SoP Document for Onboarding on OpenForge

Document Version - 1.0

OpenForge

OpenForge (https://openforge.gov.in) is part of the larger Digital India initiative, which seeks to transform India into a digitally empowered society and knowledge economy. In 2015, when the Government of India rolled out the "Policy on Collaborative Application Development by Opening the Source Code of Government Applications", the Ministry of Electronics and Information Technology (MeitY) initiated an open-source project known as OpenForge.

OpenForge 2.0 is the next evolutionary step in the development of the code repository platform, offering a host of new features and capabilities that surpass its predecessor, OpenForge. Provision to recognize and appreciate developers' contributions will be helpful for fostering a sense of achievement and motivation.

Any agency onboarded with the government should use OpenForge as a source code repository and software life cycle management tool. It may be noted that OpenForge does not host any application related data.

Prerequisites

Before using OpenForge, ensure the following:

1) Account Setup:
   - All users must have an active account on OpenForge.
2) Access Permissions:
   - Appropriate access levels (Admin, Developer, Viewer) must be assigned.
3) Tools/Software Required:
   - Git (for version control)
   - IDE (Integrated Development Environment)

System Setup

1) Initial Setup
   - Install Git on your machine (if not already installed).
   - Register for an OpenForge account.
   - Clone the repository
2) Accessing the Project Repository

- Use your OpenForge credentials to log in to the platform.
- Navigate to your project repository.
- Ensure your repository is properly synced with the local environment.

Advantages of OpenForge Platform

1. To ensure strategic control in e-Governance applications and systems from a long-term perspective.
2. To provide a platform for maintaining code repositories and version control for government source code.
3. To promote a culture of open collaborative application development between public agencies and private organizations, citizens and institutions.
4. To reduce development cycles and fasten the rollout of e-governance applications in the country.
5. To deliver e-governance services and solutions of higher quality and security.

How does OpenForge work?

OpenForge allows two possible collaboration models:

- **Government to Community (G2C):** In G2C, projects can be created in the public mode by either government or community members. Subject to the approval of the project administrators, anyone can participate in these projects and contribute to open collaborative source code development. The administrators can choose to configure the project access settings in whatever way suits them.
- **Government to Government (G2G):** In G2G, projects can be created in a controlled mode by government agencies/members only. Membership to these projects will be on approval basis (by the project administrators) and restricted to government entities only.

A special case of G2G is when a particular government department wants a project for private use without any external sharing (i.e. use OpenForge's code repository and version control for internal departmental development). Please note that this private use-case is only allowed for government entities.

Types of Projects on OpenForge

OpenForge is targeted specifically at e-governance applications. Any project (applications, frameworks, libraries, SDKs, APIs, datasets, components, plugins etc) which deals with governance or civic areas is welcome. Applications without any apparent governance overlap are

not allowed on the platform. OpenForge reserves the right to decide on the suitability of a particular application for OpenForge.

OpenForge Features

OpenForge is an open-source, all-in-one software development and project management platform. It supports agile methodologies, continuous integration, and collaboration among teams. Here are the key features of OpenForge:

1. Source Code Management (SCM)
- Git and Subversion (SVN): Built-in support for Git and SVN repositories.
- Code Browsing: View code directly within the platform, with versioning support.
- Branch and Merge Support: Facilitates branch management and merges.

2. Task Tracking
-Bug and Task Tracking: Tracks issues, bugs, enhancements, and tasks with customizable workflows.
- Custom Fields: Create custom fields to adapt to different project needs.
- Notifications and Alerts: Email and in-app notifications for issue updates.

3. Agile Project Management
- Kanban Boards: Provides visual management for task tracking.
- Scrum Support: Includes backlog management, sprint planning, and burndown charts.
- Epics and User Stories: Organizes tasks hierarchically for clear workflows.

4. Document Management
- Wikis and Files: Stores project documents, wikis, and collaboration notes.
- Version Control: Full versioning for documents, ensuring the latest updates are tracked.

5. Planning and Time Tracking
- Gantt Charts: Provides visual planning with Gantt charts.
- Milestones and Deadlines: Track project milestones and deadlines.
- Effort Estimation and Time Tracking: Time-tracking tools for accurate effort estimation.

6. Reporting and Analytics
- Project Metrics: Track project progress, tasks, and workload distribution.
- Burndown Charts: Track sprint progress with real-time data.
- Custom Reports: Generate custom reports for project analysis.

7. Customization and Flexibility

- Custom Workflows: Design workflows tailored to team needs.
- Custom Dashboards: Personalized dashboards with drag-and-drop widgets.
- Plugins: Extend functionality through available plugins.

8. Security and Permissions

- Role-Based Access Control (RBAC): Define access control at various levels (project, team, user).
- Audit Logs: Keeps track of user actions for compliance and security purposes.

