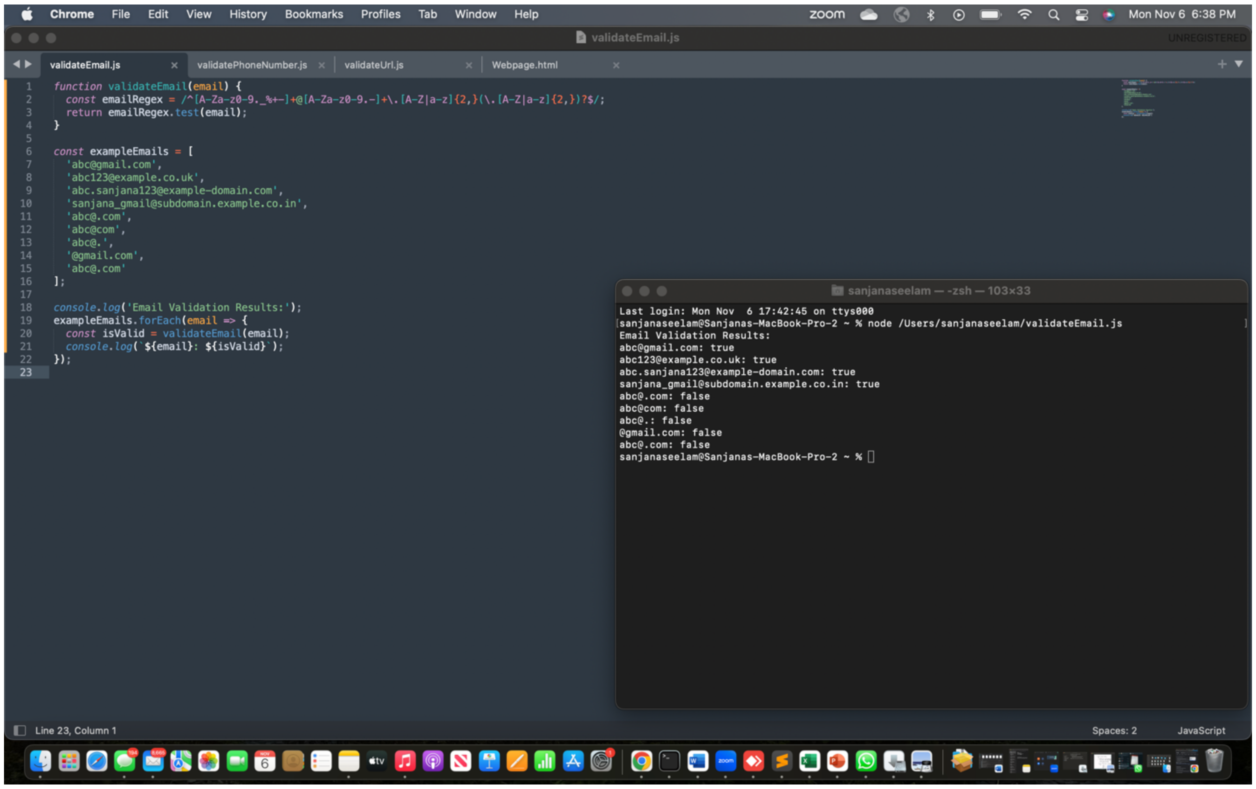
Output Screenshots:

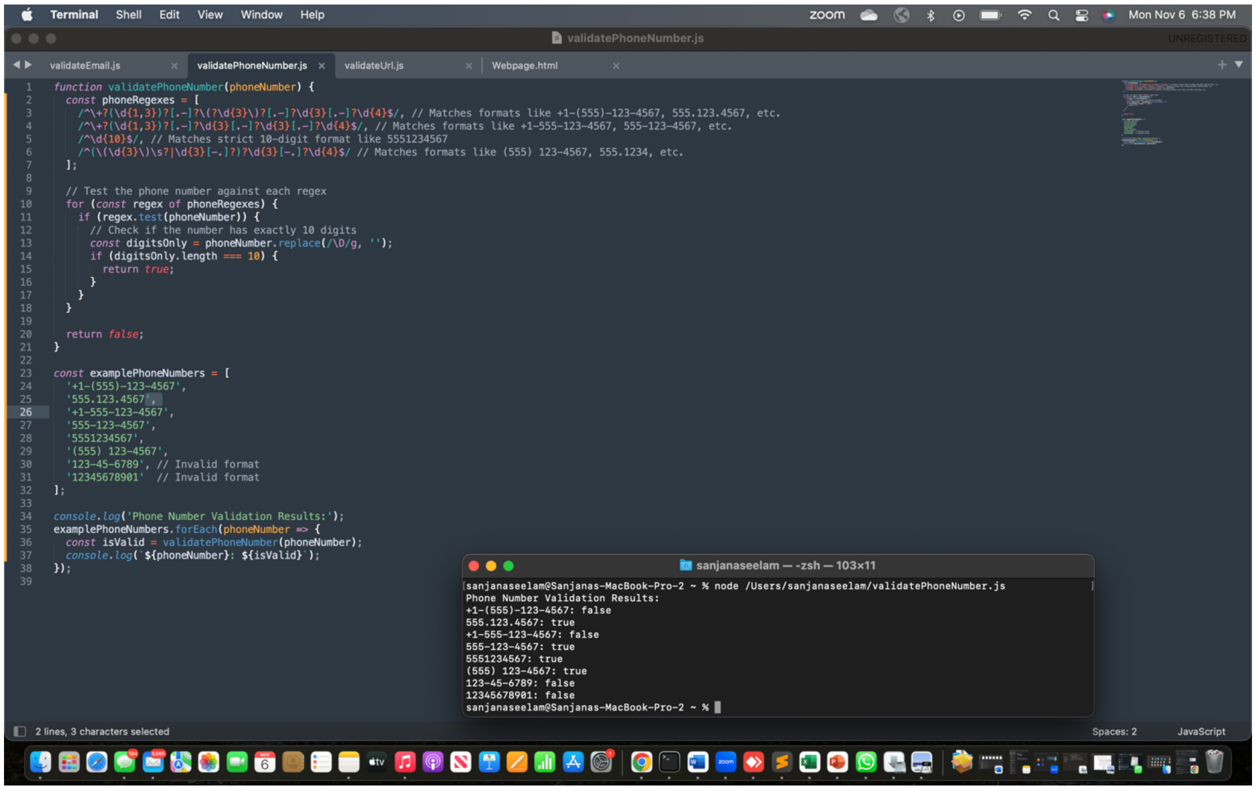
Email Validation:



```
function validateEmail(email) {
  const emailRegex = /^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}(\.[A-Z|a-z]{2,})?$/;
  return emailRegex.test(email);
}

const exampleEmails = [
  'abc@gmail.com',
  'abc123@example.co.uk',
  'abc.sanjana123@example-domain.com',
  'sanjana_gmail@subdomain.example.co.in',
  'abc@.com',
  'abc@com',
  'abc@.',
  '@gmail.com',
  'abc@.com'
];

console.log('Email Validation Results:');
exampleEmails.forEach(email => {
  const isValid = validateEmail(email);
  console.log(`${email}: ${isValid}`);
});
