

MAHENDRA ENGINEERING COLLEGE WOMEN

Category: INTERNET OF THINGS

A PROJECT REPORT

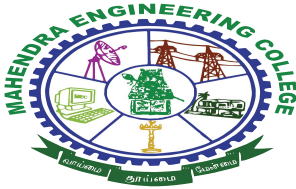
Submitted by

T. Sneka (TEAM LEADER)

S. Sathiyabama

P. Sivasakthi

C. Tamilarasi



ECHNOLOGTECHN

COMPUTER SCIENCE AND ENGINEERING

NALAIYATHIRAN2022TeamId:PNT2022TMID30395

PROJECT GUIDES

Industry Mentor: Mr. Dinesh

Faculty Mentor: J P KAVITHA

INDEX

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose2.LITE

RATURESURVEY

2.1 Existingproblem

2.2 Problem Statement

Definition3.IDEATION&PROPOSEDSO

LUTION

3.1 EmpathyMapCanvas

3.2 Ideation&Brainstorming

3.3 ProposedSolution

3.4 Problem Solution

fit4.REQUIREMENTANALYS

IS

4.1 Functionalrequirement

4.2 Non-Functional

requirements5.PROJECTDESIGN

5.1 DataFlowDiagrams

5.2 Solution&TechnicalArchitecture

5.3 UserStories

6.PROJECTPLANNING&SCHEDULING

6.1 SprintPlanning&Estimation

6.2 SprintDeliverySchedule

7. CODING & SOLUTION

7.1 Feature1

8.CONCLUSION

9.FUTURE SCOPE

10. APPENDIX

10.1 SourceCode

1.

INTRODUCTION**1.1 ProjectOverview:****Smart crop protection system**

Smart crop protection system solutions use sensors placed in crop yields to measure humidity ,temperarture , moisture and to notify farmers when crops are ready to be emptied. Over time, historical data collected by sensors can be used to identify crop patterns . The cost of these sensors is steadily decreasing, making IoT crop protection more feasible to implement and more attractive to farmer.

1.2 Purpose:

- At present, we can see crop are being damaged due to many reasons. Our primary goal is to protect the crop from being damaged .
- Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield.
- In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops .
- So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

2.

LITERATURESURVEY**2.1 ExistingProblem:**

- At present, we can see crop are being damaged due to many reasons. Our primary goal is to protect the crop from being damaged.
- ❖ Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield.
- ❖ In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops
- ❖ So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

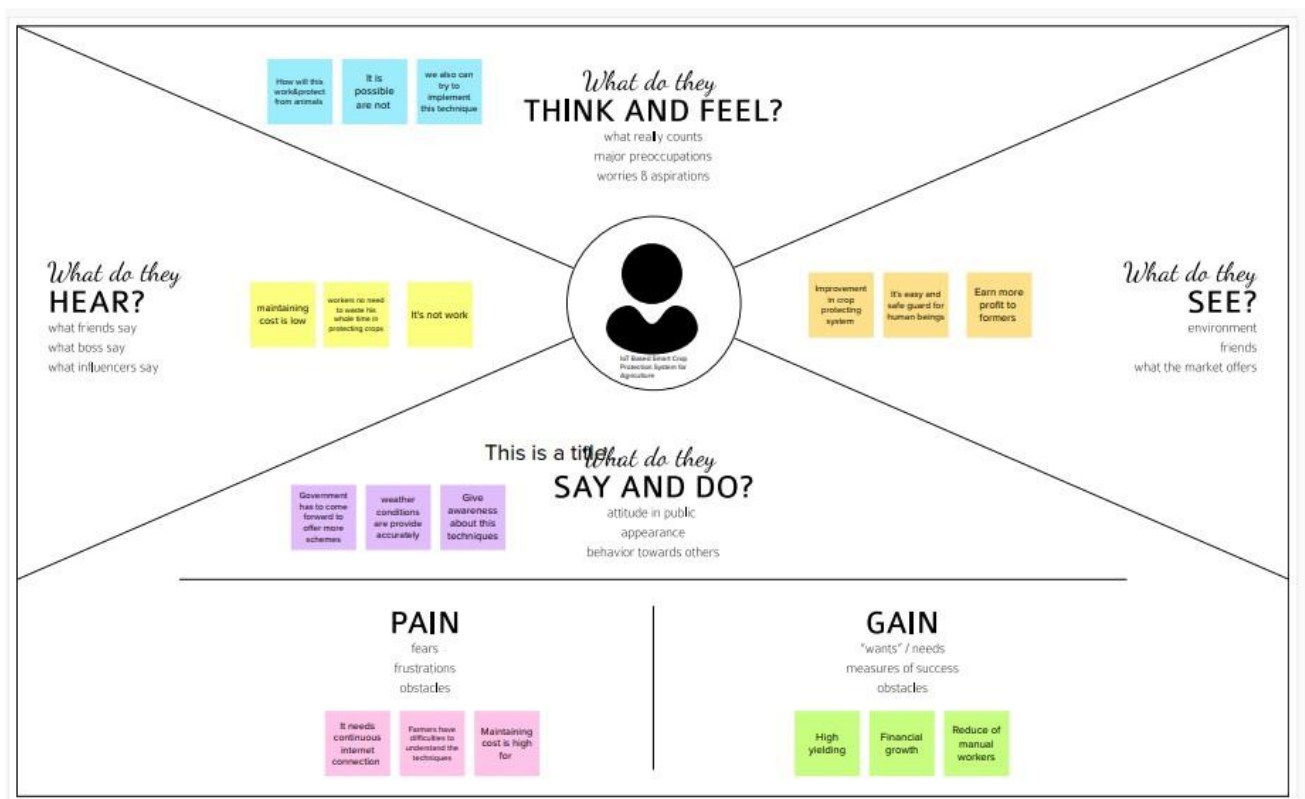
2.2 ProblemStatement Definition:

- Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield .In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops .So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

IDEATION&PROPOSEDSOLUTION

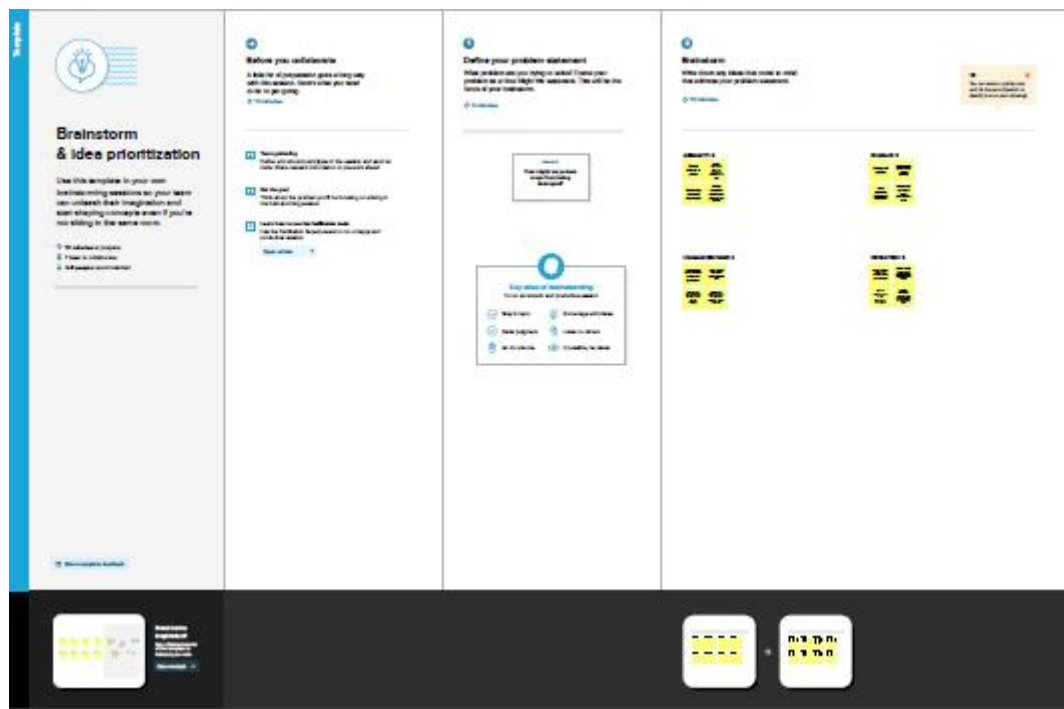
3.1 EmpathyMapCanvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to help teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation&Brainstorming:

Ideation and Brainstorming Ideation is often closely related to the practice of brainstorming, a **specific technique that is utilized to generate new ideas**. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



Group Ideas

Take some sharing cards from the following section or select notes as you go. Once all cards are shared, then proceed. One card from a partner's list should be shared to the group as a whole, by hand and if you can't find a card to share, ask the group.

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

Prioritize

Use your decision list to do the same page about what's important, making choices. Place priorities on the grid according to what's most important and what's most likely.

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

3.3 ProposedSolution:

S.No.	Parameter	Description
1.	ProblemStatement(Problemto be solved)	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
2.	Idea/Solutiondescription	The aim of the proposed work is to develop an IOT device for smart crop field monitoring system and automated irrigation system using the wireless sensor networks(WSN) . To create an IOT device for monitoring the crop field using sensors (soil moisture ,temperature ,Humidity ,etc.,) To automate the irrigation by comparing the level of soil moisture with the threshold value .
3.	Novelty/Uniqueness	Daily update about the Condition of the land send to the farmers via mail
4.	Social Impact /CustomerSatisfaction	<ul style="list-style-type: none"> • Cost effective to the society • Modernization to the society • High protection and High yield

5.	Business Model(Revenue Model)	Outcome based model Data based model Platform based model
6.	ScalabilityoftheSolution	Start small and build out

