

# DOKUMENTASI PEMBUATAN LANDING PAGE UTS

Nama: Zuhad Harkasy Al-Hikam

NIM: 1217050148

Kelas: Praktikum Pengembangan Aplikasi Website F

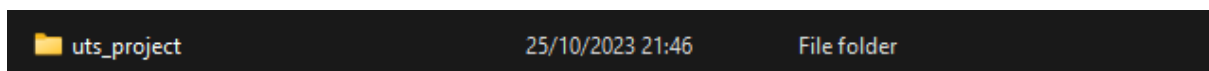
Dosen Pengampu: H. Aldy Rialdy Atmadja M.T

## Menyiapkan environment pengembangan

Sebelum memulai proses pembuatan web, terlebih dahulu saya menyiapkan tools yang dibutuhkan, diantaranya adalah vs code sebagai teks editor, Microsoft chrome untuk melakukan debugging, dan beberapa extension untuk memudahkan proses pengembangan.

## Membuat direktori project

Setelah tools siap digunakan, proses coding pun dimulai, saya mengawali proses coding dengan membuat folder/direktori khusus untuk project terkait, kali ini saya menamainya dengan ``uts_project``. Pembuatan direktori ini dapat dilakukan baik itu melalui text editor ataupun file manager. Berikut adalah hasil pembuatan direktori project.



## Membuka vs code dan memasuki direktori project

Hal ini dapat tercapai dengan berbagai cara, beberapa cara populer adalah dengan membuka folder secara manual melalui halaman awal vs code, atau dapat dengan melakukan **klik kanan di dalam folder project->show more option-> open with vs code**. Setelah masuk, tampilan explorer vs code akan seperti ini.

## Membuat file index.html

Index.html dibuat untuk menyimpan source code html yang dibuat. Adapun penempatannya pastikan berada dalam root dari folder project yang dibuat.

## Menyiapkan asset

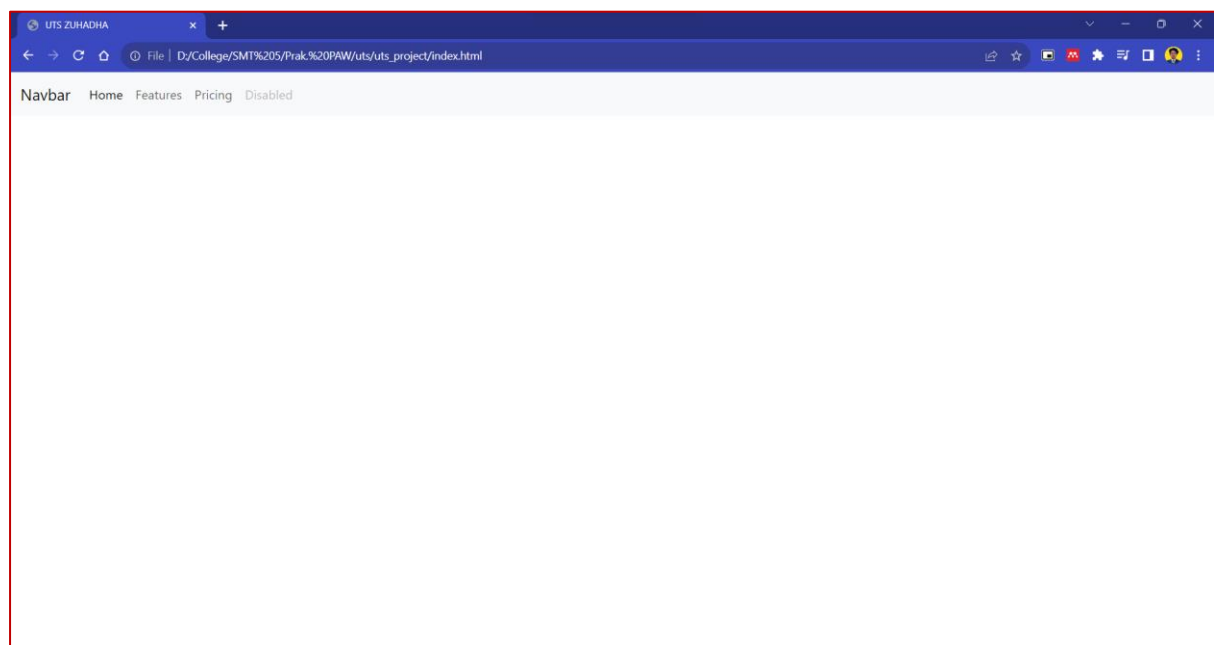
Sebelum memulai coding, alangkah baiknya jika kita terlebih dahulu menyiapkan asset apa saja yang diperlukan, asset dapat berupa gambar, logo, file, dan lain sebagainya, asset-asset ini biasanya disimpan dalam satu direktori yang sama yang bernama 'assets'. Dengan menyiapkan asset, kita dapat melakukan coding tanpa hambatan, kita dapat langsung me-refer asset yang dituju dengan filepath dan nama file yang tepat.

