

# Demonstration of The Student Project Management System

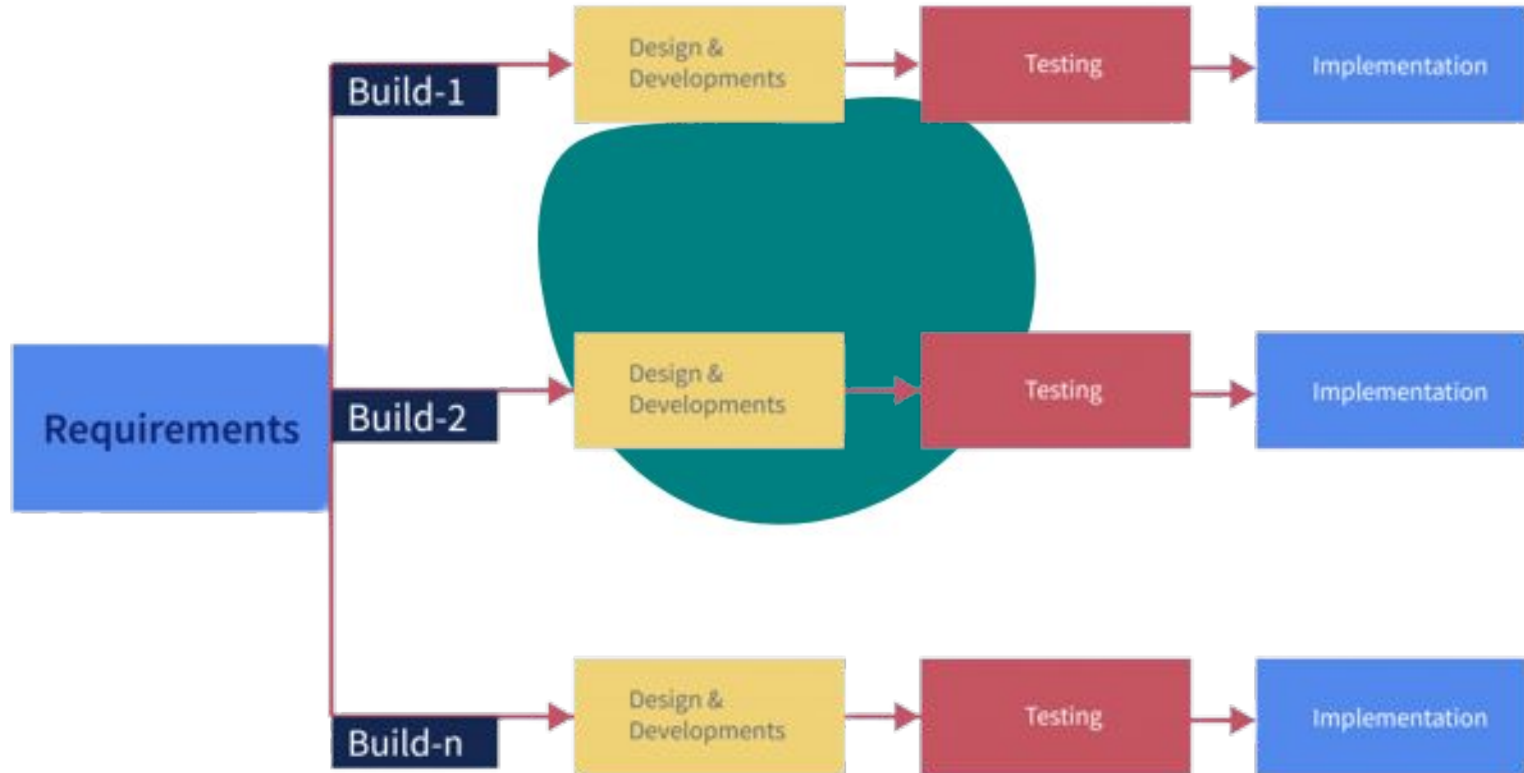
By Shabb03



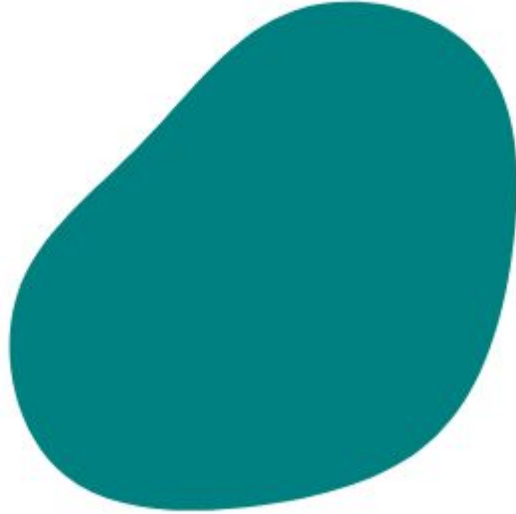
# Tools



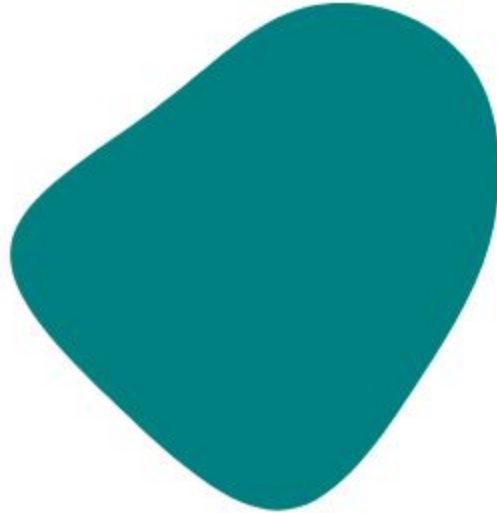
# Incremental Lifecycle Model



# Walkthrough



# Questions?



# What I would do differently

## Tools

React or Angular instead of Node

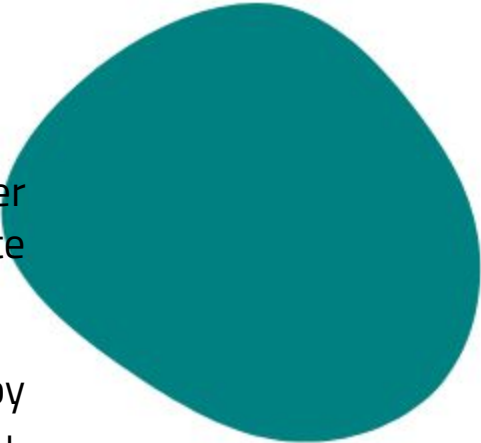
## Documenting

Document in a more neater manner with id's, sections of the project, date and time.

Perhaps automate the process by creating a webpage, Test + Method + Screenshots/Videos

## Overall

Better understanding of Django, API's and Javascript



## Plan Ahead

Make Diagrams of the Project and present them during a meeting to ensure I know what I am doing and what I have left out

## Testing

Develop test cases before starting the project rather than testing during implementation and development

# Timetable

**Chromosome:** <Student, [Assessors], Date, Time, Location>

## 1.Initial Population

Start off with a random sized population of chromosomes

## 3.Selection

Select the top 4 chromosomes with the highest fitness value

## 5.Mutation

Change the genes of some chromosomes to create new parents



## 2.Fitness Function

Check for conflicts between dates, times, locations and assessors

## 4.Crossover

Cut two parents chromosomes at point and swap the genes

# Delete Users

127.0.0.8000/admin



← → ↻ 🏠 127.0.0.1:8000/admin/project/user/

## Django administration

Home > Project > Users

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**  
Groups + Add

**PROJECT**  
Accepted + Add  
Assessorss + Add  
Deadlines + Add  
Demonstration items + Add  
Demonstrations + Add  
Filess + Add  
Markings + Add  
Meetings + Add  
Professors + Add  
Proposals + Add  
Statuss + Add  
Students + Add  
Timetables + Add  
Users + Add  
Venuess + Add

Select user to change

Action:  Go 0 of 17 selected

- ☐ USER
- ☐ fritz3
- ☐ janemurphy
- ☐ markbaker3
- ☐ studentsix
- ☐ professor6
- ☐ professor5
- ☐ studentfive
- ☐ testuser2
- ☐ studentfour
- ☐ testuser
- ☐ professorthree
- ☐ studentthree
- ☐ studenttwo
- ☐ professortwo
- ☐ professor
- ☐ student

← → ↻ 🏠 127.0.0.1:8000/admin/project/user/7/change/