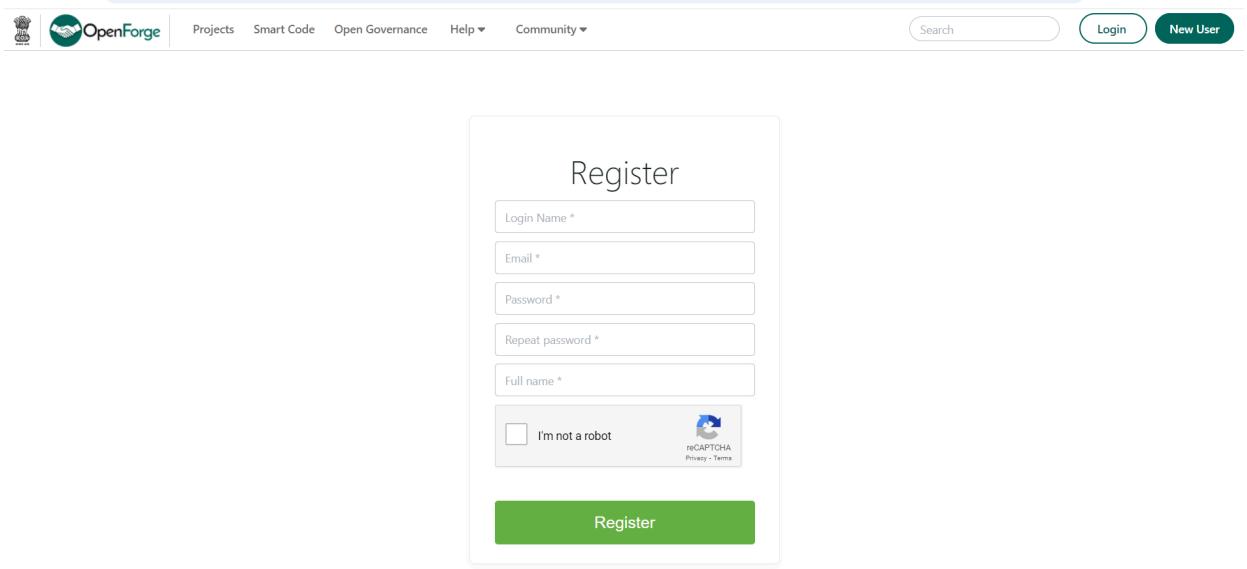
9. Multi-Project Support

- Manage multiple projects in a single instance with centralized user and project management.

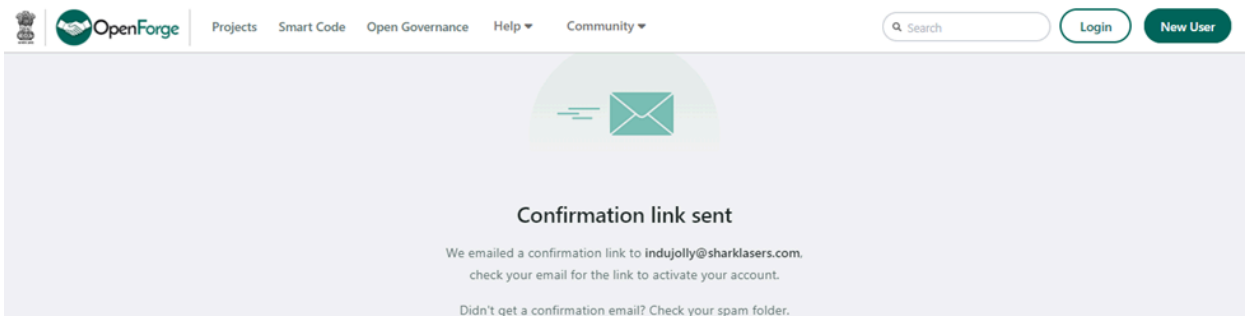
OpenForge provides flexibility and control for teams practicing Agile, DevOps, and continuous integration, making it a great choice for software development environments.

SoP for new project onboarding i.e. Project Registration and Repository Management