## Memulai coding index.html

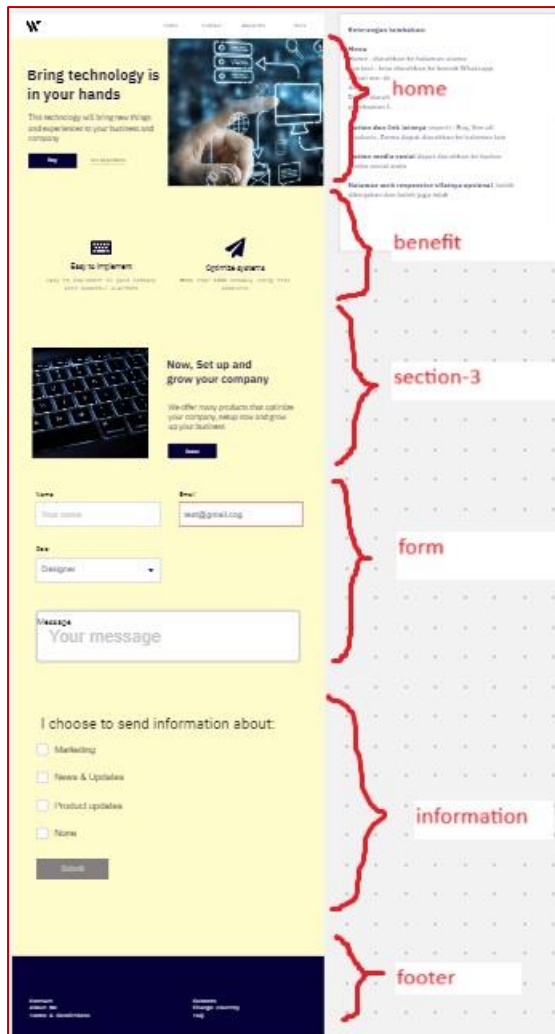
Setelah file tersedia, Langkah selanjutnya adalah dengan memulai coding pada file tersebut. Coding dilakukan berurutan mengikuti *wireframe* yang disediakan. Pada project kali ini, saya menggunakan dua metode dalam mengembangkan website, pertama adalah dengan bantuan framework bootstrap, dan yang kedua adalah dengan html+css manual. Pertama, membuat navbar, untuk memudahkan proses pembuatan website, saya mencari template pada internet dan melakukan beberapa penyesuaian untuk membuat tampilan yang sesuai, berikut adalah code yang digunakan beserta tampilan yang diberikan:

```
1 <nav class="navbar navbar-expand-lg navbar-light bg-light">
2   <div class="container-fluid">
3     <a class="navbar-brand" href="#">Navbar</a>
4     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
5       <span class="navbar-toggler-icon"></span>
6     </button>
7     <div class="collapse navbar-collapse" id="navbarNav">
8       <ul class="navbar-nav">
9         <li class="nav-item">
10          <a class="nav-link active" aria-current="page" href="#">Home</a>
11        </li>
12        <li class="nav-item">
13          <a class="nav-link" href="#">Features</a>
14        </li>
15        <li class="nav-item">
16          <a class="nav-link" href="#">Pricing</a>
17        </li>
18        <li class="nav-item">
19          <a class="nav-link disabled" href="#" tabindex="-1" aria-disabled="true">Disabled</a>
20        </li>
21      </ul>
22    </div>
23  </div>
24 </nav>
```

Tampilan:



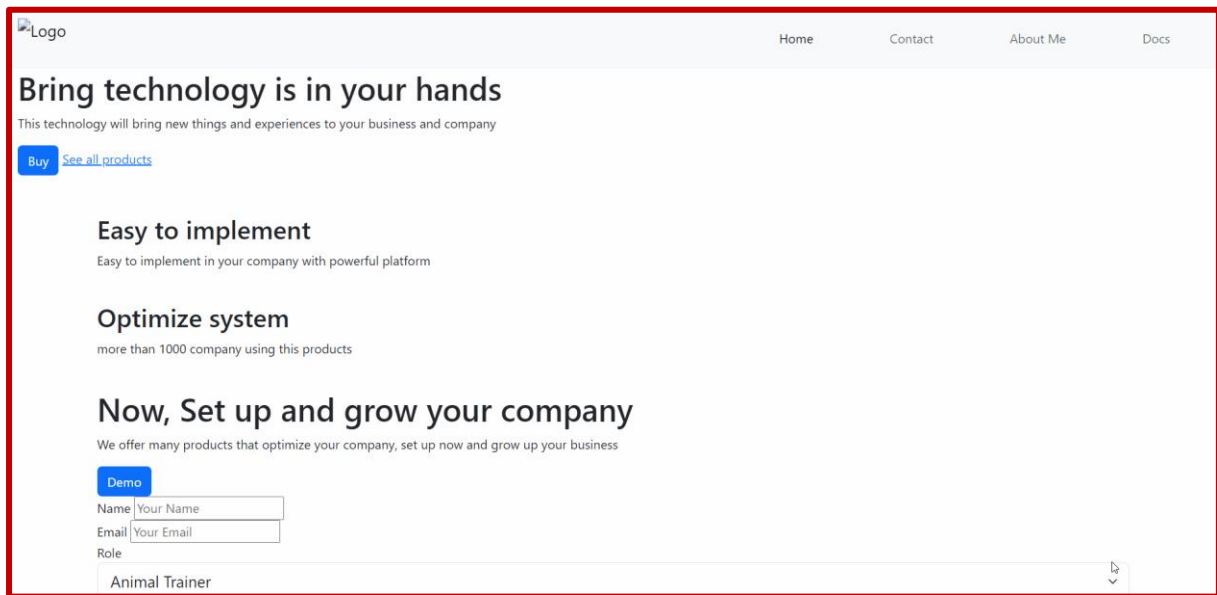
Selanjutnya, dilakukan coding html per section, saya membagi desain yang diberikan ke dalam beberapa section yaitu section 'home' atau section pertama/paling atas yang ada pada desain, selanjutnya ada section kedua yang diberi nama 'benefit'. Ketiga ada section yang tidak tahu penamaan apa yang relevan sehingga saya beri nama 'section-3'. Kemudian, terdapat beberapa input field yang dapat diisi sehingga diberi nama 'form'. Selanjutnya ada serangkaian informasi yang dapat diceklis saya beri nama 'information'. Terakhir, section yang terlihat seperti footer atau penutup dari web tersebut sehingga saya beri nama 'footer'. Section-section yang saya buat ini kemudian akan menjadi container dengan nama class section tersebut. Berikut adalah visualisasi dari pembagian section tersebut:



Setelah itu, dilakukan coding dengan menerapkan teknik ‘kontainerisasi’ di mana section dibungkus oleh container berupa tag div, hal ini dilakukan untuk mempermudah proses desain dengan css yakni penerapan flexbox dan memudahkan proses pembuatan web yang responsif terhadap ukuran layer.

