Project On

Road Defect Management

By

Vaibhav Satish Kubre

(CSE-SYA-71)

Introduction

As when Roads are constructed government delegates this task to private constructors. After the construction roads need to be maintained and here comes the role of Road Inspection officer and Road Maintenance agency. Road Inspection officer are responsible for keeping eyes on the maintenance required for the road. Usually a citizen submits a report if he/she sees a road where maintenance is required. After that Road Inspection Officer reviews that report and if it requires maintenance then forwards that report to the Road Maintenance Agency, Which then takes charge of this process and start working on the maintenance of the road.

The problems with this approach are that not every citizen is eager to make descriptive report with location and photos with it and finding how to submit a report to whom to submit. After that user may did not provide all the information and manual inspection may be needed on that case. There are plenty of problems with this approach and as issues grows maintenance required for the road also grows and in that situation following this lengthy process may become cumbersome for everyone in this cycle.

Road Defect Management overcomes this all issues with providing digital solution to all. Citizens have an android app which they can create their profile and submit report for maintenance. That report will be sent directly to the Road Inspection officer which then he can review, as app makes it compulsory and easy to click the photos and submit them, Road Inspection Officer will always have better reference and description of the issue. Road Inspection Officer have an online dashboard which they can login into and review all the reports. They can decide whether to forward this issue to Road Maintenance Agency or decline it. If it's declined then issue would still be saved for documentation purposes. If it's forwarded then now Road Maintenance Agency can see the report and all the data with it and start working on it. In this entire process all the submitted data is saved and citizen who reported the issue can always track the status of the report.

Tech Stack

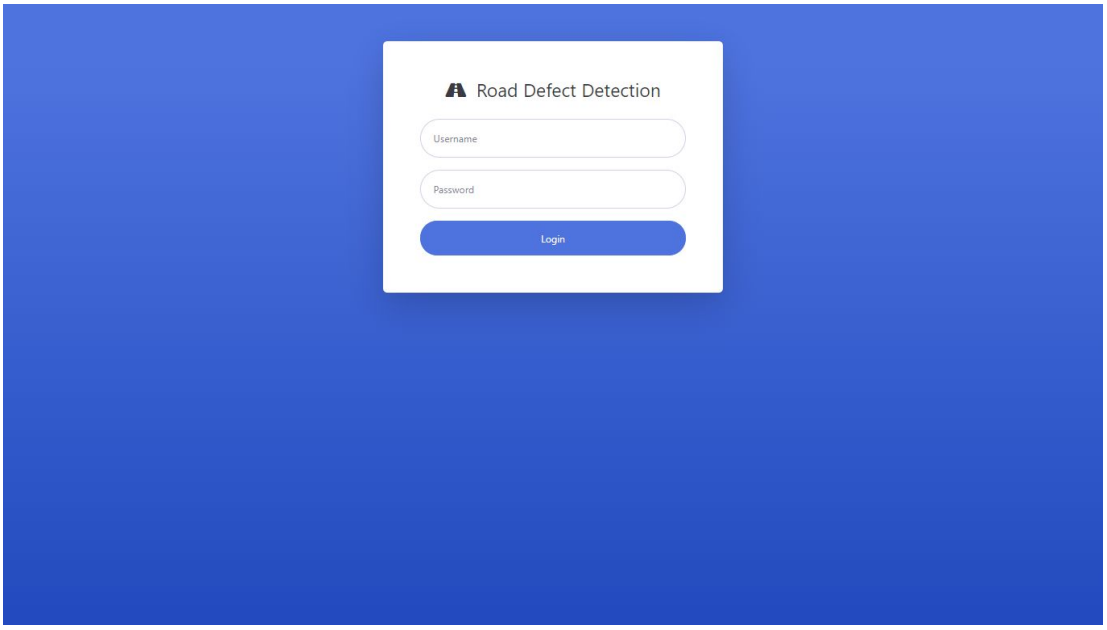
Server:

1) Python 3
2) Flask and SQLAlchemy
3) HTML and JS

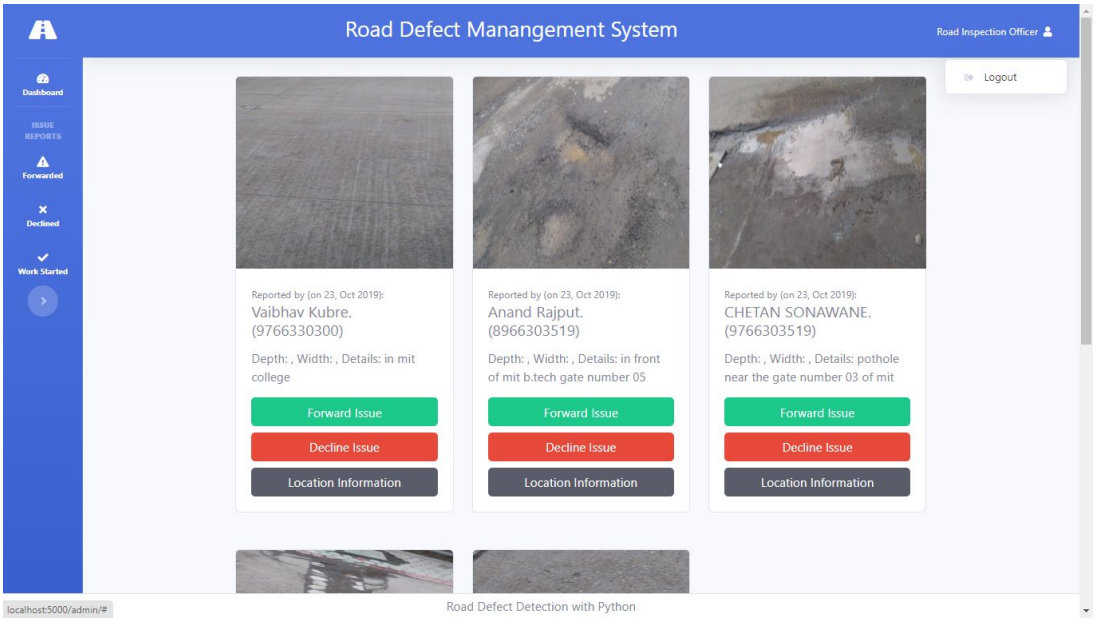
Client:

