File Encryption /Decryption

Source Code: -

HTML/CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>File Encryption/Decryption Tool</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
      font-family: 'Segoe UI', Arial, sans-serif;
    }

    body {
      background: linear-gradient(135deg, rgb(220, 103, 7),white,green);
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      padding: 1rem;
    }
```

```css
.container {
   background-color: cyan;
   padding: 2rem;
   border-radius: 12px;
   box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15);
   width: 100%;
   max-width: 500px;
   text-align: center;
}

h1 {
   color: #333;
   margin-bottom: 1.5rem;
   font-size: 1.8rem;
   font-weight: 600;
}

.form-container {
   display: flex;
   flex-direction: column;
   gap: 1rem;
}

input[type="file"] {
   padding: 0.5rem;
```

```
    border: 1px solid #b0bec5;
    border-radius: 6px;
    font-size: 1rem;
    background-color: #f5f5f5;
}

input[type="text"] {
    padding: 0.75rem;
    border: 1px solid #b0bec5;
    border-radius: 6px;
    font-size: 1rem;
    background-color: #f5f5f5;
}

select {
    padding: 0.75rem;
    border: 1px solid #b0bec5;
    border-radius: 6px;
    font-size: 1rem;
    background-color: #f5f5f5;
}

button {
    padding: 0.75rem;
    background-color: #26a69a;
    color: #ffffff;
```

```css
    border: none;
    border-radius: 6px;
    font-size: 1rem;
    font-weight: 500;
    cursor: pointer;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #00897b;
}

.status {
    margin-top: 1rem;
    color: #333;
    font-size: 0.9rem;
    font-weight: 500;
}

.status.error {
    color: #d32f2f;
}

.status.success {
    color: #2e7d32;
}
```

```
@media (max-width: 600px) {
  .container {
    padding: 1.5rem;
    max-width: 90vw;
  }

  h1 {
    font-size: 1.5rem;
  }

  input[type="file"], input[type="text"], select, button {
    font-size: 0.9rem;
    padding: 0.6rem;
  }
}

@media (min-width: 601px) and (max-width: 900px) {
  .container {
    padding: 1.75rem;
    max-width: 80vw;
  }

  h1 {
    font-size: 1.7rem;
  }
```

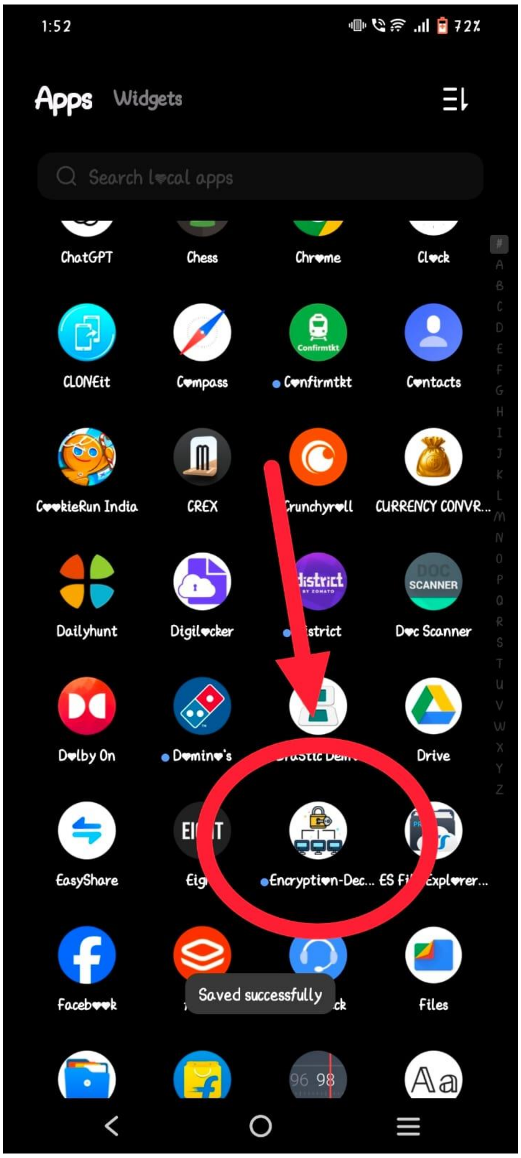
```
            input[type="file"], input[type="text"], select, button {
                font-size: 0.95rem;
            }
        }