```

```
Last login: Mon Nov  6 17:42:45 on ttys000
sanjanaseelam@Sanjanas-MacBook-Pro-2 ~ % node /Users/sanjanaseelam/validateEmail.js
Email Validation Results:
abc@gmail.com: true
abc123@example.co.uk: true
abc.sanjana123@example-domain.com: true
sanjana_gmail@subdomain.example.co.in: true
abc@.com: false
abc@com: false
abc@.: false
@gmail.com: false
abc@.com: false
sanjanaseelam@Sanjanas-MacBook-Pro-2 ~ %
```

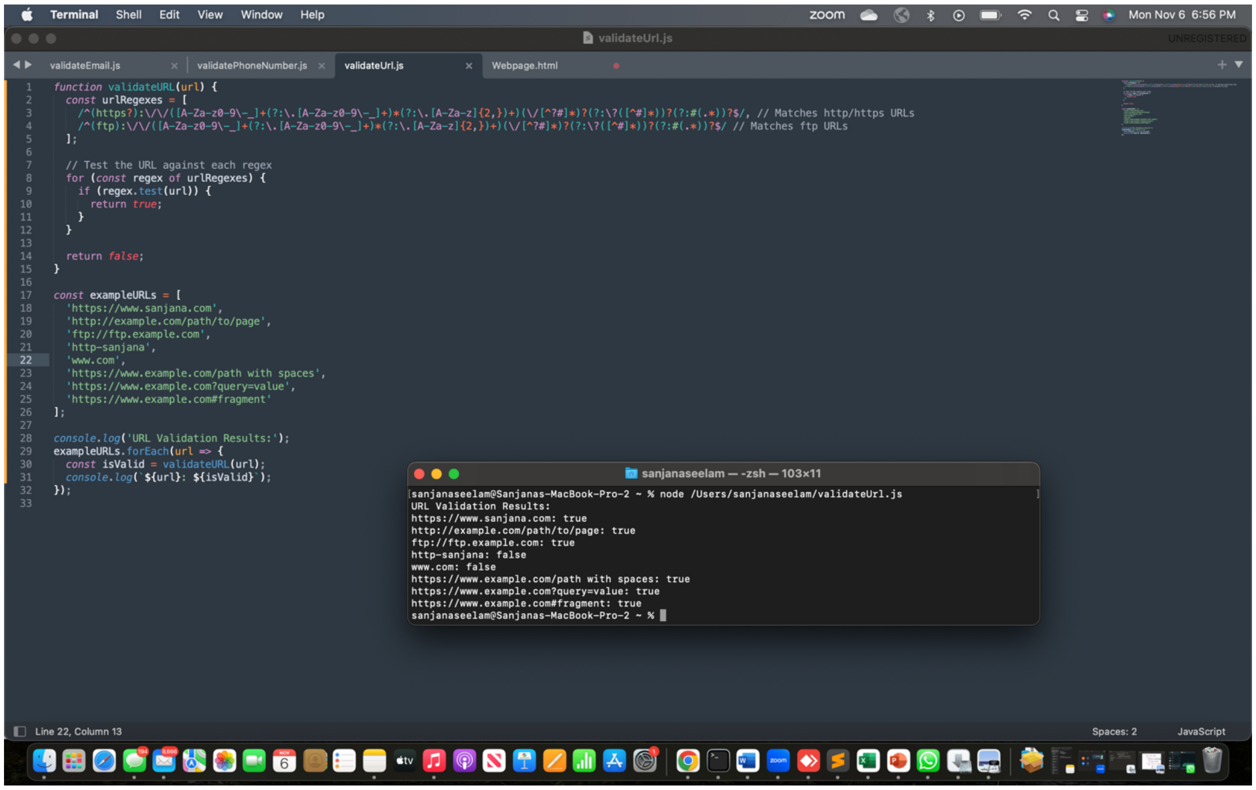
Phone Number Validation:



```
function validatePhoneNumber(phoneNumber) {
  const phoneRegexes = [
    /^\+?(\d{1,3})?[.-]?\(?\d{3}\)?[.-]?\d{3}[.-]?\d{4}$/, // Matches formats like +1-(555)-123-4567, 555.123.4567, etc.