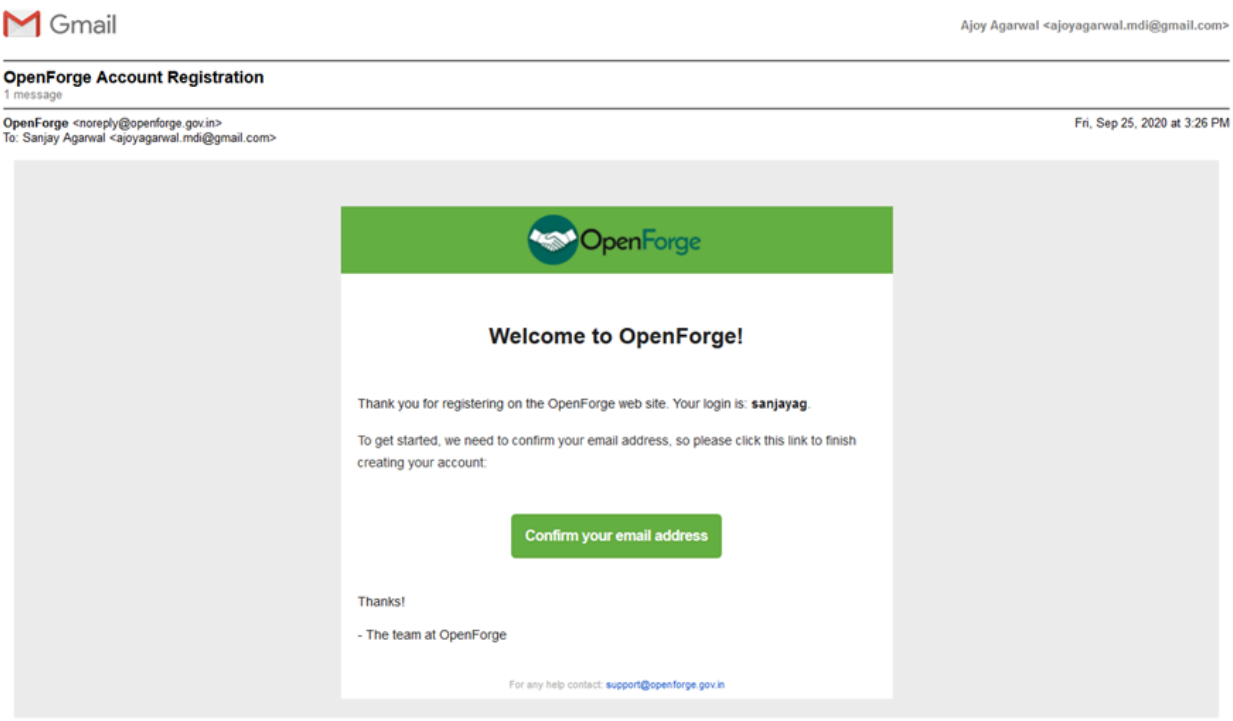
1. A user can register on OpenForge through a simple sign up process. Users have to click on the "NEW USER" link provided on the portal for first time registration. Minimal information such as the full name of the user and email ID has to be provided. User has to create login credentials such as username and password which he will use for subsequent logins. It is preferred to use government email ID



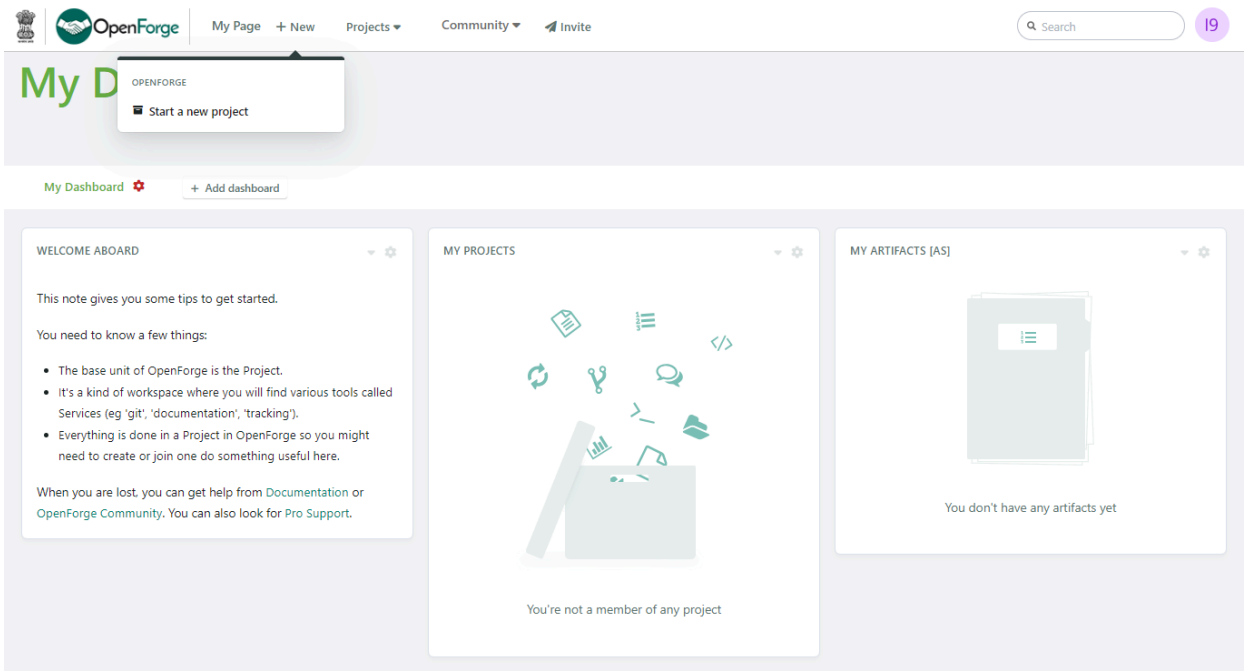
2. Once the user clicks on the "Register" button, a verification email is sent to the user on his registered email ID provided during registration. Please check the spam folder for email if you are using gmail ID.



