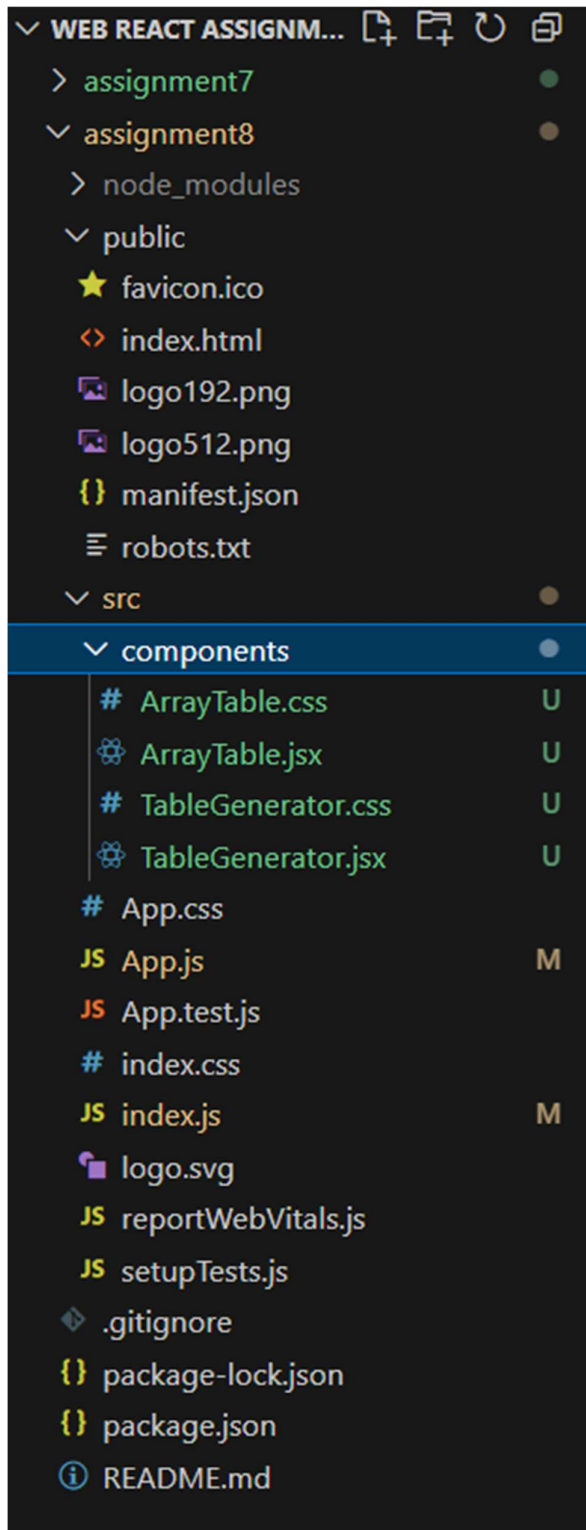


Layout Structure:



App.js

```
import React from "react";
import TableGenerator from "../components/TableGenerator";
import "../index.css";
import ArrayTable from "../components/ArrayTable";

const App = () => {
  return (
    <div className="app">
      <ArrayTable />
    </div>
  );
};

export default App;
```

index.js

```
import React from "react";
import ReactDOM from "react-dom/client";
import "../index.css";
import App from "../App";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(<App />);
```

index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
    monospace;
}
```

1. 2D Table

TableGenerator.jsx

```
import React, { useState } from "react";
import "./TableGenerator.css";

const TableGenerator = () => {
  const [rows, setRows] = useState(3);
  const [cols, setCols] = useState(3);
  const [tableData, setTableData] = useState([]);
  const [isGenerating, setIsGenerating] = useState(false);

  const handleSubmit = (e) => {
    e.preventDefault();
    setIsGenerating(true);

    setTimeout(() => {
      const newTable = [];
      for (let i = 0; i < rows; i++) {
        const row = [];
        for (let j = 0; j < cols; j++) {
          row.push(`R${i + 1}C${j + 1}`);
        }
        newTable.push(row);
      }
      setTableData(newTable);
      setIsGenerating(false);
    }, 300);
  };

  return (
    <div className="table-generator">
      <div className="glass-card">
        <h2 className="title">2D Table Generator</h2>
        <p className="subtitle">Create custom tables with dynamic dimensions</p>

