

```

import { createSlice, createAsyncThunk } from "@reduxjs/toolkit";
import { cloneDeep } from "lodash";
import ThietLapTuyenDungService from "services/ThietLapTuyenDungService";

export const getAllThietLap = createAsyncThunk(
  "tuyendung/getAllThietLap",
  async (loai, { rejectWithValue }) => {
    try {
      // 1: hình thức làm việc
      // 2 : nội làm việc
      const response = await ThietLapTuyenDungService.GetAllThietLap(loai);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getHinhThucLamViec = createAsyncThunk(
  "tuyendung/getHinhThucLamViec",
  async (loai, { rejectWithValue }) => {
    try {
      // 1: hình thức làm việc
      // 2 : nội làm việc
      const response = await ThietLapTuyenDungService.GetAllThietLap(1);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getNoiLamViec = createAsyncThunk(
  "tuyendung/getNoiLamViec",
  async (loai, { rejectWithValue }) => {
    try {
      // 1: hình thức làm việc
      // 2 : nội làm việc
      const response = await ThietLapTuyenDungService.GetAllThietLap(2);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllKhoaPhong = createAsyncThunk(
  "tuyendung/getAllKhoaPhong",
  async (loai, { rejectWithValue }) => {
    try {
      if (loai === undefined) return null;
      const response = await ThietLapTuyenDungService.GetAllKhoaPhong(loai);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllHinhThucLamViec = createAsyncThunk(
  "tuyendung/getAllHinhThucLamViec",
  async (loai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllThietLap(loai);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getDanhGiaThuViecTL = createAsyncThunk(

```

```

    "tuyendung/getDanhGiaThuViecTL",
    async (Id, { rejectWithValue }) => {
      try {
        const response = await ThietLapTuyenDungService.GetAllThietLap(Id);
        return response.data;
      } catch (err) {
        return rejectWithValue(err.message || "Error");
      }
    }
  );
export const UpSertThietLapTuyenDung = createAsyncThunk(
  "tuyendung/UpSertThietLapTuyenDung",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upSertThietLap(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

//BO TIEU CHI
export const getAllBoTieuChi = createAsyncThunk(
  "tuyendung/getAllBoTieuChi",
  async (loai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllBoTieuChi();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpSertBoTieuChi = createAsyncThunk(
  "tuyendung/UpSertBoTieuChi",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upSertBoTieuChi(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllBoTieuChiChiTiet = createAsyncThunk(
  "tuyendung/getAllBoTieuChiChiTiet",
  async (boTieuChiId, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllBoTieuChiChiTiet(
        boTieuChiId
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpSertBoTieuChiChiTiet = createAsyncThunk(
  "tuyendung/UpSertBoTieuChiChiTietList",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;

```

```

const payload = cloneDeep(data);
delete payload.onSuccess;
const response = await ThietLapTuyenDungService.upSertBoTieuChiChiTiet(
  payload
);
if (onSuccess) onSuccess(response);
return response.data;
} catch (err) {
  return rejectWithValue(err.message || "Error");
}
}
);

export const DeleteDanhGiaUngVienChiTiet = createAsyncThunk(
  "tuyendung/DeleteDanhGiaUngVienChiTiet",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await ThietLapTuyenDungService.DeleteDanhGiaUngVienChiTiet(payload.id);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

//DE XUAT TUYEN DUNG
export const searchDeXuatTuyenDung = createAsyncThunk(
  "tuyendung/searchDeXuatTuyenDung",
  async (data, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.searchDeXuatTuyenDung(
        data
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getDeXuatTuyenDung = createAsyncThunk(
  "tuyendung/getDeXuatTuyenDung",
  async (maDeXuat, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetDeXuatTuyenDung(
        maDeXuat
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllDeXuatTuyenDung = createAsyncThunk(
  "tuyendung/getAllDeXuatTuyenDung",
  async (trangThai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllDeXuatTuyenDung(
        trangThai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const XoaDeXuatTuyenDung = createAsyncThunk(
  "tuyendung/XoaDeXuatTuyenDung",