1) Android (Kotlin)
2) Retrofit2 and Glide

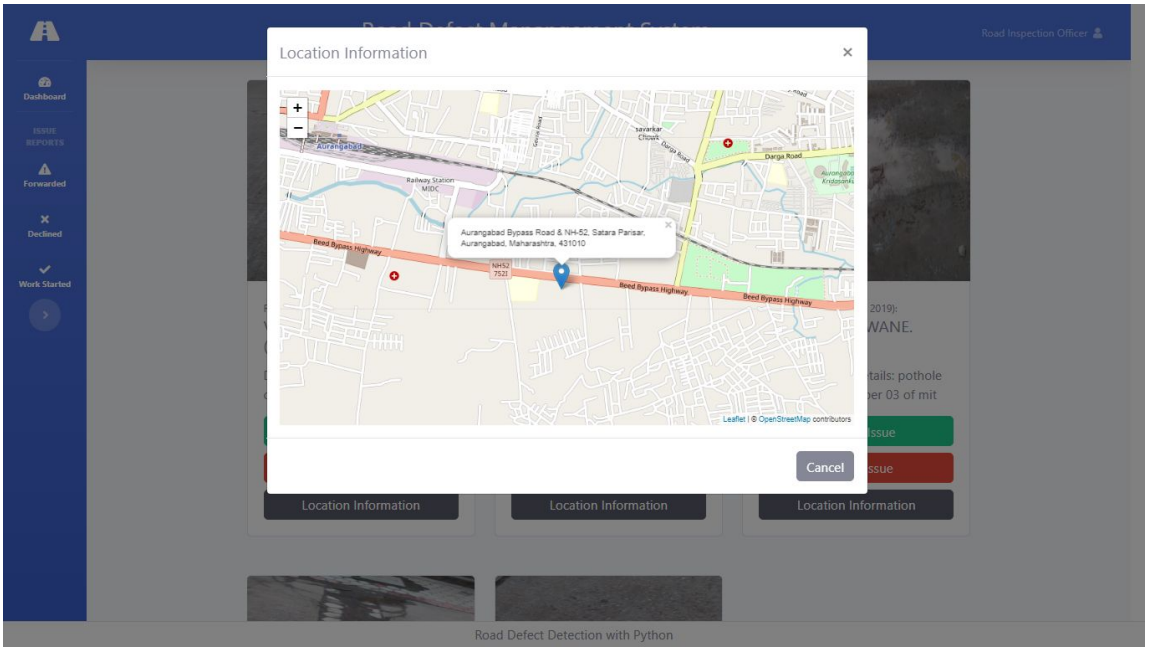
Screenshots



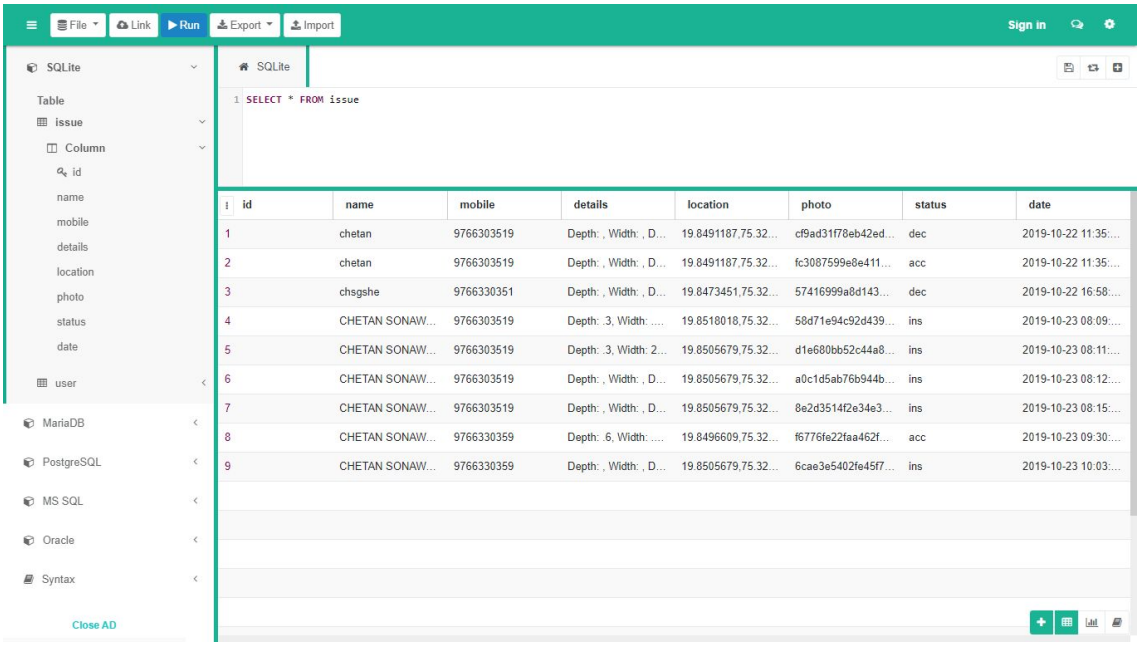
1) Login Screen



2) Dashboard



3) Information on Report



4) Database

Code

```
import os
import uuid
from flask import render_template, request, url_for, redirect, session, jsonify
from sqlalchemy import desc as sql_desc
from werkzeug.utils import secure_filename

from rddserver import app, bcrypt, db
from rddserver.models import User, Issue, issues_schema, issue_schema

def middleware_auth():
        if 'username' not in session:
        return redirect(url_for('login'))

@app.route('/')
def index():
        return render_template('index.html')

@app.route('/login', methods=['GET', 'POST'])
def login():
        if request.method == "POST":
        username = request.form['username']
        password = request.form['password']
        if username and password:
                user = User.query.filter_by(username=username).first()
                if user and bcrypt.check_password_hash(user.password, password):
                session['username'] = user.username
                session['name'] = user.name
                session['role'] = user.role
                return redirect(url_for('admin'))
        return render_template('login.html', error="Incorrect username or
password!")
        return render_template('login.html')

@app.route('/admin/', methods=['GET'], defaults={'page': 1})
```

```
@app.route('/admin/<int:page>', methods=['GET'])
def admin(page):
        middleware_auth()
        per_page = 6
        if session.get('role') == User.ROLE_INSPECTOR:
        issues = Issue.query.filter_by(status=Issue.STATUS_INSPECTOR).order_by(
                sql_desc(Issue.date)).paginate(page, per_page, error_out=False)
        return render_template('inspector.html', issues=issues)
        elif session.get('role') == User.ROLE_MAINTAINER:
        issues = Issue.query.filter_by(status=Issue.STATUS_MAINTAINER).order_by(
                sql_desc(Issue.date)).paginate(page, per_page, error_out=False)
        return render_template('maintainer.html', issues=issues)

@app.route('/reports-forwarded/', methods=['GET'], defaults={'page': 1})
@app.route('/reports-forwarded/<int:page>', methods=['GET'])
def reports_forwarded(page):
        middleware_auth()
        per_page = 6
        issues = Issue.query.filter_by(status=Issue.STATUS_MAINTAINER).order_by(
        sql_desc(Issue.date)).paginate(page, per_page, error_out=False)
        return render_template('reports.html', issues=issues)

@app.route('/reports-work-started/', methods=['GET'], defaults={'page': 1})
@app.route('/reports-work-started/<int:page>', methods=['GET'])
def reports_wip(page):
        middleware_auth()
        per_page = 6
        issues = Issue.query.filter_by(status=Issue.STATUS_ACCEPTED).order_by(
        sql_desc(Issue.date)).paginate(page, per_page, error_out=False)
        return render_template('reports.html', issues=issues)

@app.route('/reports-declined/', methods=['GET'], defaults={'page': 1})
@app.route('/reports-declined/<int:page>', methods=['GET'])
def reports_declined(page):
        middleware_auth()
```

```
        per_page = 6
        issues = Issue.query.filter_by(status=Issue.STATUS_DECLINED).order_by(
        sql_desc(Issue.date)).paginate(page, per_page, error_out=False)
        return render_template('reports.html', issues=issues)

@app.route('/logout')
def logout():
        session.pop('username', None)
        session.pop('role', None)
        return redirect(url_for('login'))



@app.route('/issue', methods=['POST'])
def add_issue():
        name = request.form['name']
        mobile = request.form['mobile']
        details = request.form['details']
        latitude = request.form['latitude']
        longitude = request.form['longitude']
        photo = request.files['photo']

        filename, ext = secure_filename(photo.filename).rsplit('.', 1)
        rnd = uuid.uuid4().hex
        filename = f"{rnd}.{ext}"
        photo.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        location = f"{latitude},{longitude}"