        <form onSubmit={handleSubmit} className="generator-form">
          <div className="input-group">
            <div className="input-wrapper">
              <label htmlFor="rows">Rows</label>
              <input
                type="number"
                id="rows"
                min="1"
                max="10"
                value={rows}
                onChange={(e) => setRows(Math.min(Number(e.target.value), 10))}
                className="dimension-input"
                required
              />
            </div>
          </div>
        </form>
      </div>
    </div>
  );
};
```

```

    <div className="input-wrapper">
      <label htmlFor="cols">Columns</label>
      <input
        type="number"
        id="cols"
        min="1"
        max="10"
        value={cols}
        onChange={(e) => setCols(Math.min(Number(e.target.value), 10))}
        className="dimension-input"
        required
      />
    </div>
  </div>

  <button type="submit" className="generate-btn" disabled={isGenerating}>
    {isGenerating ? 'Generating...' : 'Generate Table'}
  </button>
</form>

{tableData.length > 0 && (
  <div className="table-wrapper">
    <table className="styled-table">
      <tbody>
        {tableData.map((row, rowIndex) => (
          <tr key={rowIndex}>
            {row.map((cell, cellIndex) => (
              <td key={cellIndex}>
                <div className="cell-content">{cell}</div>
              </td>
            ))}
          </tr>
        ))}
      </tbody>
    </table>
  </div>
)}
</div>
</div>
);
};

export default TableGenerator;

```

TableGenerator.css

```
/* Shared Styles */
:root {
  --primary-color: #6366f1;
  --primary-light: #818cf8;
  --secondary-color: #f59e0b;
  --text-color: #1e293b;
  --text-light: #64748b;
  --bg-color: #f8fafc;
  --glass-bg: rgba(255, 255, 255, 0.9);
  --shadow: 0 4px 30px rgba(0, 0, 0, 0.1);
  --border-radius: 12px;
}

body {
  background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
  font-family: 'Inter', -apple-system, BlinkMacSystemFont, sans-serif;
  color: var(--text-color);
}

.glass-card {
  background: var(--glass-bg);
  backdrop-filter: blur(16px);
  -webkit-backdrop-filter: blur(16px);
  border-radius: var(--border-radius);
  box-shadow: var(--shadow);
  border: 1px solid rgba(255, 255, 255, 0.3);
  padding: 2rem;
  max-width: 800px;
  margin: 0 auto;
  transition: all 0.3s ease;
}

.glass-card:hover {
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.15);
}

.title {
  font-size: 1.8rem;
  font-weight: 700;
  color: var(--primary-color);
  margin-bottom: 0.5rem;
  text-align: center;
}

.subtitle {
  color: var(--text-light);
  text-align: center;
  margin-bottom: 2rem;
  font-size: 1rem;
}
```

```
/* TableGenerator Specific Styles */
.table-generator {
  padding: 2rem;
}

.generator-form {
  margin: 2rem 0;
}

.input-group {
  display: flex;
  gap: 1.5rem;
  justify-content: center;
  margin-bottom: 1.5rem;
}

.input-wrapper {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.input-wrapper label {
  font-size: 0.9rem;
  color: var(--text-light);
  margin-bottom: 0.5rem;
  font-weight: 500;
}

.dimension-input {
  width: 80px;
  padding: 0.75rem;
  border: 1px solid rgba(99, 102, 241, 0.3);
  border-radius: 8px;
  text-align: center;
  font-size: 1rem;
  font-weight: 600;
  color: var(--primary-color);
  transition: all 0.3s ease;
}

.dimension-input:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(99, 102, 241, 0.2);
}

.generate-btn {
  background: var(--primary-color);
  color: white;
  border: none;
  padding: 0.75rem 2rem;
  border-radius: 8px;
  font-size: 1rem;
}
```

```
font-weight: 600;
cursor: pointer;
transition: all 0.3s ease;
display: block;
margin: 0 auto;
width: 200px;
}