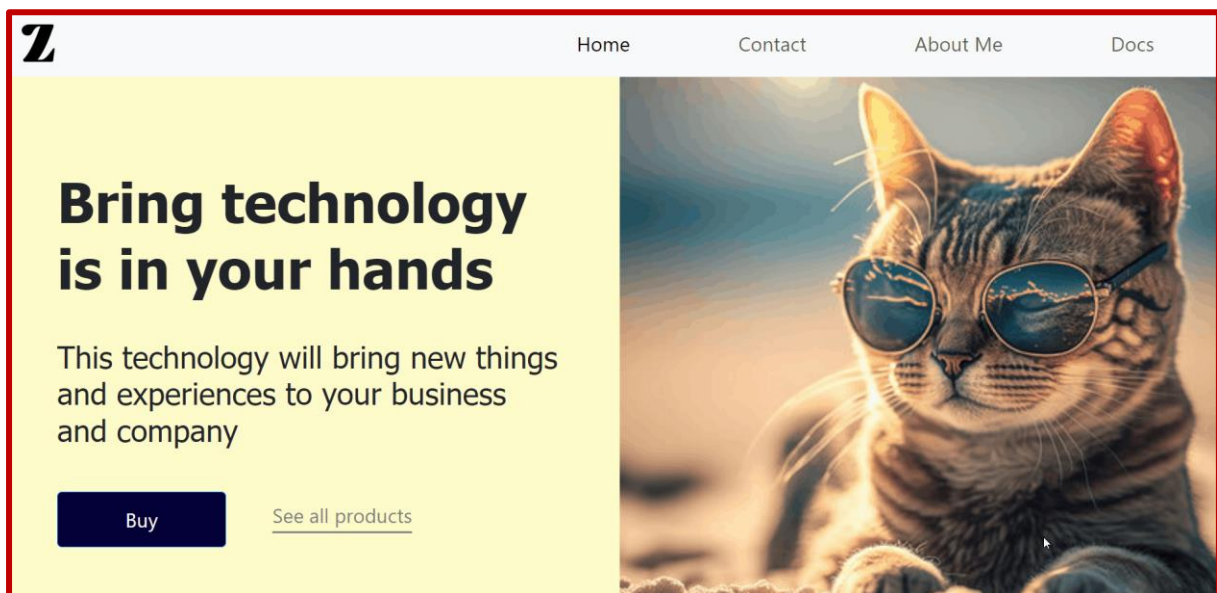
Source code dapat diakses [di sini](#)

Tampilan yang dihasilkan setelah memasukkan semua elemen adalah sebagai berikut (tekan gambar atau klik [di sini](#) untuk melihat secara lengkap):



### Memberi style dengan .css

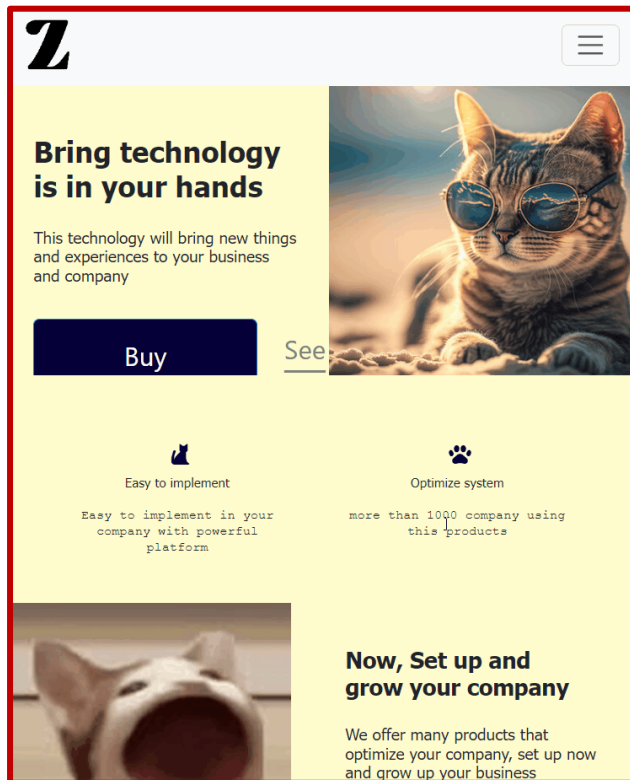
Setelah seluruh komponen dimuat dalam index.html, Langkah selanjutnya adalah memberikan style sehingga tampilan web dapat sesuai dengan yang diharapkan. Saya biasa melakukan coding dari atas ke bawah, sehingga proses pengerjaan css pun saya lakukan demikian. Bermula dari section home dan berakhir di section footer. Berikut adalah tampilan yang dihasilkan setelah index.html disambungkan dengan style.css. File css dapat diakses [di sini](#). (tekan gambar atau klik [di sini](#) untuk melihat secara lengkap):



Sekilas tampilan website sudah selesai dibuat, memang benar bahwa web tersebut sudah terlihat bagus dan rapi, tetapi tampilan web hanya terlihat demikian Ketika web dibuka melalui laptop alias website dibuka seukuran laptop. Sekarang mari kita lihat jika web dibuka dari tablet atau hp:

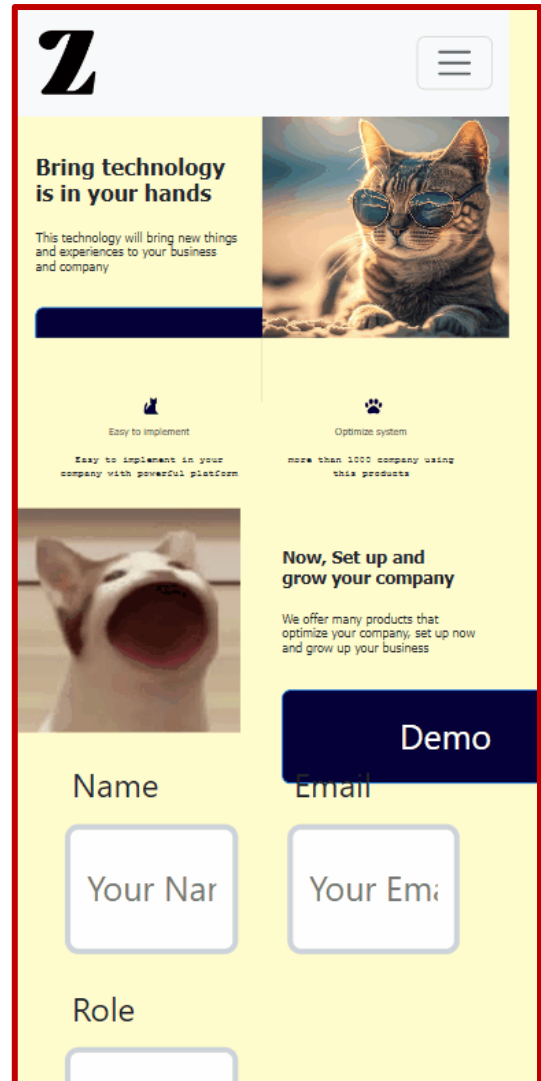
Tampilan pada ukuran layer tablet:

(tekan gambar atau klik [di sini](#) untuk melihat secara lengkap):



Tampilan pada ukuran layer smartphone:

(tekan gambar atau klik [di sini](#) untuk melihat secara lengkap):



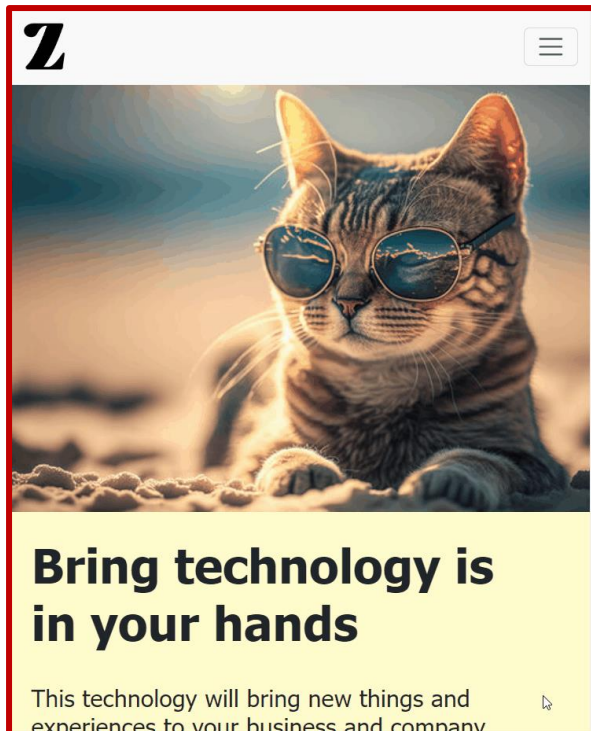
Sekilas terlihat agak rapi, tetapi jika diperhatikan, tampilan website tersebut amat berantakan dan sulit dibaca, elemen-elemen saling bertumpukan, ukuran font tidak enak dipandang, dan lain sebagainya. Untuk menangani hal tersebut, digunakanlah yang namanya media query pada style.css. Media query memungkinkan kita untuk menentukan style elemen pada html akan ditampilkan bagaimana tergantung dengan ukuran website Ketika dibuka.