3.4 PROBLEMSOLUTION FIT:

Problem-Solution Fit canvas		IOT based Smart Crop Protection for Agriculture		Version 1.0	
1 to CL	1. CUSTOMER SEGMENT(S) CS Our customers are farmers who are affected by damage of crops due to various reasons like insect attacks, animal invasion, Excess water flow, etc.	6. CUSTOMER LIMITATIONS CL <small>EG. BUDGET, DEVICES</small> Customer must have minimum knowledge about using the technology.	5. AVAILABLE SOLUTIONS AS <small>PLUSES & MINUSES</small> <ul style="list-style-type: none"> • Complete control and elimination of yield-threatening weeds • Protection from diseases for healthier farm output • Protection from insects for high yields and quality 	Explore AS, differentiate	
	2. PROBLEMS / PAINS PR <small>• ITS FREQUENCY</small> Major problems farmers face is crops being damaged by insects, animals, water and various climatic changes.	9. PROBLEM ROOT / CAUSE RC <ul style="list-style-type: none"> - Sense the animals and insects in the crop field. - Sense the water required for the crop. - Sense the required climate for the crop. 	7. BEHAVIOR BE <small>• ITS INTENSITY</small> Crop protector informs customer about the crop from being damaged by insect, animal, excess water flow, climatic changes, etc.		
Identify strong TR & EM	3. TRIGGERS TO ACT TR This triggers to protect the crops from insects, animals, excess water, climatic changes, unwanted plant growth etc.	10. YOUR SOLUTION SL The aim of the proposed work is to develop an IoT device for smart crop field monitoring system and automated irrigation system using the wireless sensor networks (WSN). To create an IoT device for monitoring the crop field using sensors (soil moisture, temperature, Humidity, etc...). To automate the irrigation by comparing the level of soil moisture with the threshold value.	8. CHANNELS of BEHAVIOR CH ONLINE Notifies the customer about the crops being damaged OFFLINE Senses the crops	Exit YOE	
	4. EMOTIONS EM <small>BEFORE / AFTER</small> Before there is no technology to protect the crop from insects, animals, excess water, climatic changes, unwanted plant growth, etc, so many farmers committed suicides.				

4 REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail .
FR-2	User Confirmation	Confirmation via Email .
FR-3	Interfacing with hardware	Interface the sensors with the software application so as to alert the farmers in case of any harm for crops .
FR-4	Database Connection	Databases are retrieved from IBM Cloudant .
FR-5	Mobile Application	Alarm and motors can be accessed from the mobile app .

4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FRNo.	Non-Functional Requirement	Description
NFR-1	Usability	The smart crop protection alerts the farmers in case of any obstacles and helps in protecting the crops
NFR-2	Security	Smart Agriculture can improve the farming practices and maintain sustainable production of crops especially by preventing the animals into the agricultural lands through IoT enabled devices
NFR-3	Reliability	With a proper power supply, SD card and programming the processor should be able to run 24/7 for years. The SD card and power supply will likely wear out faster than the Pi. The possible reasons behind Raspberry Pi failure can be power breakdowns, SD card failures, and ineligible environments.
NFR-4	Performance	Usage of an SD card module that helps to store as specified sound to scare the animals. Crop damage due to animal attack can be sensed. Network and Design Evaluation
NFR-5	Availability	Agriculture for different variety of crops is based on the monsoon changes, indoor and outdoor climatic temperatures, availability of rainfall and irrigation methods.
NFR-6	Scalability	The product shall be made available to everyone especially in remote areas for better efficiency of crop yield with the better safety of crops

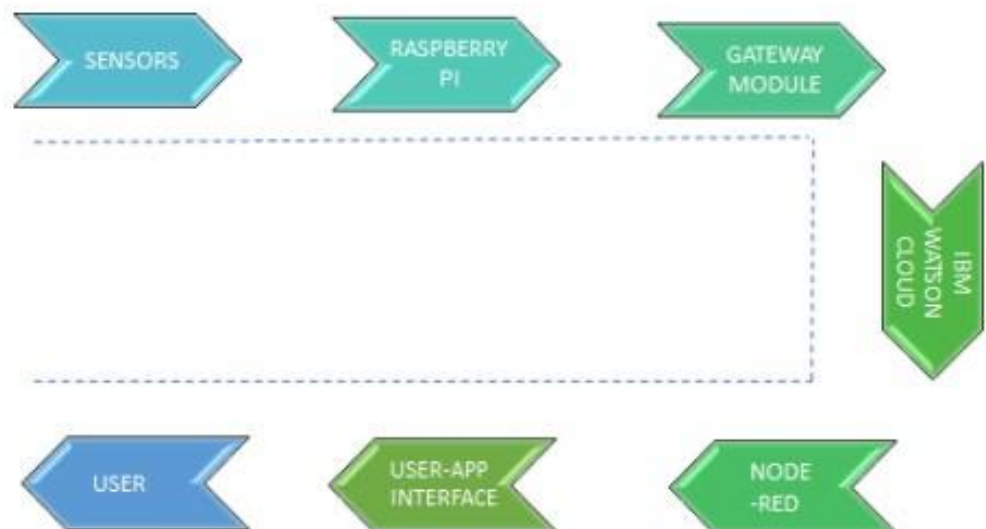
		swell asthefarmers.
--	--	---------------------

5PROJECTDESIGN

5.1 DataFlowDiagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the informationflows within a system. A neat and clear DFD can depict the right amount of the systemrequirement graphically. It shows how data enters and leaves the system, what changes theinformation,andwheredataisstored.