3. User has to verify his email ID by clicking on the "Confirm your email address" button provided in the verification email. Once done, the user will be taken to the Openforge login page for the first time. Once the user logs in successfully, he becomes a member of the openForge platform and can do subsequent login with his login credentials.

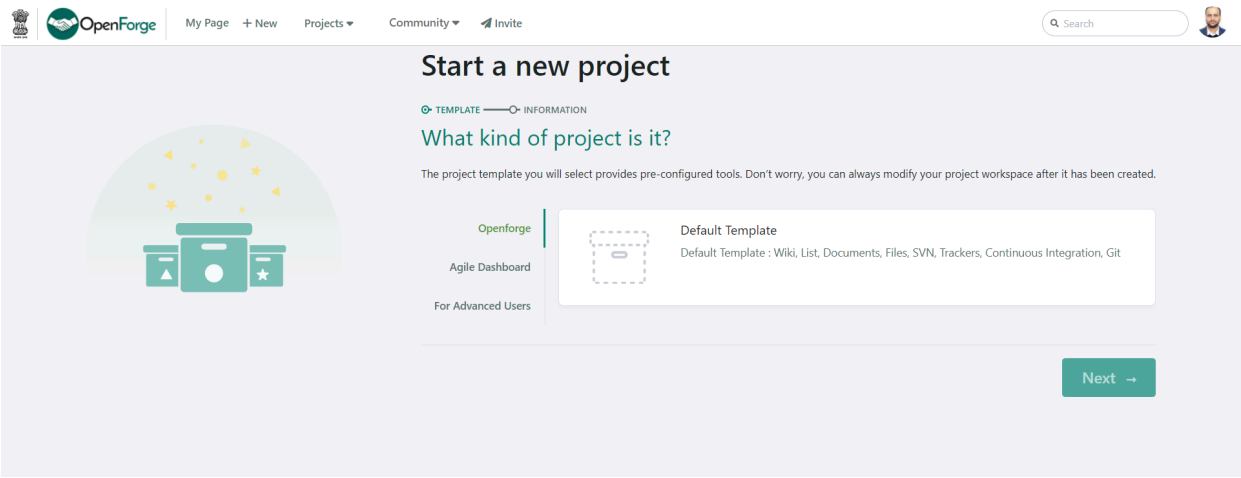


4. Once a user logs in, he/she will be able to register a project on the platform. Users have to select “New” from the top menu bar as illustrated in the below screenshot.

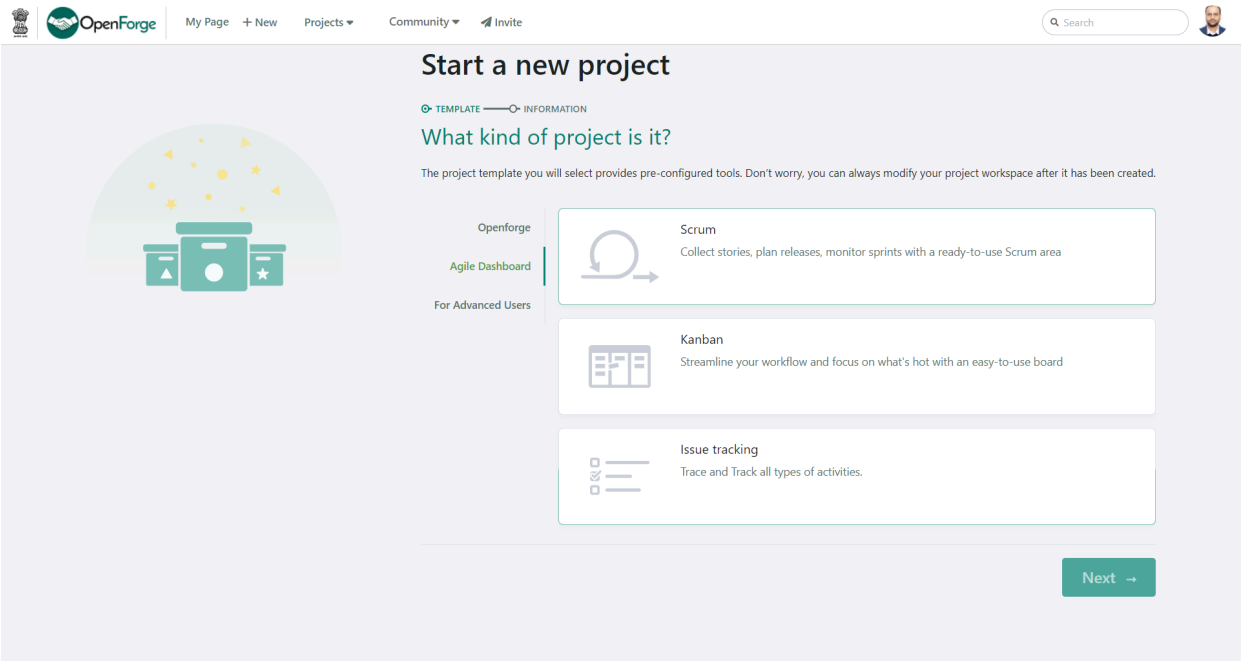


5. Users can select the default template of OpenForge or she may also select Agile Dashboard for the project template.

a. Select Default template project if project is following waterfall model.