Pada project ini, digunakan dua breakpoint sebagai media query yaitu max-width: 576px dan max-width: 992px. Dengan begitu, elemen yang ditampilkan mulai dari gambar sampai teks dapat diatur berbeda-beda sesuai dengan Batasan ukuran yang digunakan. Max-width: 576px berarti elemen akan ditampilkan sedemikian Ketika layer berukuran lebih kecil dari 576px pun sama halnya dengan max-width:992px.

Berikut adalah tampilan website setelah ditambahkan media query pada style.css:

Tampilan pada ukuran layer tablet:

(tekan gambar atau klik [di sini](#) untuk melihat secara lengkap):



Perbedaan dapat dilihat secara jelas bahwa dengan menggunakan media query, tampilan website yang kita buat dapat menyesuaikan berbagai keadaan sehingga memudahkan user dalam mengakses di manapun dan kapan pun.

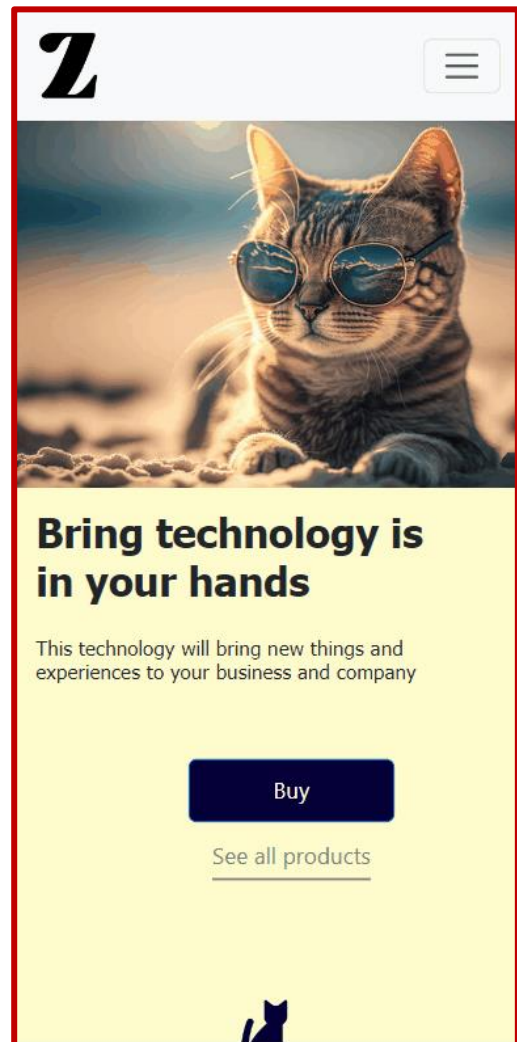
Terdapat beberapa kelemahan dalam menggunakan media query sebagai solusi dalam menyelesaikan masalah akses website melalui berbagai device (responsif) diantaranya adalah prosesnya yang memakan waktu, dan proses pembuatan css terasa lebih lama bahkan seperti melakukan styling berulang.

Selain itu, terdapat beberapa kendala dalam proses pengimplementasian style yang cukup menghambat proses pembuatan yaitu salah refer elemen. Maka penting bagi kita untuk mengingat class dan elemen apa yang harus digunakan untuk memuat tampilan tertentu. Solusi yang saya gunakan selama proses pembuatan website ini adalah dengan melakukan test dengan memberikan elemen terkait border atau warna background yang kontras dengan background yang berlaku, jika warna berubah atau border tampil, artinya elemen target berhasil diakses.

Terakhir, saya menghubungkan beberapa tombol ke berbagai lokasi yang berbeda, seperti tombol 'home' pada navigasi akan membawa user ke halaman utama, 'contact' membawa user ke chat whatsapp, 'about me' membawa user ke cv, dan docs membawa user ke halaman documentation. Halaman documentation menampilkan pdf secara embedded sehingga dapat diakses user tanpa harus mendownload file terlebih dahulu.

Tampilan pada ukuran layer smartphone:

(tekan gambar atau klik [di sini](#) untuk melihat secara lengkap):





Berikut hasil navigasi yang telah ditambahkan (tekan gambar untuk membuka video, atau klik [di sini](#)):