DATA FLOW DIAGRAM:

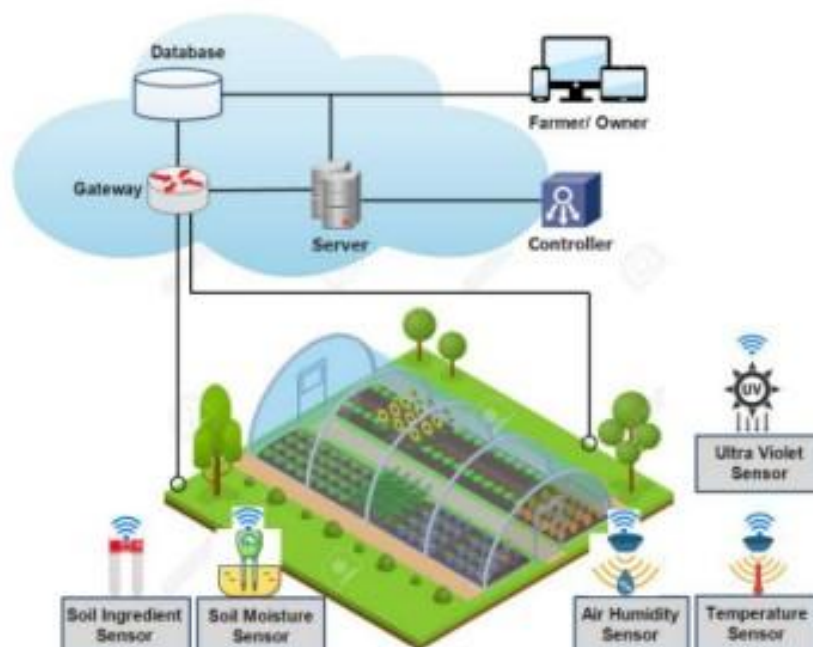


5.2 SOLUTIONANDTECHNICALARCHITECTURE

1. Summary:

The smart crops are constructed based on the sensor application and raspberry pi . It can also act as a transceiver since it is connected to the mobile phone of the user . The overall process of the sensors and raspberry pi is monitored using real time monitor which can help data transmission.This is stored and formulated using cloud data . Through which the admin can access the data and then track the location from GPS .

SOLUTION ARCHITECTURE :



2. Components&Technologies:

S.No	Component	Description	Technology
1.	UserInterface	Howuserinteractswith applicatione.g. WebUI,MobileApp, Chatbotetc.	HTML,CSS,JavaScript/A ngularJs/ReactJsetc.
2.	Application Logic-1	Logicforaprocessintheapplicatio n	Java/Python
	Database	DataType,Configurationsetc.	MySQL,NoSQL,etc
4.	IoT	To collect the data and alert the users	IBM Watson IoT Platform, Node Red.
5.	Cloud Database	Database ServiceonCloud	Cloudant DB

3. Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open Source framework	List the open-source frameworks used	Technology of Open source framework
2.	Security implementation	List all the security/access controls implemented, use of firewall etc.,	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3-tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

5.3 UserStories

Use the below template to list all the user stories for the product.

UserType	Functional Requirement(Epic)	UserStory Number	UserStory/Task	Acceptance criteria	Priority	Release
Customer(Farmer)	Maintain fields	USN-1	As a user, I can monitor the growth of crops and protect the crops against animals	I can maintain the fields with less labor	High	Sprint-1
	Analyzing problem	USN-2	As a user, I collect the required information about the problems on agriculture fields	I can ask my field owner directly.	low	Sprint-2
Project Designers	Identifying the problem and provides solutions	USN-3	As a user, I can sense the water level and flame in the field using sensor and monitor using IOT	I can perform the actions via IoT.	Medium	Sprint-1
Customer field Maintainer	Problem solution	USN-4	As a user, an area can be monitored from a remote place	Checking Process	Medium	Sprint-3
	Final process	USN-5	This proposed smart IOT-based crop protection device is found to be cost-effective and efficient	I can take necessary action if required	High	Sprint-4

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING & SCHEDULING:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project is done by gathering information about related details on technical papers and web browsing.	28 SEPTEMBER 2022
Prepare Empathy Map	Prepared Empathy Map Canvas to capture the user Pains & Gains list of problem statements.	24 SEPTEMBER 2022
Ideation	List the organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility and importance.	25 SEPTEMBER 2022
Proposed Solution	Prepared the proposed solution document which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepared problem - solution fit document.	30 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	28 SEPTEMBER 2022

6.2 SPRINTDELIVERYSCHEDULE

ProductBacklog,SprintSchedule,andEstimation

Use the below template to create product backlog and sprint schedule.