        new_issue = Issue(name=name,
                        mobile=mobile,
                        details=details,
                        location=location,
                        photo=filename)
        db.session.add(new_issue)

        db.session.commit()
        return issue_schema.jsonify(new_issue)
```

```
@app.route('/issue/<mobile>', methods=['GET'])
def get_issues_by_number(mobile):
        issues = Issue.query.filter_by(
        mobile=mobile).order_by(sql_desc(Issue.date)).all()
        result = issues_schema.dump(issues)

        return jsonify(result)

@app.route('/issue/forward', methods=["POST"])
def forward_request():
        id = request.form['id']

        issue = Issue.query.filter_by(id=id).first()
        issue.status = Issue.STATUS_MAINTAINER
        db.session.commit()
        return redirect(url_for('admin'))

@app.route('/issue/decline', methods=["POST"])
def accpet_request():
        id = request.form['id']

        issue = Issue.query.filter_by(id=id).first()
        issue.status = Issue.STATUS_DECLINED
        db.session.commit()
        return redirect(url_for('admin'))

@app.route('/issue/accpet', methods=["POST"])
def decline_request():
        id = request.form['id']

        issue = Issue.query.filter_by(id=id).first()
        issue.status = Issue.STATUS_ACCEPTED
        db.session.commit()
        return redirect(url_for('admin'))
```

```
@app.route('/maps/<float:latitude>/<float:longitude>')
def maps(latitude, longitude):
        return render_template("maps.html", latitude=latitude, longitude=longitude)
```

HTML Code

```
{% extends "layouts/base.html" %}

{% block bodytags %}
class="bg-gradient-primary"
{% endblock %}

{% block content %}
<div class="container">

        <!-- Outer Row -->
        <div class="row justify-content-center">

        <div class="col-xl-6 col-lg-12 col-md-9">

                <div class="card o-hidden border-0 shadow-lg my-5">
                <div class="card-body p-0">
                        <!-- Nested Row within Card Body -->
                        <div class="row">
                        <div class="col-lg-12">
                                <div class="p-5">
                                <div class="text-center">
                                        <h1 class="h4 text-gray-900 mb-4">
                                        <i class="fas fa-road"></i>
                                         Road Defect Detection
                                        </h1>
                                </div>
                                <form action="/login" method="post" class="user">
                                        <div class="form-group">
```

```
                                    <input type="text" class="form-control
form-control-user" name="username"
                                           id="username"
                                           aria-describedby="username"
placeholder="Username">
                                    </div>
                                    <div class="form-group">
                                    <input type="password" class="form-control
form-control-user" name="password"
                                           id="password"
                                           placeholder="Password">
                                    </div>
                                    <button type="submit" class="btn btn-primary
btn-user btn-block">
                                    Login
                                    </button>

                                    {% if error %}
                                    <div class="bg-danger shadow text-white
text-center p-2 mt-4 mb-2 rounded-sm">
                                    {{ error }}
                                    </div>
                                    {% endif %}

                                </form>
                                </div>
                            </div>
                            </div>
                        </div>
                        </div>
                </div>
                </div>
        </div>
        </div>
</div>
{% endblock %}
```