```

```

    async (data, { rejectWithValue }) => {
      try {
        const { onSuccess } = data;
        const payload = cloneDeep(data);
        delete payload.onSuccess;
        const response = await ThietLapTuyenDungService.XoaDeXuatTuyenDung(
          payload.id
        );
        if (onSuccess) onSuccess(response);
        return response.data;
      } catch (err) {
        return rejectWithValue(err.message || "Error");
      }
    }
  );
export const UpSertDeXuatTuyenDung = createAsyncThunk(
  "tuyendung/UpSertDeXuatTuyenDung",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upSertDeXuatTuyenDung(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const upTrangThai = createAsyncThunk(
  "tuyendung/upTrangThai",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upTrangThai(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

//HO SO UNG VIEN
export const searchHoSoUngVien = createAsyncThunk(
  "tuyendung/searchHoSoUngVien",
  async (data, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.searchHoSoUngVien(data);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllHoSoUngVien = createAsyncThunk(
  "tuyendung/getAllHoSoUngVien",
  async (trangthai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllHoSoUngVien(
        trangthai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

export const getAllHoSoThuViec = createAsyncThunk(
  "tuyendung/getAllHoSoThuViec",
  async (trangthai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.getAllHoSoThuViec(
        trangthai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getHoSoUngVien = createAsyncThunk(
  "tuyendung/getHoSoUngVien",
  async (Id, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetHoSoUngVien(Id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const XoaHoSoUngVien = createAsyncThunk(
  "tuyendung/XoaHoSoUngVien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.XoaHoSoUngVien(
        payload.id
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const GetAllNhanVien = createAsyncThunk(
  "tuyendung/GetAllNhanVien",
  async (Id, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllNhanVien();
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpSertHoSoUngVien = createAsyncThunk(
  "tuyendung/UpSertHoSoUngVien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upSertHoSoUngVien(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpSertHoSoVtNhanVien = createAsyncThunk(
  "tuyendung/UpSertHoSoVtNhanVien",
  async (param, { rejectWithValue }) => {

```

```

    try {
      const response = await ThietLapTuyenDungService.UpSertHoSoVtNhanVien(
        param
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const upTrangThaiHoSoUngVien = createAsyncThunk(
  "tuyendung/upTrangThaiHoSoUngVien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upTrangThaiHoSoUngVien(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

//DANH GIA UNG VIEN
export const searchDanhGiaUngVien = createAsyncThunk(
  "tuyendung/searchDanhGiaUngVien",
  async (trangthai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.SearchDanhGiaUngVien(
        trangthai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getAllDanhGiaUngVien = createAsyncThunk(
  "tuyendung/getAllDanhGiaUngVien",
  async (trangthai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllDanhGiaUngVien(
        trangthai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const getDanhGiaUngVien = createAsyncThunk(
  "tuyendung/getDanhGiaUngVien",
  async (Id, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetDanhGiaUngVien(Id);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const GetAllDanhGiaUngVienChiTiet = createAsyncThunk(
  "tuyendung/GetAllDanhGiaUngVienChiTiet",
  async (BoLoc, { rejectWithValue }) => {
    try {
      const response =
        await ThietLapTuyenDungService.GetAllDanhGiaUngVienChiTiet(BoLoc);
      return response.data;
    }

```

```

    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpSertDanhGiaUngVien = createAsyncThunk(
  "tuyendung/UpSertDanhGiaUngVien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upSertDanhGiaUngVien(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const UpDateDanhGiaChiTiet = createAsyncThunk(
  "tuyendung/UpDateDanhGiaChiTiet",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.UpDateDanhGiaChiTiet(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const DuyetTrungTuyen = createAsyncThunk(
  "tuyendung/DuyetTrungTuyen",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.DuyetTrungTuyen(payload);

      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const XoaDanhGiaUngVien = createAsyncThunk(
  "tuyendung/XoaDanhGiaUngVien",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.XoaDanhGiaUngVien(
        payload
      );

      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

```

```

export const XoaDanhGiaThuViec = createAsyncThunk(
  "tuyendung/XoaDanhGiaThuViec",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.XoaDanhGiaThuViec(
        payload
      );

      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const DuyetChinhThuc = createAsyncThunk(
  "tuyendung/DuyetChinhThuc",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.DuyetChinhThuc(payload);

      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const ChuyenHsThuViec = createAsyncThunk(
  "tuyendung/ChuyenHsThuViec",
  async (BoLoc, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.ChuyenHsThuViec(BoLoc);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

//DANH GIA THU VIEC
export const searchDanhGiaThuViec = createAsyncThunk(
  "tuyendung/searchDanhGiaThuViec",
  async (trangthai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.SearchDanhGiaThuViec(
        trangthai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getAllDanhGiaThuViec = createAsyncThunk(
  "tuyendung/getAllDanhGiaThuViec",
  async (trangthai, { rejectWithValue }) => {
    try {
      const response = await ThietLapTuyenDungService.GetAllDanhGiaThuViec(
        trangthai
      );
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);

export const getDanhGiaThuViec = createAsyncThunk(