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (FRC)	User Story Number	User Story / Task	Story Points (#)	Priority (Low to High)	Team Member
Sprint-1	Registration	USN-1	As a user, I can register for the required dataset by entering my email, password, and confirming my password.	3	High	Umareshwari
Sprint-1		USN-2	As a user, I will receive confirmation email and the SMS once I have registered for the application.	2	High	Umareshwari
Sprint-2	Cloud services	USN-3	As a user, I can register for the application through Facebook or any social media.	1	Low	Ahmanya
Sprint-4		USN-4	As a user, I can register for the application through Gmail/web service.	2	Medium	Dhanani
Sprint-3	Login	USN-5	As a user, I can log into the application network by entering email & password.	4	High	Venkatesh
Sprint-2	Pre processing	USN-6	As a farmer, the user must be able to find the system easy to access so pre-processes and other task must be perfect.	3	High	Dhanani
Sprint-1	Collecting Dataset	USN-7	To collect various sources of animal threats and keep developing a dataset.	3	Medium	Umareshwari
Sprint-4	Integrating	USN-8	To integrate the available dataset and keep improving the accuracy of finding animals.	2	High	Ahmanya
Sprint-3		USN-9	To find and use appropriate compiler to run and test the data so that we can implement our program.	1	Low	Venkatesh
Sprint-2		USN-10	Request Savitri Engineering College to deploy the project in our campus and test.	1	Low	Dhanani
Sprint-1	Training	USN-11	As programmer, we need to train our data perfectly so that the program runs smoothly.	3	High	Umareshwari
Sprint-3		USN-12	Train the data using our available services and IBM dataset from server and improve that.	2	Medium	Venkatesh
Sprint-4	Coding	USN-13	To modify the code according to our program and improve the efficiency of that code.	4	High	Ahmanya
Sprint-2		USN-13	To improve performance.	1	Low	Venkatesh
Sprint-2	Record	USN-5	To record the data and plot the graph to show the characteristics officially.	4	High	Venkatesh
Sprint-1	Planning	USN-4	Plan the programming language and feasibility.	3	Medium	Umareshwari, Ahmanya
Sprint-4		USN-14	Demonstrate the working and improve accuracy overall.	2	Low	Ahmanya

ProjectTracker, Velocity&BurndownCharts

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5Days	20Oct2022	24Oct2022	20	21Oct2022
Sprint-2	20	5Days	25Oct2022	29Oct2022	20	27Oct2022
Sprint-3	20	5Days	31Oct 2022	4Nov2022	20	02Nov2022
Sprint-4	20	7Days	5Nov 2022	11Nov2022	20	08Nov2022

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	20 Oct 2022	24 Oct 2022	20	21 Oct 2022
Sprint-2	20	5 Days	25 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-3	20	5 Days	31 Oct 2022	4 Nov 2022	20	2 Nov 2022
Sprint-4	20	7 Days	5 Nov 2022	11 Nov 2022	20	8 Nov 2022

Velocity:

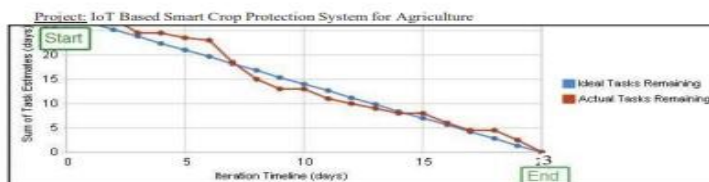
We have a 23-day sprint duration, and the velocity of the team is 20 (points per sprint).

To Find: Calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = \frac{23}{20} = 1.15$$

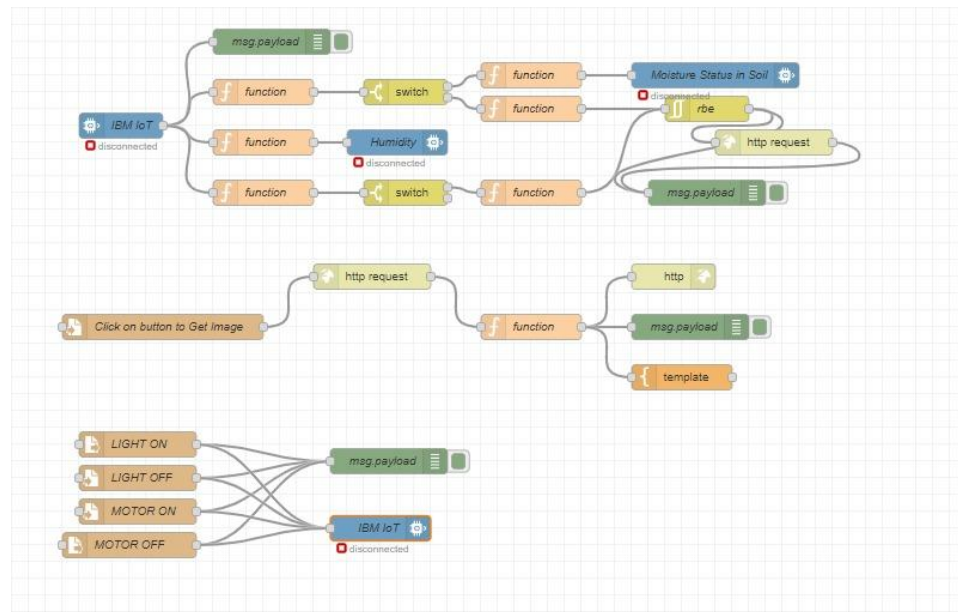
Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



7. CODINGANDSOLUTIONING

NODEREDSERVICEASSOCIATEDWITHIBMCLLOUD:



IBM Watson IoT Platform

umasevi814@gmail.com
ID: jeglvm

Browse Action Device Types Interfaces

Add Device +

Identity Device Information Recent Events State Logs

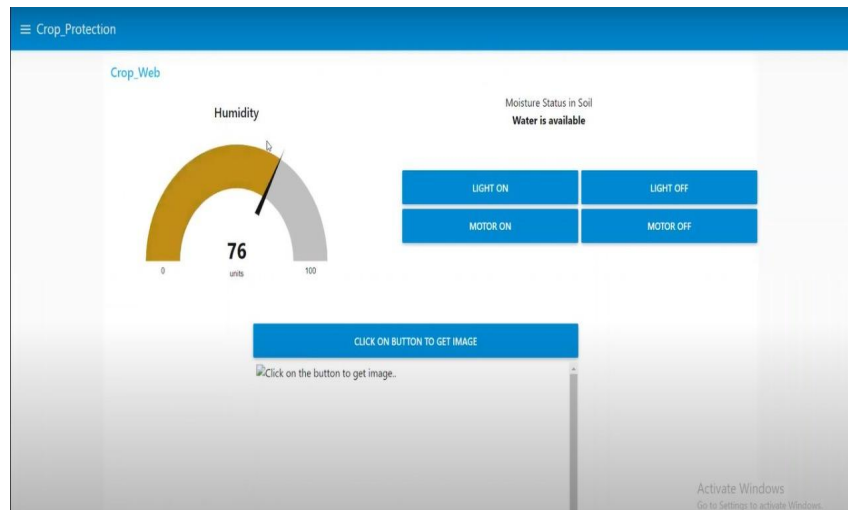
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Humidity	{"randomNumber":36}	json	a few seconds ago
Temperature	{"Temperature":3}	json	a few seconds ago
Moisture	{"Moisture":54}	json	a few seconds ago
Humidity	{"randomNumber":70}	json	a few seconds ago
Temperature	{"Temperature":68}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 Simulation running

NoderedDashboard:



9. **CONCLUSION**

We presented an intelligent Smart crop protection system. The system is based on IoT sensors. It is responsible for measuring the waste level in the smart crop. When the smart crop gets affected almost there will be information received by the admin, Since the admin can access the data and location of the crop. Later send this data (through Internet) to a server for storage and processing.

This data helps to compute the optimized collection routes for the workers. In future, we would like to enhance the system for different kind of crop management system .

10. **FUTURESCOPE**

The advantage of this work is its contribution in making a Smart crop. Among the many challenges that a city faces, crop protection management is of utmost importance. This is because, it is directly related to food of people living in the area. We are further extending this work to address problems of segregating different kind of crops (e.g., paddy, wheat, etc.,), and identifying different agricultural department for collecting it. The optimization algorithms may be devised accordingly depending on the requirements. In future, we would like to enhance the system for different kind of crops .

11.APPENDIX

11.1 SOURCECODE:

PYTHONCODETOPUBLISHDATA

```

import cv2
import numpy as np
import twiot.sdk.device
import playsound
import random
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError

# CloudantDB
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resource_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

# This is how you authenticate
metadata = (('authorization', 'key 0620e202302b4508b90eab7efe7475e4'),)
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID = "g5d4qO8EIgv4TWUCJj4hfEzgalqEjrDbE82AJDWlAOHo"
COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

```

```

COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-
storage:global:a/c2fa2836eaf3434bbc8b5b58feff3f0:62e450fd-4c82-4153-ba41-ccb53adb8111::"
clientdb = cloudant("apikey-W2njldnwtjO16V53LAVUCqPwc2aHTLmlj1xXvtdGKJBn",
"88cc5f47c1a28afbfb8ad16161583f5a", url="https://d6c89f97-cf91-48b7-b14b-c99b2fe27c2f-
bluemix.cloudantnosqldb.appdomain.cloud")
clientdb.connect()

```

```
#Create resource
```

```

cos = ibm_boto3.resource("s3",
ibm_api_key_id=COS_API_KEY_ID,
ibm_service_instance_id=COS_RESOURCE_CRN,
ibm_auth_endpoint=COS_AUTH_ENDPOINT,
config=Config(signature_version="oauth"),
endpoint_url=COS_ENDPOINT
)

```

```
def = multi_part_upload(bucket_name, item_name, file_path):
```

```
try:
```

```
print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
```

```
    #set 5 MB chunks
```

```
part_size = 1024 * 1024 * 5
```

```
    #set threadhold to 15 MB
```

```
file_threshold = 1024 * 1024 * 15
```

```
    #set the transfer threshold and chunk size
```

```
transfer_config = ibm_boto3.s3.transfer.TransferConfig(
```

```
multipart_threshold=file_threshold,
```

```
multipart_chunksize=part_size
```

```
)
```

```
    #the upload_fileobj method will automatically execute a multi-part upload
```

```
    #in 5 MB chunks size
```

```
with open(file_path, "rb") as file_data:
```

```
cos.Object(bucket_name, item_name).upload_fileobj(
```

```
Fileobj=file_data,
```

```
Config=transfer_config
```

```

        )

print("Transfer for {0} Complete!\n".format(item_name))
exceptClientError as be:
print("CLIENT ERROR: {0}\n".format(be))
except Exception as e:
print("Unable to complete multi-part upload: {0}".format(e))

defmyCommandCallback(cmd):
print("Command received: %s" % cmd.data)
command=cmd.data['command']
print(command)
if(commamd=="lighton"):
print('lighton')
elif(command=="lightoff"):
print('lightoff')
elif(command=="motoron"):
print('motoron')
elif(command=="motoroff"):
print('motoroff')
myConfig = {
    "identity": {
        "orgId": "chytun",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiot.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

database_name = "sample"