    /^\+?(\d{1,3})?[.-]?\d{3}[.-]?\d{3}[.-]?\d{4}$/, // Matches formats like +1-555-123-4567, 555-123-4567, etc.
    /^\d{10}$/, // Matches strict 10-digit format like 5551234567
    /^(\(\d{3}\)\s?|\d{3}[-.]?)?\d{3}[-.]?\d{4}$/ // Matches formats like (555) 123-4567, 555.1234, etc.
  ];

  // Test the phone number against each regex
  for (const regex of phoneRegexes) {
    if (regex.test(phoneNumber)) {
      // Check if the number has exactly 10 digits
      const digitsOnly = phoneNumber.replace(/\D/g, '');
      if (digitsOnly.length === 10) {
        return true;
      }
    }
  }

  return false;
}

const examplePhoneNumbers = [
  '+1-(555)-123-4567',
  '555.123.4567',
  '+1-555-123-4567',
  '555-123-4567',
  '5551234567',
  '(555) 123-4567',
  '123-45-6789', // Invalid format
  '12345678901'  // Invalid format
];

console.log('Phone Number Validation Results:');
examplePhoneNumbers.forEach(phoneNumber => {
  const isValid = validatePhoneNumber(phoneNumber);
  console.log(`${phoneNumber}: ${isValid}`);
});
```

```
sanjanaseelam@Sanjanas-MacBook-Pro-2 ~ % node /Users/sanjanaseelam/validatePhoneNumber.js
Phone Number Validation Results:
+1-(555)-123-4567: false
555.123.4567: true
+1-555-123-4567: false
555-123-4567: true
5551234567: true
(555) 123-4567: true
123-45-6789: false
12345678901: false
sanjanaseelam@Sanjanas-MacBook-Pro-2 ~ %
```

Url Validation:



```javascript
function validateURL(url) {
  const urlRegexes = [
    /^(https?):\/\/([A-Za-z0-9\-_]+(?:\.[A-Za-z0-9\-_]+)*(?:\.[A-Za-z]{2,})+)(\/[^?#]*)?(?:\?([^#]*))?(?:#(.*))?$/, // Matches http/https URLs
    /^(ftp):\/\/([A-Za-z0-9\-_]+(?:\.[A-Za-z0-9\-_]+)*(?:\.[A-Za-z]{2,})+)(\/[^?#]*)?(?:\?([^#]*))?(?:#(.*))?$/ // Matches ftp URLs
  ];

  // Test the URL against each regex
  for (const regex of urlRegexes) {
    if (regex.test(url)) {
      return true;
    }
  }

  return false;
}

const exampleURLs = [
  'https://www.sanjana.com',
  'http://example.com/path/to/page',
  'ftp://ftp.example.com',
  'http-sanjana',
  'www.com',
  'https://www.example.com/path with spaces',
  'https://www.example.com?query=value',
  'https://www.example.com#fragment'
];

console.log('URL Validation Results:');
exampleURLs.forEach(url => {
  const isValid = validateURL(url);
  console.log(`${url}: ${isValid}`);
});
```

```
sanjanaseelam@Sanjanas-MacBook-Pro-2 ~ % node /Users/sanjanaseelam/validateUrl.js
URL Validation Results:
https://www.sanjana.com: true
http://example.com/path/to/page: true
ftp://ftp.example.com: true
http-sanjana: false
www.com: false
https://www.example.com/path with spaces: true
https://www.example.com?query=value: true
https://www.example.com#fragment: true
sanjanaseelam@Sanjanas-MacBook-Pro-2 ~ %
```