```



```

    "tuyendung/getDanhGiaThuViec",
    async (Id, { rejectWithValue }) => {
      try {
        const response = await ThietLapTuyenDungService.GetDanhGiaThuViec(Id);
        return response.data;
      } catch (err) {
        return rejectWithValue(err.message || "Error");
      }
    }
  );
export const GetAllDanhGiaThuViecChiTiet = createAsyncThunk(
  "tuyendung/GetAllDanhGiaThuViecChiTiet",
  async (BoLoc, { rejectWithValue }) => {
    try {
      const response =
        await ThietLapTuyenDungService.GetAllDanhGiaThuViecChiTiet(BoLoc);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpSertDanhGiaThuViec = createAsyncThunk(
  "tuyendung/UpSertDanhGiaThuViec",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.upSertDanhGiaThuViec(
        payload
      );
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const UpDateDanhGiaThuViecChiTiet = createAsyncThunk(
  "tuyendung/UpDateDanhGiaThuViecChiTiet",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response =
        await ThietLapTuyenDungService.UpDateDanhGiaThuViecChiTiet(payload);
      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const DuyetTrungTuyenThuViec = createAsyncThunk(
  "tuyendung/DuyetTrungTuyenThuViec",
  async (data, { rejectWithValue }) => {
    try {
      const { onSuccess } = data;
      const payload = cloneDeep(data);
      delete payload.onSuccess;
      const response = await ThietLapTuyenDungService.DuyetTrungTuyen(payload);

      if (onSuccess) onSuccess(response);
      return response.data;
    } catch (err) {
      return rejectWithValue(err.message || "Error");
    }
  }
);
export const ChuyenHsChinhThuc = createAsyncThunk(
  "tuyendung/ChuyenHsChinhThuc",

```

```

    async (BoLoc, { rejectWithValue }) => {
      try {
        const response = await ThietLapTuyenDungService.ChuyenHsThuViec(BoLoc);
        return response.data;
      } catch (err) {
        return rejectWithValue(err.message || "Error");
      }
    }
  );

```

```

const initialState = {
  loading: false,
  ThietLapTuyenDungList: [],
  HinhThucLamViecList: [],
  NoiLamViecList: [],
  DanhGiaThuViecTLList: [],
  BoTieuChiList: [],
  BoTieuChiChiTietList: [],
  DeXuatTuyenDungList: [],
  DeXuatTuyenDungAll: [],
  KhoaPhongList: [],
  DeXuatTuyenDungDetail: {},
  HoSoUngVienList: [],
  HoSoUngVienSearch: [],
  HoSoUngVienDetail: {},
  GetAllNhanVienList: [],
  DanhGiaUngVienList: [],
  DanhGiaThuViecAll: [],
  DanhGiaUngVienChiTietList: [],
  DanhGiaThuViecList: [],
  DanhGiaThuViecDetail: {},
  DanhGiaThuViecChiTietList: [],
};

```

```

export const ThietLapTuyenDungSlice = createSlice({
  name: "tuyenDung",
  initialState,
  reducers: {
    showLoading: (state) => {
      state.loading = true;
    },
  },
  extraReducers: (builder) => {
    builder
      .addCase(getHinhThucLamViec.pending, (state) => {
        state.loading = true;
      })
      .addCase(getHinhThucLamViec.fulfilled, (state, action) => {
        state.loading = false;
        state.HinhThucLamViecList = action.payload;
      })
      .addCase(searchDanhGiaThuViec.fulfilled, (state, action) => {
        state.loading = false;
        state.DanhGiaThuViecList = action.payload;
      })
      .addCase(getAllThietLap.fulfilled, (state, action) => {
        state.loading = false;
        state.ThietLapTuyenDungList = action.payload;
      })
      .addCase(getAllDeXuatTuyenDung.fulfilled, (state, action) => {
        state.loading = false;
        state.DeXuatTuyenDungAll = action.payload;
      })
      .addCase(GetAllDanhGiaThuViecChiTiet.fulfilled, (state, action) => {
        state.loading = false;
        state.DanhGiaThuViecChiTietList = action.payload;
      })
      .addCase(getHinhThucLamViec.rejected, (state, action) => {
        state.loading = false;
      })

      .addCase(getNoiLamViec.pending, (state) => {
        state.loading = true;
      })
  }
});