```

```

my_database = clientdb.create_database(database_name)
if my_database.exists():
    print(f'"{database_name}" successfully created.')
cap=cv2.VideoCapture("garden.mp4")
if(cap.isOpened()==True):
    print('File opened')
else:
    print('File not found')

while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv3.cvtColor(frame, cv2.COLOR_BGR@GRAY)
    imS= cv2.resize(frame, (960,540))
    cv2.imwrite('ex.jpg',imS)
    with open("ex.jpg", "rb") as f:
        file_bytes = f.read()

        #This is the model ID of a publicly available General model. You may use any other public or
        custom model ID.

    request = service_pb2.PostModeloutputsRequest(
        model_id='e9359dbe6ee44dbc8842ebe97247b201',
        inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))
        )])

    response = stub.PostModelOutputs(request, metadata=metadata)
    if response.status.code != status_code_pb2.SUCCESS:
        raise Exception("Request failed, status code: " + str(response.status.code))
    detect=False
    for concept in response.outputs[0].data.concepts:
        #print('%12s: %.f % (concept.name, concept.value))
    if(concept.value>0.98):
        #print(concept.name)
    if(concept.name=="animal"):
        print("Alert! Alert! animal detected")

```

```

playsound.playsound('alert.mp3')
picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
cv2.imwrite(picname+'.jpg',frame)
multi_part_upload('Dhakshesh', picname+'.jpg', picname+'.jpg')
    json_document={"link":COS_ENDPOINT+'/'+Dhakshesh+'/'+picname+'.jpg'}
new_document = my_database.create_document(json_document)
if new_document.exists():
    print(f"Document successfully created.")
    time.sleep(5)
    detect=True
    moist=random.randint(0,100)
    humidity=random.randint(0,100)
    myData={'Animal':detect,'moisture':moist,'humidity':humidity}
    print(myData)
    if(humidity!=None):
        client.publishEvent(eventId="status",msgFormat="json", daya=myData, qos=0, onPublish=None)
    print("Publish Ok..")
    client.commandCallback = myCommandCallback
    cv2.imshow('frame',imS)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
    client.disconnect()
    cap.release()
    cv2.destroyAllWindows()

