1. HTML (upload_and_display.html):
    - <form>: ទម្រង់ HTML នេះអនុញ្ញាតឱ្យអ្នកប្រើអាចផ្ទុកឯកសារអក្សរ។
    - {% if bow %}: វាគឺជាស្វែងរកមេរៀន Jinja2 ដែលពិនត្រិការយកនូវ Bag-of-Words (bow) មុននឹងបង្ហាញលទ្ធផល។
    - <table>: តាមរយៈតារាង HTML នេះត្រូវបានប្រើដើម្បីបង្ហាញ Bag-of-Words ដែលមានទម្រង់ស្តង់ដារ។
    - {% for word, frequency in bow.items( ) %}: នេះគឺជាស្វែងរកមេរៀន Jinja2 ដែលចែកចាប់លទ្ធផល (ពាក្យ និង ចំនួនទម្រង់) ក្នុងវរៀរ Bag-of-Words។
2. កូដ Python (app.py):
    - generate_bow(file_path): នេះជាអនុគមន៍ដែលយើងចំណាយរបស់របរវាសនាដែលយើងទទួលយកទម្រង់ឯកសារអក្សរហើយប្រើ CountVectorizer ពី scikit-learn ដើម្បីបញ្ចូលអក្សរទៅក្នុងទម្រង់ Bag-of-Words (Term-Document Matrix)។ ទម្រង់លទ្ធផលនោះត្រូវបានបក្សនាឱ្យជាវ៉ូតាមសកលលោកទាំងឡាយនៅក្នុងអក្សរ។
    - array_to_string(array): នេះជាអនុគមន៍ដែលយើងប្រើដើម្បីបក្សនា NumPy array (នៅទីនេះគឺទម្រង់ Bag-of-Words) ទៅជាប្រយ័ត្ននូវលទ្ធផលអាន់ក្រែតទំនិញពេលបង្ហាញលទ្ធផលនៅលើទំព័រ។
    - index( ): នេះគឺជាអនុគមន៍ដើម។ វាជាអ្នករបស់ផ្ទាំងសារពេលគេសរសេរព័ត៌មានដើម្បីផ្ទុកទំព័រដើមតាងទិន្នន័យ (/)។
    - process_file( ): នេះគឺជាអនុគមន៍ដែលត្រូវបញ្ចូលព័ត៌មាននៃការផ្ទុកឯកសារ។ វារកតំបន់ស្វែងរកឯកសារនិងព្យាងប្រូរប្រារទាំងអស់។ ការប្រងិតឯកសារ Bag-of-Words និងបញ្ជីទម្រង់តាងនៅលើទំព័រវេបសាយ។
    - ប្រូរប្រារទាំង១ងាយងាយស្តិតក្នុង main។
3. កម្មវិធី Flask:
    - កម្មវិធី Flask ត្រូវបានបង្កើតដោយប្រើ Flask(__name__)។
    - តាមទម្រង់ @app.route decorator ត្រូវបានកំណត់ទីតាំងដោយប្រើរបៀប index( ) និង /process_file ត្រូវបានកំណត់ដោយប្រើរបៀប process_file( )។
    - កម្មវិធីត្រូវរត់ជាប្រតិបត្តិបរនៅដោយ app.run(debug=True)។