```

```

.addCase(getNoiLamViec.fulfilled, (state, action) => {
  state.loading = false;
  state.NoilamViecList = action.payload;
})
.addCase(getDanhGiaThuViecTL.fulfilled, (state, action) => {
  state.loading = false;
  state.DanhGiaThuViecTLList = action.payload;
})
.addCase(getDanhGiaThuViec.fulfilled, (state, action) => {
  state.loading = false;
  state.DanhGiaThuViecDetail = action.payload;
})
.addCase(getNoiLamViec.rejected, (state, action) => {
  state.loading = false;
})

.addCase(getDanhGiaUngVien.fulfilled, (state, action) => {
  state.loading = false;
  state.DanhGiaUngVienDetail = action.payload;
})
.addCase(getAllThietLap.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getDeXuatTuyenDung.fulfilled, (state, action) => {
  state.loading = false;
  state.DeXuatTuyenDungDetail = action.payload;
})
.addCase(GetAllDanhGiaUngVienChiTiet.fulfilled, (state, action) => {
  state.loading = false;
  state.DanhGiaUngVienChiTietList = action.payload;
})
.addCase(getHoSoUngVien.fulfilled, (state, action) => {
  state.loading = false;
  state.HoSoUngVienDetail = action.payload;
})

.addCase(GetAllNhanVien.fulfilled, (state, action) => {
  state.loading = false;
  state.GetAllNhanVienList = action.payload;
})
.addCase(UpSertThietLapTuyenDung.pending, (state) => {
  state.loading = true;
})
.addCase(getAllKhoaPhong.fulfilled, (state, action) => {
  state.loading = true;
  state.KhoaPhongList = action.payload;
})
.addCase(UpSertThietLapTuyenDung.fulfilled, (state, action) => {
  state.loading = false;
})
.addCase(UpSertThietLapTuyenDung.rejected, (state, action) => {
  state.loading = false;
})
.addCase(getAllBoTieuChi.pending, (state) => {
  state.loading = true;
})
.addCase(getAllBoTieuChi.fulfilled, (state, action) => {
  state.loading = false;
  state.BoTieuChiList = action.payload;
})
.addCase(searchHoSoUngVien.fulfilled, (state, action) => {
  state.loading = false;
  state.HoSoUngVienSearch = action.payload;
})
.addCase(getAllHoSoUngVien.fulfilled, (state, action) => {
  state.loading = false;
  state.HoSoUngVienList = action.payload;
})
.addCase(getAllHoSoThuViec.fulfilled, (state, action) => {
  state.loading = false;
  state.DanhGiaThuViecAll = action.payload;
})
.addCase(getAllBoTieuChi.rejected, (state, action) => {
  state.loading = false;
}

```

```

    })
    .addCase(getAllBoTieuChiChiTiet.pending, (state) => {
      state.loading = true;
    })
    .addCase(getAllBoTieuChiChiTiet.fulfilled, (state, action) => {
      state.loading = false;
      state.BoTieuChiChiTietList = action.payload;
    })
    .addCase(getAllBoTieuChiChiTiet.rejected, (state, action) => {
      state.loading = false;
    })

    .addCase(searchDeXuatTuyenDung.pending, (state) => {
      state.loading = true;
    })
    .addCase(searchDeXuatTuyenDung.fulfilled, (state, action) => {
      state.loading = false;
      state.DeXuatTuyenDungList = action.payload;
    })
    .addCase(searchDeXuatTuyenDung.rejected, (state, action) => {
      state.loading = false;
    })

    .addCase(getAllDanhGiaUngVien.fulfilled, (state, action) => {
      state.loading = false;
      state.DanhGiaUngVienList = action.payload;
    })
    .addCase(searchDanhGiaUngVien.fulfilled, (state, action) => {
      state.loading = false;
      state.DanhGiaUngVienList = action.payload;
    });
  },
});

export const { showLoading, setHisInfo } = ThietLapTuyenDungSlice.actions;

export default ThietLapTuyenDungSlice.reducer;

```