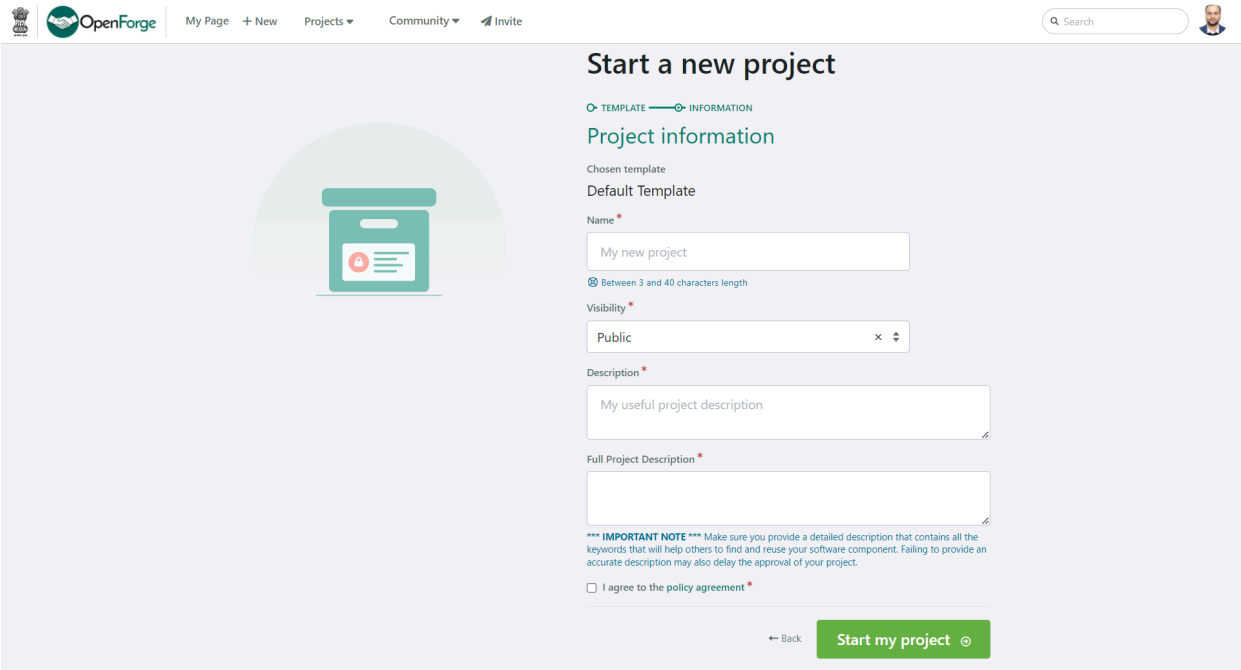


b. Select Agile Dashboard if following scrum or Kanban for project development. .



6. User has to provide basic details of the project such as Name of the project, Short Description and Full Project Description. It is important to have a meaningful short and full description of the project so that interested guests/members get meaningful information when

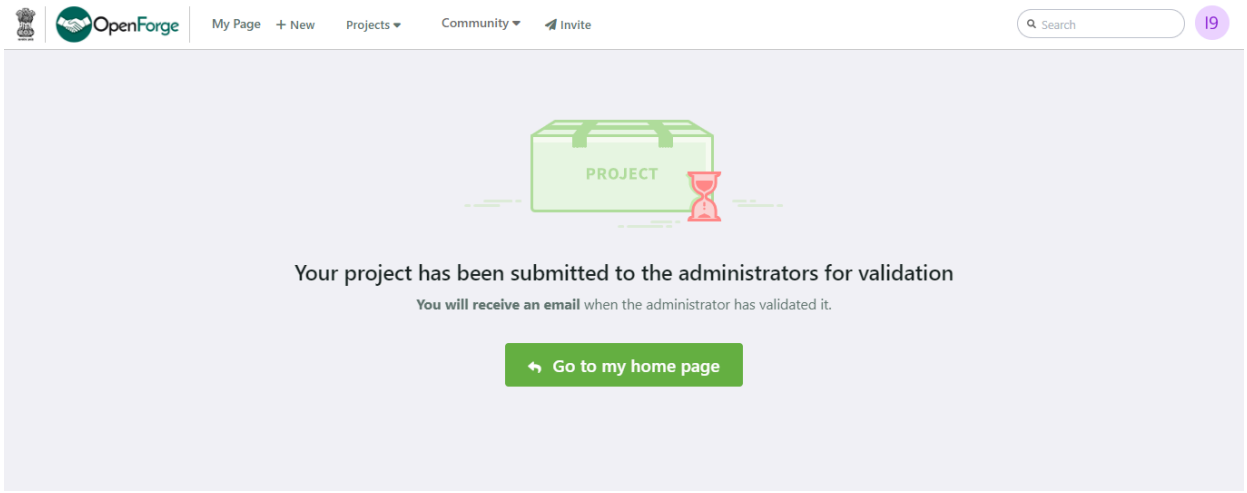
they browse these projects on the portal. It is recommended to Select “Visibility” as “Public” so that the projects is discoverable by everyone.