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**  
Groups + Add

**PROJECT**  
Accepted + Add  
Assessorss + Add  
Deadlines + Add  
Demonstration items + Add  
Demonstrations + Add  
Filess + Add  
Markings + Add  
Meetings + Add  
Professors + Add  
Proposals + Add  
Statuss + Add  
Students + Add  
Timetables + Add  
Users + Add  
Venuess + Add

admin | log entry | Can view log entry  
auth | group | Can add group  
auth | group | Can change group  
auth | group | Can delete group  
auth | group | Can view group  
auth | permission | Can add permission  
Specific permissions for this user. Hold down "Control", or "Command" on a Mac, to select multiple.

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Username:

First name:

Last name:

Email address:

☐ Staff status  
Designates whether the user can log into this admin site.

☒ Active  
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Date joined: Date: 2023-01-21 Today | 📅  
Time: 17:53:13 Now | ⌚

User type:

Delete



# Delete Data

## Delete Data

This will delete all data excluding registered users

Enter Message: "I confirm to Delete all data"

Delete Data

```
1 class DeleteAllSerializer(serializers.ModelSerializer):
2     class Meta:
3         model = Accepted
4         fields = ['message']
5
6     def create(self, validated_data):
7         message = validated_data['message']
8
9         request = self.context.get('request', None)
10        if request:
11            if message != "I confirm to Delete all data":
12                raise serializers.ValidationError({'Error': 'Incorrect Message'})
13
14            Proposal.objects.all().delete()
15            Files.objects.all().delete()
16            Accepted.objects.all().delete()
17            Meeting.objects.all().delete()
18            Status.objects.all().delete()
19            Venues.objects.all().delete()
20            Marking.objects.all().delete()
21            Timetable.objects.all().delete()
22            Demonstration.objects.all().delete()
23            DemonstrationItem.objects.all().delete()
24            Assessors.objects.all().delete()
25            return (['message': "System Data Cleared"])
26        return None
```