.generate-btn:hover {
  background: var(--primary-light);
  transform: translateY(-2px);
}

.generate-btn:disabled {
  background: #cbd5e1;
  cursor: not-allowed;
  transform: none;
}

/* ArrayTable Specific Styles */
.array-table-converter {
  padding: 2rem;
}

.json-input {
  width: 100%;
  padding: 1rem;
  border: 1px solid rgba(99, 102, 241, 0.3);
  border-radius: 8px;
  font-family: 'Fira Code', monospace, 'Inter', sans-serif;
  font-size: 0.9rem;
  resize: vertical;
  min-height: 120px;
  transition: all 0.3s ease;
}

.json-input:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(99, 102, 241, 0.2);
}

.convert-btn {
  background: var(--secondary-color);
  color: white;
  border: none;
  padding: 0.75rem 2rem;
  border-radius: 8px;
  font-size: 1rem;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s ease;
  display: block;
  margin: 1.5rem auto 0;
}
```

```
width: 200px;
}

.convert-btn:hover {
  background: #f97316;
  transform: translateY(-2px);
}

.convert-btn:disabled {
  background: #cbd5e1;
  cursor: not-allowed;
  transform: none;
}

.error-message {
  color: #ef4444;
  font-size: 0.9rem;
  margin-top: 0.5rem;
  text-align: center;
}

/* Shared Table Styles */
.table-wrapper {
  margin-top: 2rem;
  overflow-x: auto;
  border-radius: var(--border-radius);
  box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
}

.styled-table {
  width: 100%;
  border-collapse: separate;
  border-spacing: 0;
  border-radius: var(--border-radius);
  overflow: hidden;
  animation: fadeIn 0.5s ease;
}

.styled-table tr:first-child td:first-child {
  border-top-left-radius: var(--border-radius);
}

.styled-table tr:first-child td:last-child {
  border-top-right-radius: var(--border-radius);
}

.styled-table tr:last-child td:first-child {
  border-bottom-left-radius: var(--border-radius);
}

.styled-table tr:last-child td:last-child {
  border-bottom-right-radius: var(--border-radius);
}
```



```
.styled-table td {
  padding: 1rem;
  border: 1px solid rgba(203, 213, 225, 0.5);
  text-align: center;
  transition: all 0.2s ease;
}

.styled-table tr:nth-child(even) td {
  background-color: rgba(241, 245, 249, 0.5);
}

.styled-table tr:hover td {
  background-color: rgba(224, 231, 255, 0.3);
}

.cell-content {
  padding: 0.5rem;
  min-height: 20px;
}

@keyframes fadeIn {
  from { opacity: 0; transform: translateY(10px); }
  to { opacity: 1; transform: translateY(0); }
}

/* Responsive Styles */
@media (max-width: 768px) {
  .glass-card {
    padding: 1.5rem;
  }

  .input-group {
    flex-direction: column;
    gap: 1rem;
  }

  .dimension-input, .generate-btn, .convert-btn {
    width: 100%;
  }
}
```

2D Table Generator

Create custom tables with dynamic dimensions

Rows

3

Columns

3

Generate Table

2D Table Generator

Create custom tables with dynamic dimensions

Rows

04

Columns

05

Generate Table

R1C1	R1C2	R1C3	R1C4	R1C5
R2C1	R2C2	R2C3	R2C4	R2C5
R3C1	R3C2	R3C3	R3C4	R3C5
R4C1	R4C2	R4C3	R4C4	R4C5

2. Table from String

ArrayTable.jsx

```
import React, { useState } from "react";
import "./ArrayTable.css";

const ArrayTable = () => {
  const [inputValue, setInputValue] = useState('');
  const [tableData, setTableData] = useState([]);
  const [isGenerating, setIsGenerating] = useState(false);
  const [error, setError] = useState(null);

  const handleGenerateTable = () => {
    setIsGenerating(true);
    setError(null);

    setTimeout(() => {
      try {
        const parsedData = JSON.parse(inputValue);
        if (Array.isArray(parsedData) && parsedData.every((row) => Array.isArray(row))) {
          setTableData(parsedData);
        } else {
          setError("Input must be a 2D array (array of arrays)");
        }
      } catch (error) {
        setError("Invalid JSON input. Please check your syntax.");
      } finally {
        setIsGenerating(false);
      }
    }, 300);
  };

  return (
    <div className="array-table-converter">
      <div className="glass-card">
        <h2 className="title">2D Array to Table Converter</h2>
        <p className="subtitle">Paste your 2D array in JSON format</p>