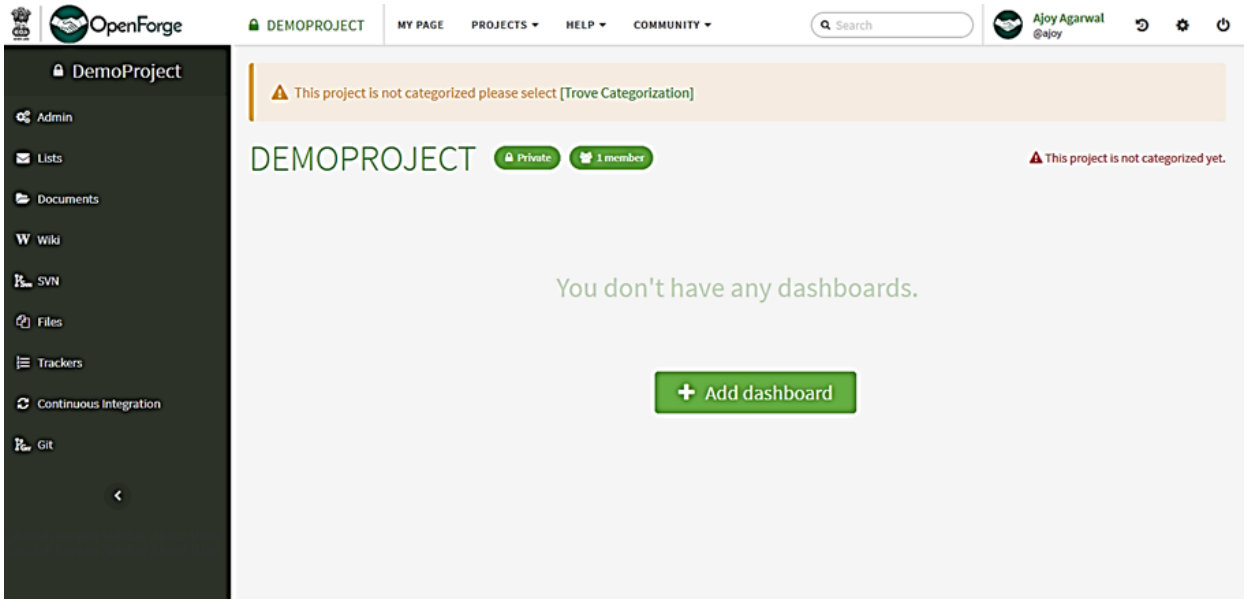
Short Description provides one liner information of the project for a guest whenever he browses the list of projects on the portal.

Full Project Description provides a summary information of the project whenever the user browses the list of projects on the portal and clicks on a certain project.

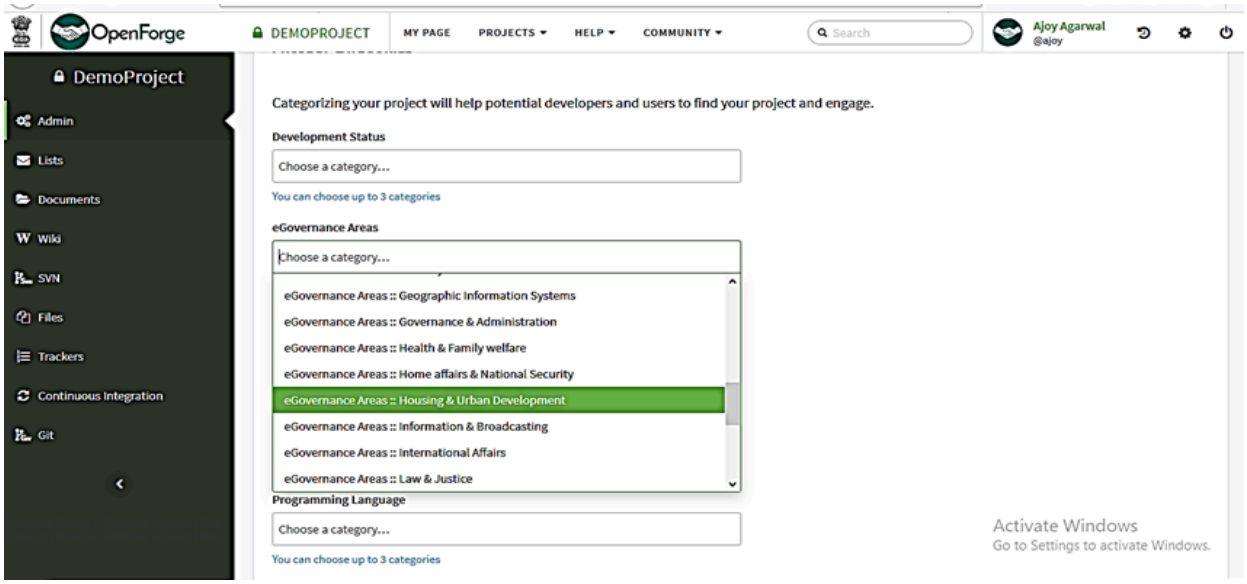
7. Once the user clicks on the “Create the project” button, it is sent to the administrator for approval. The users who have used government email IDs, their projects are auto approved by the system.



