HTML code to display file upload button:

```
<div>
  <input type="file" onChange={(e) => addPath(e)} />
</div>
```

addPath javascript function to get the image buffer from the selected file:

```
const addPath = async (event) => {
  const file = event.target.files[0]
  const buffer = await file.arrayBuffer()
  setUpload(buffer)
}
```

This image buffer will be uploaded to the ipfs (I have used ReactJS framework for the client application development. You may have to make some changes in the code, but the underlying HTML, javascript code will remain same).

Code snippet to include ipfs reference:

```
const ipfsClient = require('ipfs-http-client')
const ipfs = ipfsClient({ host: 'localhost', port: '5001', protocol: 'http' })
```

Function to upload file buffer to ipfs:

```
const uploadFile = async (file) => {
  const fileAdded = await ipfs.add(file)
  if (fileAdded.path != '') {
    return fileAdded.path
  } else {
    console.error('Could not upload the file to ipfs network')
  }
}
```

This method will return the CID (hash) value to uniquely identify your file in ipfs. This Hash value will be stored in the blockchain as like any other parameter value.

javascript function to retrieve image from ipfs:

```
const getImage = (hash) => {
  var strWindowFeatures =
    'location=yes,height=570,width=520,scrollbars=yes,status=yes'
  window.open(
    // 'https://ipfs.io/ipfs/' + hash + '?filename=' + hash,
    'http://localhost:8080/ipfs/' + hash,
    '_blank',
    strWindowFeatures,
  )
}
```