

# Basic Documentation

Basic Documentation contains insights for the various files and functionality of LabelBuddy.

1. label\_buddy/media: This folder contains all the media data that are stored and created during the use of LabelBuddy.
  - a. **audio**: folder that stores the audio tracks that are uploaded as tasks for annotation.
  - b. **data**: folder that stores the temporary wave form files extracted from the audiowaveform program during an audio track upload.
  - c. **datasets**: folder that stores training and validation datasets for the docker container - LabelBuddy interaction.
  - d. **file\_testing**: folder that stores the model weight files from a new prediction model addition to be tested for validation.
  - e. **images**: folder that stores the static images to be used in the various Label Buddy pages etc.
  - f. **model\_weights**: folder that stores the model weight files from a new prediction model addition.
  - g. **model\_yaml\_files**: folder that stores the yaml files from a new prediction model addition using Docker Containers
2. label\_buddy/dockerization: folder that contains the containerization files for a set of pre-loaded models:
  - a. [PANNs](#)
  - b. [musicnn](#)
  - c. [YOHO](#)
3. label\_buddy/htmlcov: folder that contains the files for the testing coverage to be presented in a live server format.
4. label\_buddy/label\_buddy: folder that contains the asgi, settings, urls and wsgi python files for the corresponding app configuration.
5. label\_buddy/project: folder that contains all the files revolving around the project django model.
  - a. **migrations**: folder that contains all the django generated data about the database migrations.
  - b. **model\_weights**: folder that is used to store the model weight files associated with the project model.
  - c. **admin.py**: python file that contains information about the way the project and prediction models will be displayed in the admin site.
  - d. **forms.py**: python file that contains information about the django forms that are used for projects or prediction models.
  - e. **helpers.py**: python file that contains helper functions for the overall procedures of project and prediction model.
  - f. **models.py**: python file that contains information about the way the project and prediction models will be stored in the database.
  - g. **permissions.py**: python file that contains information about the rest framework permission in project and prediction models.

- h. **serializers.py**: python file that contains information about the rest framework serialization in project and prediction models.
  - i. **urls.py**: python file that contains information about the url and path django configuration of project and prediction models.
  - j. **views.py**: python file that contains information about the rest framework django views functionality of project and prediction models.
6. label\_buddy/static: folder that contains the static source code in CSS and JavaScript for the functionality of the various pages.
    - a. **css**: folder that contains all the css files.
    - b. **images**: folder that contains all the images for the rendering of the pages.
    - c. **js**: folder that contains all the JavaScript files.
  7. label\_buddy/taks: folder that contains a similar format with the Project and Users models.
  8. label\_buddy/templates: folder that contains the static source code in HTML for the functionality of the various pages.
  9. label\_buddy/tests: folder that contains all the unit test files that for the abovementioned coverage.
  10. label\_buddy/users: folder that contains a similar format with the Project and Tasks models.
  11. .coverage: file that contains the information about the unit test coverage.
  12. .coveragerc: file that contains the information about the way the unit test coverage will be executed.
  13. db.sqlite:file that contains information about the database storage.
  14. manage.py: Manage.py in Django is a command-line utility that works similar to the django-admin command. It is your tool for executing many Django-specific tasks