

# PGR: UI Implementation - Guidelines & FAQs

## Enable PGR

Modules are enabled by MDMS config at <https://github.com/egovernments/egov-mdms-data/blob/DEV/data/pb/tenant/citymodule.json>

Create a new branch for the state if already doesn't exist from the master of repo [egovernments/digit-ui](#).

To add and enable any module in the new UI, `src/App.js` needs to be changed.

We export only the init function of the module to take care of all the initializations.

```
1import { PGRModule, PGRLinks, PGRReducers } from "@egovernments/digit-ui-module-pgr";
2
3const enabledModules = ["PGR"];
4window.Digit.ComponentRegistryService.setupRegistry({
5  PGRLinks,
6  PGRModule,
7});
8
```

## Changing CSS

Refer: [DIGIT UI: Implementation - Development Guidelines & FAQs](#)

## Customizing fields in a form

First, add any new component created in the registry,

```
1Digit.ComponentRegistryService.setupRegistry({
2  SelectName: SelectName
3});
4
```

Complaint form - `Digit.Customizations.PGR.complaintConfig`

Default config is available at [default complaintConfig](#)

Create a config similar to the following

```
1export const config = {
2  routes: {
3    "complaint-type": {
4      nextStep: "pincode",
5    },
6    landmark: {
7      nextStep: "apartment",
8    },
9    apartment: {
10     component: "SelectName",

```

```

11  texts: {
12    header: "Apartment or Society",
13    cardText: "CS_COMPLAINT_SUBTYPE_TEXT",
14    submitBarLabel: "PT_COMMONS_NEXT",
15  },
16  inputs: [
17    {
18      label: "Apartment",
19      type: "text",
20      name: "additionalDetails.apartment",
21      validation: {
22        minLength: 6,
23        maxLength: 7,
24      },
25      error: "CORE_COMMON_PINCODE_INVALID",
26    },
27  ],
28  nextStep: "upload-photos",
29 },
30 },
31};

```

Any new field should be part of additionalDetails

### Customizing views

Complaint details table on the details page of any complaint. –

Digit.Customizations.PGR.getComplaintDetailsTableRows

The function has the following params available passed by callee:

- id: complaint id
- service: complaint response object
- role: currently logged in user role CITIZEN or EMPLOYEE
- t: function used for translation

```

1getComplaintDetailsTableRows: ({ id, service, role, t }) => {
2  if (role === "CITIZEN") {
3    return {
4      CS_COMPLAINT_DETAILS_COMPLAINT_NO: id,
5      CS_COMPLAINT_DETAILS_APPLICATION_STATUS: `CS_COMMON_${service.applicationStatus}`,
6      CS_ADDCOMPLAINT_COMPLAINT_TYPE: `SERVICEDEFS.${complaintType}`,
7      CS_ADDCOMPLAINT_COMPLAINT_SUB_TYPE:
8      `SERVICEDEFS.${service.serviceCode.toUpperCase()}`,
9      CS_COMPLAINT_ADDITIONAL_DETAILS: service.description,
10     CS_COMPLAINT_FILED_DATE:
11     Digit.DateUtils.ConvertTimestampToDate(service.auditDetails.createdTime),
12     ES_CREATECOMPLAINT_ADDRESS: [
13       service.address.landmark,
14       Digit.Utils.locale.getLocalityCode(service.address.locality, service.tenantId),
15       service.address.city,
16       service.address.pincode,
17     ],
18   };
19 }

```

```
18 return {};  
19 }
```

### **FAQ**

In addition to any [FAQs](#) available in the main implementation document.