2.2 Code Migration Exercise: -

Step 1: Set Up a Sample JavaScript Project

```
= my-js-project/
            - src/
                  - index.js
                   -utils.js
            - package.json
            - .gitignore
            - README.md
```

src/index.js:

```
-   const { Shub } = require('./utils');
    console.log(Shub('World'));
```

src/utils.js:

```
-   function Shub(name)
    {
      return `Hii, ${name}!`;
    }
    module.exports = { Shub };
```

Step 2: Initialize a TypeScript Project

```
cd my-js-project
npm init –y
npx tsc --init
```

Step 3: Install TypeScript and Necessary Packages

```
npm install typescript @types/node --save-dev
```

Step 4: Rename JavaScript Files to TypeScript

Rename your .js files to .ts:

Step 5: Update Configuration Files

```
"scripts":
 {
   "build": "tsc",
   "start": "node dist/index.js"
}
```

Step 6: Update TypeScript Configuration

```
{
  "compilerOptions": {
    "outDir": "./dist",
    "rootDir": "./src",
    "strict": true,
    "esModuleInterop": true
  },
  "include": ["src"]
}
```

Step 7: Add Type Annotations

src/utils.ts:

```
function Shub(name: string): string
{
  return `Hello, ${name}!`;
}
export { Shub };
```

src/index.ts:

```
import { Shub } from './utils';
console.log(Shub('World'));
```

Step 8: Handle Common Conversion Issues

1 Module Imports/Exports:

- JavaScript uses require and module.exports.
- TypeScript uses ES6 modules: import and export.

2.Type Errors:

- If you encounter type errors, adjust your types accordingly.

3.Non-TypeScript Libraries:

- For libraries without type definitions, use @types or create custom .d.ts files.

Step 9: Compile and Run the Project

Compile the TypeScript code:

```
npm run build
```

Run the compiled JavaScript:

```
npm start
```

Example of Resolving Common Issues Handling Dynamic Typing in Functions:

src/utils.ts:

```
function Shub(name: any): string
{
   if (typeof name !== 'string')
{
      throw new Error('name must be a string');
   }
   return `Hello, ${name}!`;
}

export { Shub };
```