        @media (min-width: 901px) {
            .container {
                max-width: 500px;
            }
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>File Encryption/Decryption Tool</h1>
        <div class="form-container">
            <input type="file" id="fileInput" accept="*/*">
            <input type="text" id="keyInput" placeholder="Enter encryption/decryption key">
            <select id="operation">
                <option value="encrypt">Encrypt</option>
                <option value="decrypt">Decrypt</option>
            </select>
            <button onclick="processFile()">Process</button>
        </div>
        <div id="status" class="status"></div>
    </div>
```

JS

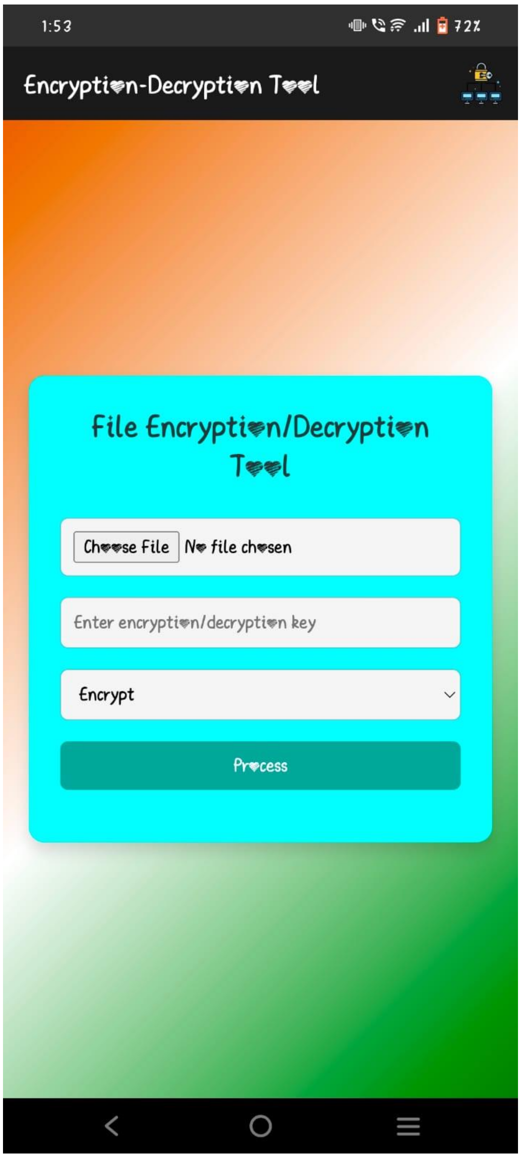
```
  <script>
    async function processFile() {
      const fileInput = document.getElementById('fileInput');
      const keyInput = document.getElementById('keyInput');
      const operation = document.getElementById('operation').value;
      const status = document.getElementById('status');

      status.textContent = '';
      status.className = 'status';

      if (!fileInput.files[0]) {
        status.textContent = 'Please select a file.';
        status.className = 'status error';
        return;
      }
      if (!keyInput.value.trim()) {
        status.textContent = 'Please enter a key.';
        status.className = 'status error';
        return;
      }

      const file = fileInput.files[0];
      const key = keyInput.value;

      try {
        const fileContent = await readFile(file);

        let result;
        let outputFileName;

        if (operation === 'encrypt') {
          result = CryptoJS.AES.encrypt(fileContent, key).toString();
          outputFileName = file.name + '.encrypted';
        } else {
          try {
            const bytes = CryptoJS.AES.decrypt(fileContent, key);
            result = bytes.toString(CryptoJS.enc.Utf8);
            if (!result) {
              throw new Error('Decryption failed. Invalid key or corrupted file.');
            }
            outputFileName = file.name.replace('.encrypted', '.decrypted');
          } catch (error) {
            status.textContent = 'Decryption failed. Invalid key or corrupted file.';
            status.className = 'status error';
            return;
          }
        }

        const blob = new Blob([result], { type: 'text/plain' });
        const url = URL.createObjectURL(blob);
```