        <div className="input-container">
          <textarea
            value={inputValue}
            onChange={(e) => setInputValue(e.target.value)}
            placeholder='Enter a 2D array in JSON format, e.g.: [{"A1", "A2"}, {"B1",
"B2"}]}'
            rows={5}
            className="json-input"
          />

          {error && <div className="error-message">{error}</div>}

          <button
```

```

        onClick={handleGenerateTable}
        className="convert-btn"
        disabled={isGenerating || !inputValue.trim()}
      >
        {isGenerating ? 'Converting...' : 'Convert to Table'}
      </button>
    </div>

    {tableData.length > 0 && (
      <div className="table-wrapper">
        <table className="styled-table">
          <tbody>
            {tableData.map((row, rowIndex) => (
              <tr key={rowIndex}>
                {row.map((cell, cellIndex) => (
                  <td key={cellIndex}>
                    <div className="cell-content">
                      {typeof cell === 'object' ? JSON.stringify(cell) : cell}
                    </div>
                  </td>
                ))}
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    )}
  </div>
</div>
);
};

export default ArrayTable;

```

ArrayTable.css

```
/* Shared Styles */
:root {
  --primary-color: #6366f1;
  --primary-light: #818cf8;
  --secondary-color: #f59e0b;
  --text-color: #1e293b;
  --text-light: #64748b;
  --bg-color: #f8fafc;
  --glass-bg: rgba(255, 255, 255, 0.9);
  --shadow: 0 4px 30px rgba(0, 0, 0, 0.1);
  --border-radius: 12px;
}

body {
  background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
  font-family: 'Inter', -apple-system, BlinkMacSystemFont, sans-serif;
  color: var(--text-color);
}

.glass-card {
  background: var(--glass-bg);
  backdrop-filter: blur(16px);
  -webkit-backdrop-filter: blur(16px);
  border-radius: var(--border-radius);
  box-shadow: var(--shadow);
  border: 1px solid rgba(255, 255, 255, 0.3);
  padding: 2rem;
  max-width: 800px;
  margin: 0 auto;
  transition: all 0.3s ease;
}

.glass-card:hover {
  box-shadow: 0 8px 32px rgba(0, 0, 0, 0.15);
}

.title {
  font-size: 1.8rem;
  font-weight: 700;
  color: var(--primary-color);
  margin-bottom: 0.5rem;
  text-align: center;
}

.subtitle {
  color: var(--text-light);
  text-align: center;
  margin-bottom: 2rem;
  font-size: 1rem;
}
```

```
/* TableGenerator Specific Styles */
.table-generator {
  padding: 2rem;
}

.generator-form {
  margin: 2rem 0;
}

.input-group {
  display: flex;
  gap: 1.5rem;
  justify-content: center;
  margin-bottom: 1.5rem;
}

.input-wrapper {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.input-wrapper label {
  font-size: 0.9rem;
  color: var(--text-light);
  margin-bottom: 0.5rem;
  font-weight: 500;
}

.dimension-input {
  width: 80px;
  padding: 0.75rem;
  border: 1px solid rgba(99, 102, 241, 0.3);
  border-radius: 8px;
  text-align: center;
  font-size: 1rem;
  font-weight: 600;
  color: var(--primary-color);
  transition: all 0.3s ease;
}

.dimension-input:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(99, 102, 241, 0.2);
}

.generate-btn {
  background: var(--primary-color);
  color: white;
  border: none;
  padding: 0.75rem 2rem;
  border-radius: 8px;
  font-size: 1rem;
}
```

```
font-weight: 600;
cursor: pointer;
transition: all 0.3s ease;
display: block;
margin: 0 auto;
width: 200px;
}

