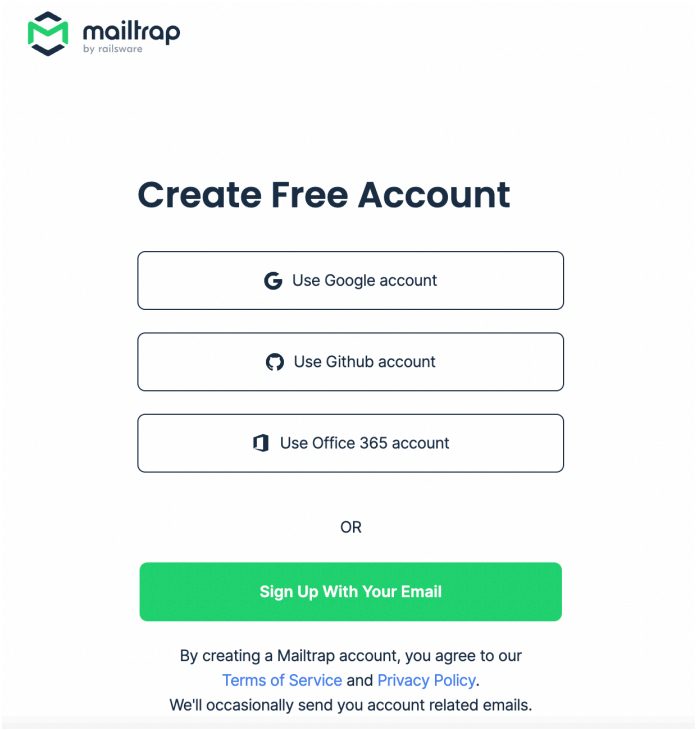
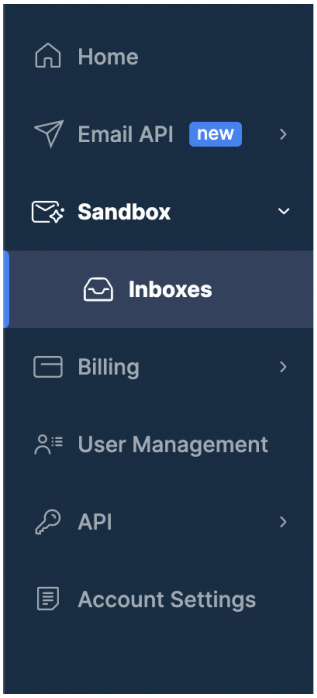
Setting up mail server

1. Go to https://mailtrap.io/
2. Click “sign up” button if you don’t have an account
3. Create a free account

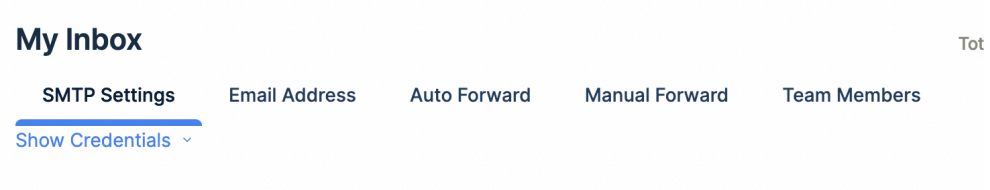


4. Go to Sandbox – inboxes at the side dashboard

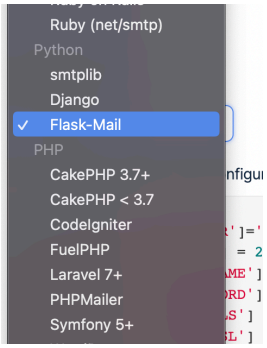


5. Click at the inbox you want to use

6. Go to SMTP settings



7. Choose Flask-Mail at Integrations



8. Copy the setting below and paste it to __init__.py under create_app()

```python
def create_app():
    app = Flask(__name__)
    app.config['SECRET_KEY'] = "secret keys"
    #Set your own sql server url based on the user you create and the database you use
    app.config['SQLALCHEMY_DATABASE_URI'] = "sqlite:///app.db"
    app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
    #Setup Your own smtp mail server here
    #app.config['MAIL_SERVER']=''
    #app.config['MAIL_PORT'] =
    #app.config['MAIL_USERNAME'] =
    #app.config['MAIL_PASSWORD'] =
    #app.config['MAIL_USE_TLS'] =
    #app.config['MAIL_USE_SSL'] =
    #mail.init_app(app)
    db.init_app(app)
    login_manager.init_app(app)
    from .main import main as main_blueprint
    app.register_blueprint(main_blueprint)
    return app
```

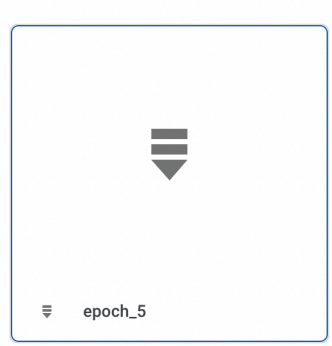
9. Uncomment the sending email function in view.py

```python
#Generate confirmation token
#Uncomment this if you set up your mail server
#token = user.generate_confirmation_token()
#send_mail(sender = "admin@appname",templates = 'email/confirm',to = user.email, user = user, token = token)
flash("Create Account successful. A confirmation email has been sent to your email")
return redirect(url_for("main.login", create_account = True))
```

10. Run the app

Setting up machine learning model

1. Go and download model checkpoints
   https://drive.google.com/drive/folders/1e3pajKB8ogWH-BynGu3k2B6rOedngEma?usp=sharing



2. Create a folder inside app and put the checkpoints there
3. Edit the hard code path for loading checkpoints

```
#Fill the checkpoints path according to your path where you put the model checkpoints
checkpoints = torch.load("resources/epoch_5", map_location='cpu')
model.load_state_dict(checkpoints['model_state_dict'])
model.eval()
```