8. Once your project is approved by the administrator of the portal, please categorize it since it helps in easy browsing and cataloging of the project. The creator of the project is by default the administrator of the project.



9. Please categorize your project as per the categorization applicable.

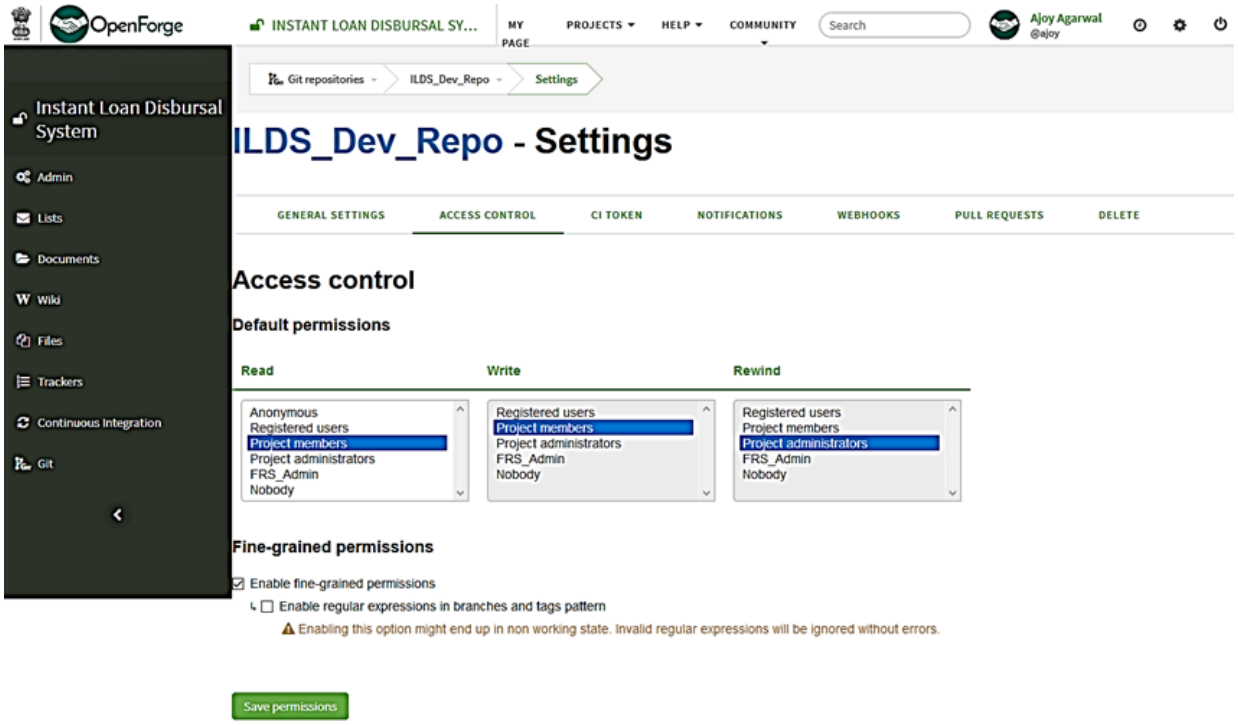


10. If a project is to be made available only to the selected group of people for downloading and making changes, please change the settings of the git repo to make it available to project members only or the group you form within project members. Please see the changes you will make in such a case.