.generate-btn:hover {
  background: var(--primary-light);
  transform: translateY(-2px);
}

.generate-btn:disabled {
  background: #cbd5e1;
  cursor: not-allowed;
  transform: none;
}

/* ArrayTable Specific Styles */
.array-table-converter {
  padding: 2rem;
}

.json-input {
  width: 100%;
  padding: 1rem;
  border: 1px solid rgba(99, 102, 241, 0.3);
  border-radius: 8px;
  font-family: 'Fira Code', monospace, 'Inter', sans-serif;
  font-size: 0.9rem;
  resize: vertical;
  min-height: 120px;
  transition: all 0.3s ease;
}

.json-input:focus {
  outline: none;
  border-color: var(--primary-color);
  box-shadow: 0 0 0 3px rgba(99, 102, 241, 0.2);
}

.convert-btn {
  background: var(--secondary-color);
  color: white;
  border: none;
  padding: 0.75rem 2rem;
  border-radius: 8px;
  font-size: 1rem;
  font-weight: 600;
  cursor: pointer;
  transition: all 0.3s ease;
  display: block;
  margin: 1.5rem auto 0;
}
```

```
width: 200px;
}

.convert-btn:hover {
  background: #f97316;
  transform: translateY(-2px);
}

.convert-btn:disabled {
  background: #cbd5e1;
  cursor: not-allowed;
  transform: none;
}

.error-message {
  color: #ef4444;
  font-size: 0.9rem;
  margin-top: 0.5rem;
  text-align: center;
}

/* Shared Table Styles */
.table-wrapper {
  margin-top: 2rem;
  overflow-x: auto;
  border-radius: var(--border-radius);
  box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1);
}

.styled-table {
  width: 100%;
  border-collapse: separate;
  border-spacing: 0;
  border-radius: var(--border-radius);
  overflow: hidden;
  animation: fadeIn 0.5s ease;
}

.styled-table tr:first-child td:first-child {
  border-top-left-radius: var(--border-radius);
}

.styled-table tr:first-child td:last-child {
  border-top-right-radius: var(--border-radius);
}

.styled-table tr:last-child td:first-child {
  border-bottom-left-radius: var(--border-radius);
}

.styled-table tr:last-child td:last-child {
  border-bottom-right-radius: var(--border-radius);
}
```



```
.styled-table td {
  padding: 1rem;
  border: 1px solid rgba(203, 213, 225, 0.5);
  text-align: center;
  transition: all 0.2s ease;
}

.styled-table tr:nth-child(even) td {
  background-color: rgba(241, 245, 249, 0.5);
}

.styled-table tr:hover td {
  background-color: rgba(224, 231, 255, 0.3);
}

.cell-content {
  padding: 0.5rem;
  min-height: 20px;
}

@keyframes fadeIn {
  from { opacity: 0; transform: translateY(10px); }
  to { opacity: 1; transform: translateY(0); }
}

/* Responsive Styles */
@media (max-width: 768px) {
  .glass-card {
    padding: 1.5rem;
  }

  .input-group {
    flex-direction: column;
    gap: 1rem;
  }

  .dimension-input, .generate-btn, .convert-btn {
    width: 100%;
  }
}
```

2D Array to Table Converter

Paste your 2D array in JSON format

Enter a 2D array in JSON format, e.g.: [{"A1", "A2"}, [{"B1", "B2"}]]

Convert to Table

2D Array to Table Converter

Paste your 2D array in JSON format

```
[  
  ["Name", "Age", "State", "Hobby"],  
  ["Aman", 28, "Jharkhand", "Coding"],  
  ["Ashish", 32, "Bihar", "Hiking"],  
  ["Prasenjit", 24, "Tripura", "Gaming"],  
  ["Sumit", 30, "Uttar Pradesh", "Reading"]  
]
```

Convert to Table

Name	Age	State	Hobby
Aman	28	Jharkhand	Coding
Ashish	32	Bihar	Hiking
Prasenjit	24	Tripura	Gaming
Sumit	30	Uttar Pradesh	Reading