```
                const a = document.createElement('a');
                a.href = url;
                a.download = outputFileName;
                a.click();
                URL.revokeObjectURL(url);

                status.textContent = `${operation === 'encrypt' ? 'Encryption' : 'Decryption'} successful! File downloaded.`;
                status.className = 'status success';
            } catch (error) {
                status.textContent = 'An error occurred: ' + error.message;
                status.className = 'status error';
            }
        }
        function readFile(file) {
            return new Promise((resolve, reject) => {
                const reader = new FileReader();
                reader.onload = () => resolve(reader.result);
                reader.onerror = () => reject(reader.error);
                reader.readAsText(file);
            });
        }
    </script>
</body>
</html>
```

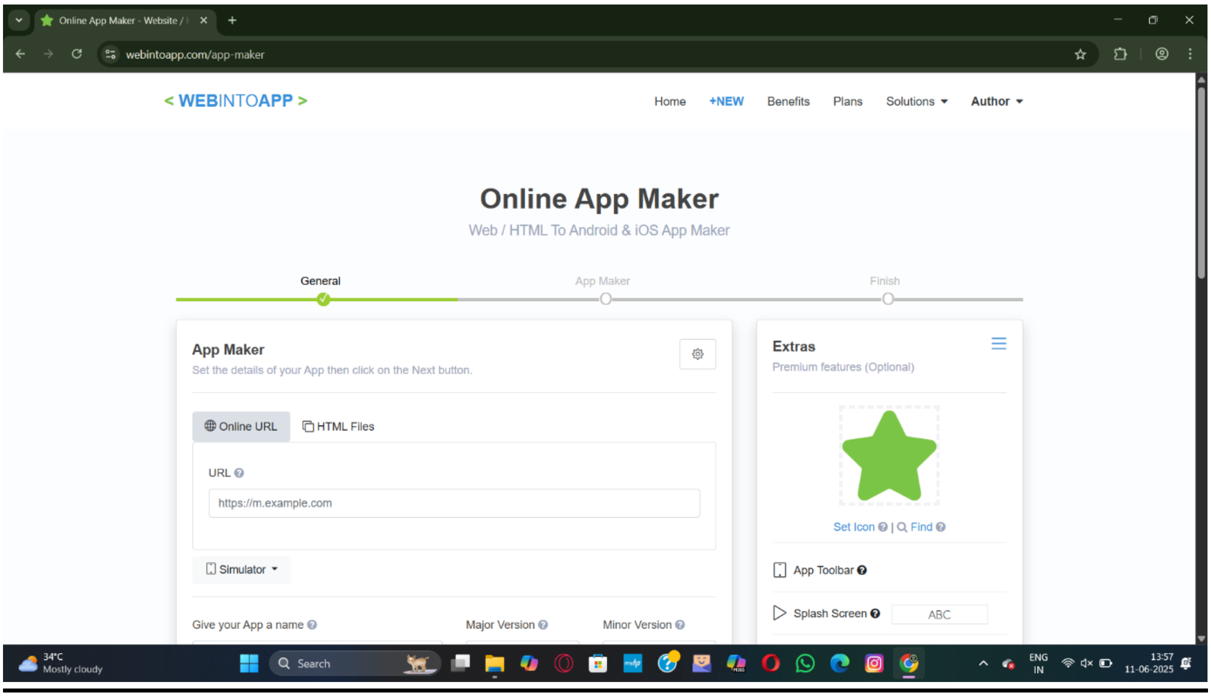
Images



Saved successfully



File Encryption/Decryption Tool



Online App Maker

Web / HTML To Android & iOS App Maker

App Maker

Set the details of your App then click on the Next button.

URL: https://m.example.com

Give your App a name

Major Version

Minor Version

Extras

Premium features (Optional)