App: Entry Point of the application

- **App Context is initialized with values:**
    - **Servers, SetServers: Getters/Setters for the array containing server positions.**
    - **Shards, setShards: Numerical value for the number of shards.**
    - **Vacancy, setVacancy: Vacancy is the array containing the vacant space around the servers, which consists of a object with 2 values:**
        - **Start: starting position of the vacant space**
        - **End: ending position of the vacant space**
    - **Data, setData: Includes array with attributes of all the shards like, type(server,vacant,request) ,etc.**
- **Contains the Menu**
- **Contains the main diagram, with circle, innerCircle and shards.**

Whenever number of shards changes:

- **Servers array resets to contain only one server.**
- **Vacancy array resets to contain only one vacancy(1-n).**

Menu: It contains the actions one can perform on the application which are as follows:

- **Number of shards: It is a range selector, which sets the value of shards using setShards and the shards changes accordingly.**
- **Add Server: A button which creates the new servers.**
- **Delete Server: An option selector where we can select the server we want to delete using the servers array.**

Adding a server:

- **Server position is calculated through a function newServerPlacement, basically it is the appropriate position for the next server calculated using vacancy.**
- **Servers array is updated with the new server position.**
- **Based on the new position:**
    - **If it was vacant: It normally becomes a server and 3 values get initialised:**
        - **maxLoad: This is the max load that server can get.**
        - **currentLoad: This is the percentage of max load which are of type request.**
        - **Requests: An array that will contain the requests served by this server.**
    - **If it was request: It becomes a server and same process followed but before that :**
        - **Next server is found using the servedBy attribute of request and the currentLoad is adjusted, as the request is changed to a server, now the request get dropped and so does the currentLoad.**

- **Loads are managed for the server and the next server, as that will be affected too using a loadManager() function.**
- **Vacancy is also managed using vacancyDivide function.**

Removing a server:

- **As the initial server cannot be deleted, it is checked.**
- **Vacancy is managed using vacancyDivideRemoval function.**
- **Next server is recognised using the vacancy array.**
- **All the requests of this server are redirected to the next server:**
    - **The next server requests array is populated by this servers requests.**
    - **servedBy attribute of all request changes to the next server.**
- **CurrentLoad and MaxLoad of next server is adjusted.**
- **This server is removed from the servers array.**

InsertDivs: This is the component which receives number of shards and create the diagram.

- **Array is created with all the shards initialised as vacant but only the 0th div as server which is looped to create shards and data array is also populated with same.**
- **Angle is calculated for the rotation for each arc according to the number of shards.**
- **Variable is calculated which helps in the shapes of the shards using clip path.(4 should be used, but used 5 because giving better results): but basically 0-100 is the values for a triangle, we adjust the same value according to the shards.**

Arc: The arcs are created based on the angle(for rotation angle) and variable(for shape).

- **Color is managed using the state according to the type:**
    - **If server: black color is used**
    - **If request: blue color is used**
    - **If vacant : random light color is generated using getRandomColor function.**
- **Based on the data type, click handler is used:**
    - **If server: it shows the server name, current load(percentage) and maxLoad(percentage).**
    - **If vacant: It is changed to type request with:**
        - **servedBy attribute initialised.**
        - **The currentLoad of the next server, and the servedBy attribute of this request is managed using currLoadManager function.**
    - **If request:**
        - **servedBy is displayed.**

Utilities:

- **Algorithms:**
  - **upperBound: Basically used to find the next available server of the index provided(if no server found, then 0 is considered as the last server).**
- **Deque: It is created for the vacancy because:**
  - **PushFront: Used when server is deleted, so we get the largest vacancy possible which must be pushed from front.**
  - **PushBack: When new server is created, then we need to delete the front vacancy, as it is used and need to push the two new vacancies at the end.**
- **LoadManager:**
  - **loadManager: Accepts data,vacancy, current position, total shards to manage the current and max load of new server.**
    - **Finds the next server.**
    - **Sets the maxLoad for this and next server using vacancy.**
    - **Manages the requests:**
      - **Requests which are smaller then the new server are served by new server, and also there servedBy attribute is changed to new server.**
      - **Requests which are bigger then the new server are served by the next server only.**
    - **Current Load of new and next server is calculated based on the size of request arrays.**
  - **CurrLoadManager: Accepts data, current position of new request and servers array to manage the current load on the next server:**
    - **Next server position is found using upperBound.**
    - **Current request's servedBy attribute is set to next server.**
    - **Current load of the next server is increased by 1.**
    - **This request's position is pushed in the requests array of next server.**
- **randomColor:**
  - **getRandomColor: A light shade random color is generated using rgb values randomly selected in the light shaded range.**
- **serverUtils:**
  - **newServerPlacement: The position is calculated using the first value of the vacancy(in the center of the vacant area.)**
  - **vacancyDivide: when new server is created, vacancy is adjusted:**
    - **Two vacancies are created which are as follows:**
      - **Starting from the next position of the current server and ending on the previous position of the next server.**
      - **Starting from the next position of the previous server and ending on the previous position of the new server.**
    - **The first vacancy is deleted.**
  - **vacancyDivideRemoval: When server is deleted:**
    - **The two vacancies are removed which were related to the server to be deleted.**
    - **The new vacancy is pushed in the front which will be created after deleting the current server.**