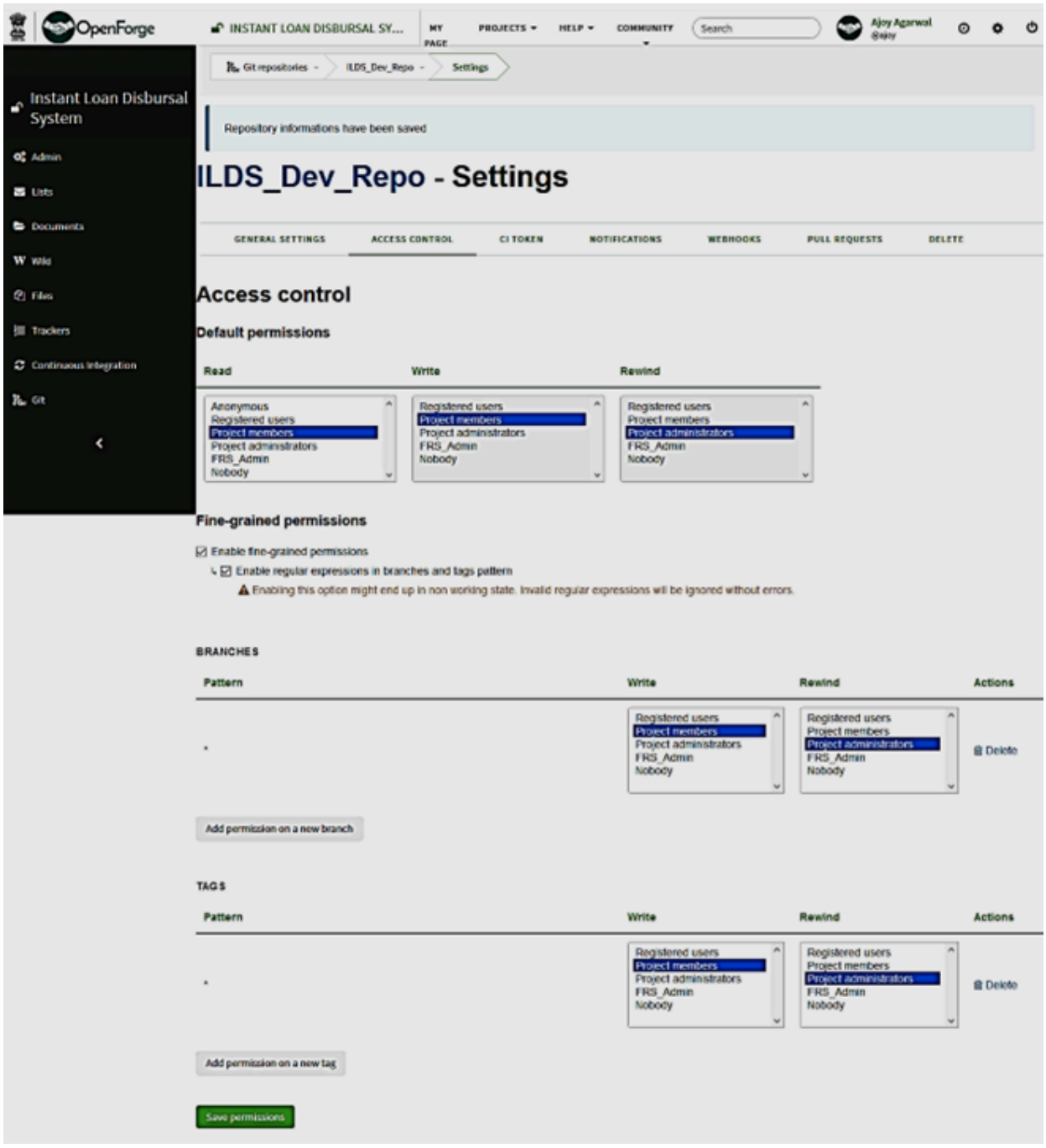
    a. If you want to make code downloadable for everyone(Source Open) but do not want any code commits outside your project members, select **Read-> Registered users, Write -> Project members, Rewind-> Project administrators.**

    b. If you want to make code downloadable only to project members (but do not want any code commits outside your project members), select **Read-> Project members, Write -> Project members, Rewind-> Project administrators**. Project admin can also create sub groups inside the project and provide access as per their requirement to different repos in the project.

    Granular level permission till branch level can be provided to the set of groups. Click on the checkbox against “Enable fine-grained permissions” and “Enable regular expressions in branches and tags pattern” in the Fine-grained permissions section at the bottom of the page.



Please see the below screen for fine grained permission as reference.



SOP for developers for using OpenForge using GIT

1. Before working on new feature /bug fixing, tasks should be created with detailed descriptions on Openforge or task tracker that you manage at your end. Every task should have a unique task ID.

2. A new branch should be created by every developer for each new task/bug fix every time. The branch name should contain a task ID.

3. For every commit, put a meaningful commit message so that while visiting the branch history/referring to a particular commit, the purpose of every commit is very clear to everyone.

4. The work done in one branch by one developer should be reviewed by a peer/senior developer by pulling the branch in his local area. Openforge provides Pull Request Feature just like github for this purpose. So, pull requests should be made by the developer.

5. Once the branch code is reviewed, the requested branch can be merged easily by Pull Request Feature to the parent branch.

6. Every developer should pull code from the develop/parent branch into his local branch everyday morning before starting work on his branch. The local branch should always be kept updated in this way.

7. Before leaving office, all local code should be committed to the developers own branch and that branch should be pushed/updated to Openforge.

8. For static files such as css, html, images keep a separate repository. Since these files are not changed very frequently, it should not be a part of the everyday check in check out process.

9. Maintain version.txt in each repo for keeping track of release notes and versions.

10. At any point of time the master branch should have the latest code running on production.

11. Make use of the .gitignore file for keeping native changes at the local machine. It avoids accidental overwriting of configuration files at server level.

12. When the developers branch is merged to develop/master and code has gone live on production, the branches should be deleted from Openforge since those are temporary branches.