```

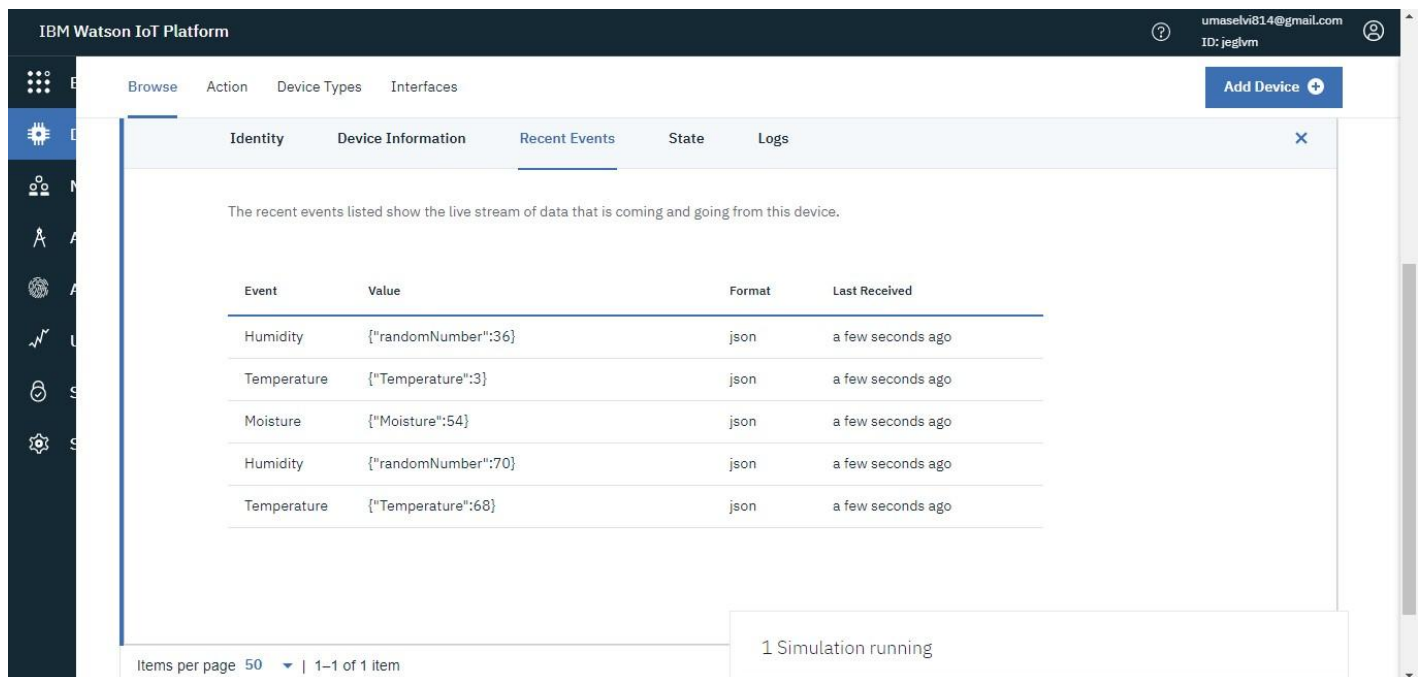

OUTPUT



```

IDLE Shell 3.8.8
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/HP/Desktop/crop/crop_protect.py =====
2021-04-06 12:52:19,640 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:hj5fmy:NodeMCU:12345
'sample' successfully created.
File opened
{'Animal': False, 'moisture': 17, 'humidity': 41}
Publish Ok..
{'Animal': False, 'moisture': 84, 'humidity': 16}
Publish Ok..
{'Animal': False, 'moisture': 48, 'humidity': 43}
Publish Ok..
{'Animal': False, 'moisture': 0, 'humidity': 3}
Publish Ok..
{'Animal': False, 'moisture': 73, 'humidity': 68}
Publish Ok..
{'Animal': False, 'moisture': 26, 'humidity': 26}
Publish Ok..
{'Animal': False, 'moisture': 96, 'humidity': 59}
Publish Ok..

```



IBM Watson IoT Platform

umasev814@gmail.com
ID: jeglvm

Browse Action Device Types Interfaces

Add Device +

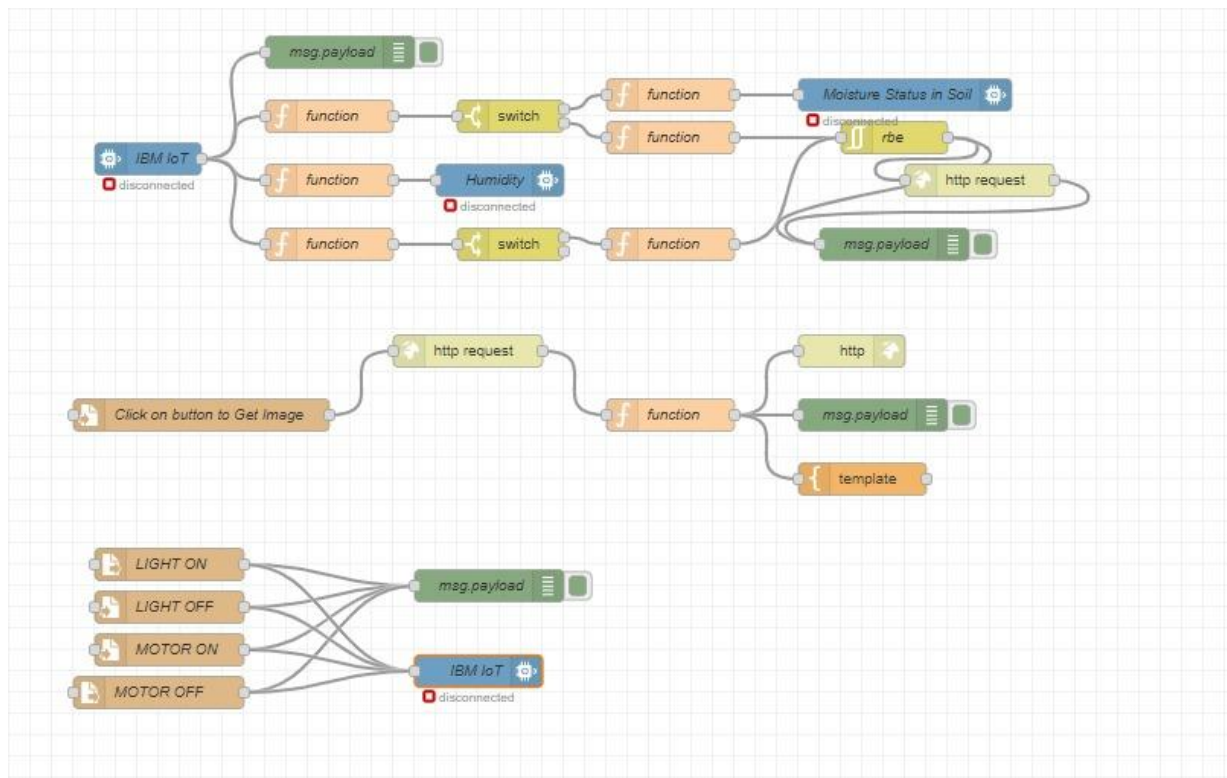
Identity Device Information **Recent Events** State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Humidity	{"randomNumber":36}	json	a few seconds ago
Temperature	{"Temperature":3}	json	a few seconds ago
Moisture	{"Moisture":54}	json	a few seconds ago
Humidity	{"randomNumber":70}	json	a few seconds ago
Temperature	{"Temperature":68}	json	a few seconds ago

Items per page 50 | 1-1 of 1 item

1 Simulation running



TECH TO SPEECH:

```
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
import playsound
```

```
authenticator = IAMAuthenticator('v9n8Zn4r5VpcMVz_HyRY0DrS13jSzph2IEFioVj4-vmT')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)
```

```
text_to_speech.set_service_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm')
```

```
with open('alert.mp3', 'wb') as audio_file:
    audio_file.write(
        text_to_speech.synthesize(
            'Alert! Alert! Animal Detected.',
            voice='en-US_ALLisonV3Voice',
            accept='audio/mp3'
        ).get_result().content
    )
    playsound.playsound('alert.mp3')
```

