```
import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import axiosInstance from "auth/FetchInterceptor";
import { UPLOAD_FILE } from "constants/ApiConstant";

export const uploadFileURL = createAsyncThunk(
  "uploadfile/uploadFileURL",
  async (data, { rejectWithValue }) => {
    try {
      const { formData, onUploadProgress } = data;
      const response = await axiosInstance.post(UPLOAD_FILE, formData, {
        baseURL: process.env.REACT_APP_HRM_ENDPOINT_URL,
        ContentType: "multipart/form-data",
      });
      if (onUploadProgress) onUploadProgress(response.data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

const initialState = {
  loading: false,
  customerList: [],
};

export const reportSlice = createSlice({
  name: "uploadfile",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(uploadFileURL.pending, (state) => {
        state.loading = true;
      })
      .addCase(uploadFileURL.fulfilled, (state, action) => {
        state.loading = false;
      })
      .addCase(uploadFileURL.rejected, (state, action) => {
        state.loading = false;
      });
  },
});

export const { showLoading } = reportSlice.actions;

export default reportSlice.reducer;
```